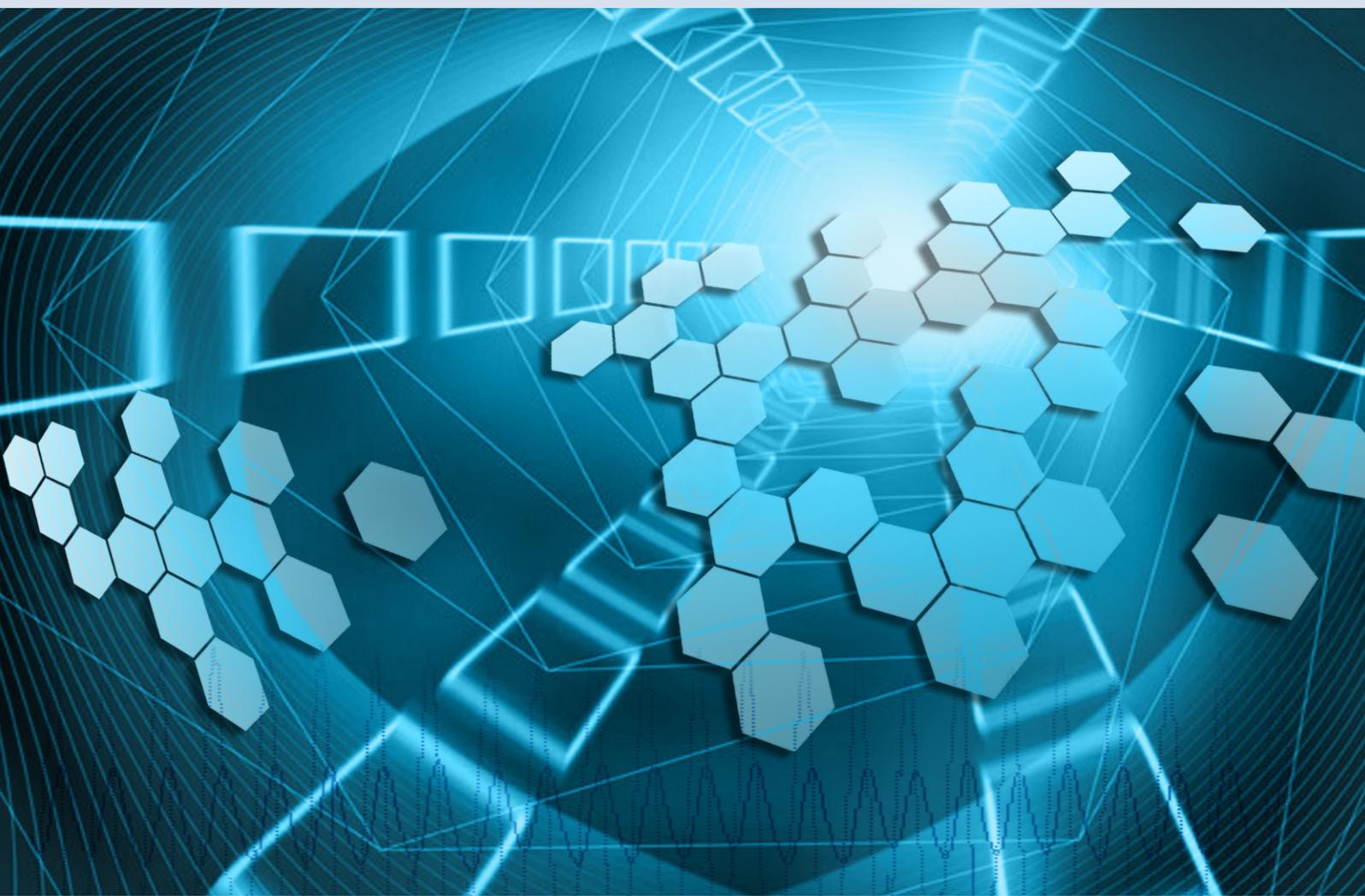


International Journal of Communications, Network and System Sciences

ISSN: 1913-3715

Volume 2, Number 1, February 2009



JOURNAL EDITORIAL BOARD

ISSN 1913-3715 (Print) ISSN 1913-3723 (Online)

[Http://www.scirp.org/journal/ijcns/](http://www.scirp.org/journal/ijcns/)

Editor-in-Chief

Prof. Tom Hou Department of Electrical and Computer Engineering, Virginia Tech., USA

Editorial Board

Prof. Dharma P. Agrawal University of Cincinnati, USA
Prof. Jong-Wha Chong Hanyang University, Korea (South)
Prof. Laurie Cuthbert University of London at Queen Mary, UK
Dr. Franca Delmastro National Research Council, Italy
Prof. Klaus Doppler Nokia Research Center, Nokia Corporation, Finland
Prof. Thorsten Herfet Saarland University, Germany
Dr. Li Huang Sticing IMEC Nederland, Netherlands
Prof. Chun Chi Lee Shu-Te University, Taiwan (China)
Prof. Myoung-Seob Lim Chonbuk National University, Korea (South)
Prof. Zhihui Lv Fudan University, China
Prof. Jaime Lloret Mauri Polytechnic University of Valencia, Spain
Dr. Kosai Raouf University of Joseph Fourier, Grenoble, France
Prof. Bimal Roy Indian Statistical Institute, India
Prof. Heung-Gyoon Ryu Chungbuk National University, Korea (South)
Prof. Rainer Schoenen RWTH Aachen University, Germany
Dr. Lingyang Song Philips Research, Cambridge, UK
Prof. Guoliang Xing Michigan State University, USA
Dr. Hassan Yaghoobi Mobile Wireless Group, Intel Corporation, USA

Editorial Assistant

Jing CAI Ting CHAN Wuhan University, China

Guest Reviewers

Joseph Y. H. So <i>MIT University</i>	Giorgio Di Natale <i>LIRMM</i>	Omar Elkeelany <i>Tennessee Technological University</i>
Hao Ma <i>Zhejiang University</i>	Yoan Shin <i>Soongsil University</i>	Zhi Fang <i>Institute of Semiconductors, CAS</i>
Xiaojing Huang <i>University of Wollongong</i>	Fengjie Yin <i>Liaoning University</i>	Feipei Lai <i>National Taiwan University Hospital</i>
Xiao-Hui Lin <i>Shenzhen University</i>	Fang Liu <i>Bell Labs Research China</i>	Yong Liang Guan <i>Nanyang Technological University</i>
Alejandro Jiménez Horas <i>Carlos III University of Madrid</i>	Riadh Dhaou <i>University of Toulouse</i>	Fasheng Wang <i>Neusoft Insitute of Information, Northeastern Univ.</i>

TABLE OF CONTENTS

Volume 2

February 2009

Game Theoretic Analysis of Joint Rate and Power Allocation in Cognitive Radio Networks	
D. LI, X. H. DAI, H. ZHANG.....	1
Synchronous Dynamic Adjusting: An Anti-Collision Algorithm for an RF-UCard System	
J. C. CAO, L. SHU, Z. D. LU.....	8
Performance Analysis of Sub-Rating for Handoff Calls in HCN	
X. L. WU, M. HE, F. WANG, J. ZHENG, E. REGENTOVA, G. S. HAO.....	21
A Priority Queuing Model for HCF Controlled Channel Access (HCCA) in Wireless LANs	
R. GHAZIZADEH, P. Z. FAN, Y. PAN.....	30
Gateways Placement in Backbone Wireless Mesh Networks	
M. TANG.....	44
Parallel-Transmission: A New Usage of Multi-Radio Diversity in Wireless Mesh Network	
Y. HU, S. B. YANG, Q. ZHANG, P. ZHANG.....	51
Generality Challenges and Approaches in WSNs	
F. TIRKAWI, S. FISCHER.....	58
Five Basic Types of Insider DoS Attacks of Code Dissemination in Wireless Sensor Networks	
Y. ZHANG, X. S. ZHOU, Y. M. JI, Y. W. LAW, M. PALANISWAMI.....	64
On the Scalable Fairness and Efficient Active Queue Management of RED	
H. WANG, X. H. LIN, K. Y. ZHOU, N. XIE, H. LI.....	73
A Genetic Based Fuzzy Q-Learning Flow Controller for High-Speed Networks	
X. LI, Y. W. JING, N. JIANG, S. Y. ZHANG.....	84

International Journal of Communications, Network and System Sciences (IJCNS)

Journal Information

SUBSCRIPTIONS

The *International Journal of Communications, Network and System Sciences* (Online at Scientific Research Publishing, www.SciRP.org) is published quarterly by Scientific Research Publishing, Inc. 5005 Paseo Segovia, Irvine, CA 92603-3334, USA.

E-mail: ijcns@scirp.org

Subscription rates: Volume 2 2009

Print: \$50 per copy.

Electronic: free, available on www.SciRP.org.

To subscribe, please contact Journals Subscriptions Department, E-mail: ijcns@scirp.org

Sample copies: If you are interested in subscribing, you may obtain a free sample copy by contacting Scientific Research Publishing, Inc at the above address.

SERVICES

Advertisements

Advertisement Sales Department, E-mail: ijcns@scirp.org

Reprints (minimum quantity 100 copies)

Reprints Co-ordinator, Scientific Research Publishing, Inc. 5005 Paseo Segovia, Irvine, CA 92603-3334, USA.

E-mail: ijcns@scirp.org

COPYRIGHT

Copyright© 2009 Scientific Research Publishing, Inc.

All Rights Reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as described below, without the permission in writing of the Publisher.

Copying of articles is not permitted except for personal and internal use, to the extent permitted by national copyright law, or under the terms of a license issued by the national Reproduction Rights Organization.

Requests for permission for other kinds of copying, such as copying for general distribution, for advertising or promotional purposes, for creating new collective works or for resale, and other enquiries should be addressed to the Publisher.

Statements and opinions expressed in the articles and communications are those of the individual contributors and not the statements and opinion of Scientific Research Publishing, Inc. We assumes no responsibility or liability for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained herein. We expressly disclaim any implied warranties of merchantability or fitness for a particular purpose. If expert assistance is required, the services of a competent professional person should be sought.

PRODUCTION INFORMATION

For manuscripts that have been accepted for publication, please contact:

E-mail: ijcns@scirp.org

Game Theoretic Analysis of Joint Rate and Power Allocation in Cognitive Radio Networks

Dong LI, Xianhua DAI, Han ZHANG

School of Information and Science Technology

Sun Yat-Sen University, Guangzhou, China

E-mail: lid3@mail2.sysu.edu.cn

Received September 10, 2008; revised November 26, 2008; accepted December 30, 2008

Abstract

Spectrum sharing is an essential enabling functionality to allow the coexistence between primary user (PU) and cognitive users (CUs) in the same frequency band. In this paper, we consider joint rate and power allocation in cognitive radio networks by using game theory. The optimum rates and powers are obtained by iteratively maximizing each CU's utility function, which is designed to guarantee the protection of primary user (PU) as well as the quality of service (QoS) of CUs. In addition, transmission rates of some CUs should be adjusted if corresponding actual signal-to-interference-plus-noise ratio (SINR) falls below the target SINR. Based on the modified transmission rate for each CU, distributed power allocation is introduced to further reduce the total power consumption. Simulation results are provided to demonstrate that the proposed algorithm achieves a significant gain in power saving.

Keywords: Rate Allocation, Power Allocation, Cognitive Radio, Game Theory

1. Introduction

Conventional fixed spectrum allocation policy has recently resulted in the intense competition for the use of radio spectrum due to the increasing number of various bandwidth-consuming wireless services. Moreover, these bands are not occupied or underutilized by licensed users most of time. According to the study from Federal Communication Commission (FCC) [1], the utilization of licensed bands ranges from 15% to 85%. In order to alleviate the problem of spectrum scarcity and improve the spectrum utilization, cognitive radio has been proposed as a flexible spectrum usage model [2-4]. In this technique, cognitive (unlicensed) user (CU) are allowed to have opportunistic access to idle spectrums or to the busy ones without causing harmful interference to the primary (licensed) user (PU). The major advantage of cognitive radio technology is its ability to search available spectrums in its surrounding environment and adjust its transmit parameters accordingly to enhance the system performance. The transmit parameters, for example, include modulation scheme, beamforming weight center frequency, transmit power and so on. In this paper, we focus on the allocation of transmission rate and power in cognitive radio networks. In particular,

in such spectrum sharing model where CU operating in the same frequency band with PU, these transmit parameters should be adjusted to maintain interference introduced to PU within a given limit while satisfying the quality of service (QoS) of CU. Therefore, interference introduced by multi-user or co-channel transmission at the same time or over the same frequency radio channel is inevitable.

In this paper, we consider applying game theory to spectrum sharing in cognitive radio networks by adjusting transmission rate and power. Related works are shown in [5-6]. In [5] and [6], two different distributed power allocation algorithms are investigated in a game theoretic perspective, with regard to the idle and busy spectrum, respectively. However, these power allocation algorithms are not effective in guaranteeing the QoS of CUs in bad channel conditions, due to the rigid target signal-to-interference-plus-noise ratio (SINR) constraint. Further studies are given in [7-9], in which joint rate and power allocation has been considered by using game theory. These works can be applied to the systems where only CUs share the same frequency band with the absence of PU. Unfortunately, they can not be extended to the scenario of the co-existence of PU and multiple CUs in the same frequency band, since they do not

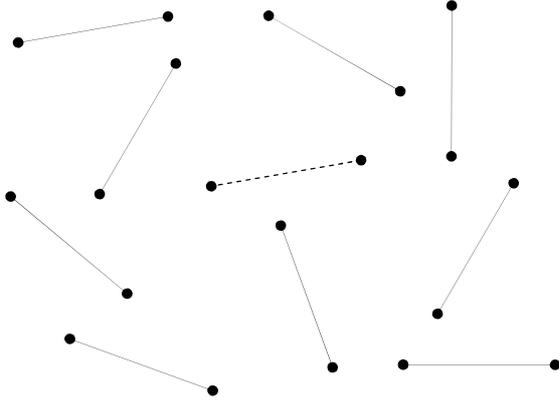


Figure 1. System model with one PU in dashed line and N CUs in solid line.

consider the protection of PU. While literature [10] and [11] only consider the problem of joint rate and power allocation of a single CU in the presence of PU, where there is no competition for the available spectrum resource among multiple CUs.

In this contribution, our goal is, therefore, to jointly optimize the transmission rates and power levels in order to accommodate CUs as many as possible, while guaranteeing the protection of PU and the QoS of CUs. In this paper, we first formulate the problem as a supermodular game [5] which is proved to have at least a Nash Equilibrium (NE), and then give the solution by maximizing the utility function for each CU. After the adjustment of transmission rate, the improved SINR is assured to converge to the predefined target SINR for each CU. It will demonstrate that the distributed power allocation algorithm [12] based on the adaptive transmission rate can further reduce the total power consumption. The major advantage of our proposed algorithm is that it is implemented in a totally distributed manner without the need of access point (AP) [13], and its computational complexity is low since the transmit powers require only few iterations to converge.

The rest of this paper is organized as follows: Section 2 presents the system model and basic assumptions. Section 3 formulates the problem using game theory. In Section 4, we develop the proposed algorithm for joint rate and power allocation in cognitive radio networks. Performance analysis of the proposed algorithm is investigated in Section 5. Section 6 concludes this paper.

Notation: All vectors and matrices are denoted in bold letters. \mathbf{I}_N stands for $N \times N$ identity matrix. $\mathbf{A}_{i,j}$ denotes the (i, j) thelement of the matrix \mathbf{A} .

2. System Model

The cognitive radio network under consideration is composed of one PU and N CUs, which are modeled as a collection of separate (N+1) transmit-receive pairs with a single channel, as illustrated in Figure 1. All CUs are

allowed to transmit at the same time and share the same frequency band by adopting code division multiplexing access (CDMA). The transmission mode for each CU is half-duplex in order to avoid self-interference [14] caused by one node simultaneously transmitting and receiving. The channel propagation model is characterized by average path loss, which is given by [15]

$$\overline{PL}(d) = \overline{PL}(d_0) + 10\alpha \log_{10}\left(\frac{d}{d_0}\right) \text{dB} \quad d \geq d_0 \quad (1)$$

where d_0 and d are the reference and transmitter-receiver (T-R) distance, respectively. α denotes path loss exponent, which depends on propagation circumstance.

Then, the actual SINR for i th CU can be expressed as

$$\text{SINR}_i = \frac{B}{R} \frac{G_{ii}P_i}{\sum_{j=1, i \neq j}^N G_{ij}P_j + G_{i0}P_0 + \eta} \geq \gamma \quad (2)$$

where P_i and P_0 denote power level of i th CU and PU, respectively. B is the spectrum bandwidth, R is the transmission rate and $K = \frac{B}{R}$ denotes the spreading gain. G_{ii} is the channel gain over CU i , G_{ij} and G_{i0} represent the channel gain between CU j ' transmitter, PU's transmitter and CU i ' receiver, respectively. η is the background noise power. γ is the target SINR for all CUs and the constraint $\text{SINR}_i \geq \gamma$ guarantee the QoS for i th CU. On the other hand, the total interference introduced to PU is given by

$$\xi_0 = \sum_{i=1}^N G_{0i}P_i \leq \xi \quad (3)$$

where G_{0i} represents the channel gain between CU i ' transmitter and PU's receiver and ξ denotes the maximum tolerable interference introduced to PU.

Throughout this paper, we make the following assumptions:

- The local information of channel gains and SINR measurements at the receivers of all CUs are sent to their respective transmitters via a dedicated feedback channel.
- All CUs are well synchronized, are assumed to be immobile or move slowly so that the corresponding channel gain remains constant during the convergence of transmit powers.

3. Game Formulation

The objective of this algorithm is to assign constrained transmit powers and available transmission rates to all CUs, in order to minimize the total power consumptions while satisfying the target SINR constraint of CUs. Besides, we should also consider maintaining the interference introduced to the PU within a given inter-

ference limit, since CUs coexist with the PU in the same frequency band. Therefore, we can formulate the following constrained optimization problem.

$$\text{minimize} \quad \sum_{i=1}^N P_i \quad (4)$$

$$\text{subject to} \quad \text{SINR}_i \geq \gamma \quad (5)$$

$$\xi_0 \leq \xi \quad (6)$$

where $P_i \in [0, P_{\max}]$ and P_{\max} is the maximum transmit power.

In what follows, we start with introducing three basic elements in this game, and then prove the existence of Nash Equilibrium (NE). Finally we will give the solution to this game based on the above analysis.

3.1. Elements of the Game

In normal form, a game consists of three elements in the following way.

$$G = \{ \mathbb{N}, A, \{U_i\}_{i \in \mathbb{N}} \} \quad (7)$$

where $\mathbb{N} = \{1, 2, \dots, N\}$ is the set of players, $A = A_1 \times A_2 \times \dots \times A_N$ denotes action space for all players with A_i presenting the set of action of player i , while $\{U_i\}_{i \in \mathbb{N}} = \{U_1, U_2, \dots, U_N\}$ is the set of utility functions.

More specifically in this game, the set of players \mathbb{N} is given by the set of CUs. The action for each player (or CU) is denoted as $A_i = P_i$. In order to capture each CU's satisfaction with both the constraints (5) and (6), the utility function of player $i \in \mathbb{N}$ is expressed as

$$\begin{aligned} U_i(P_i, P_{-i}) &= -(\gamma - 10 \log_{10} \text{SINR}_i)^2 - \left(\frac{\omega_i \xi}{\beta} - G_{0i} P_i \right)^2 \\ &= -(\gamma - 10 \log_{10} \left(\frac{B}{R} G_{ii} P_i \right) + 10 \log_{10} \left(\sum_{j=1, j \neq i}^N G_{ij} P_j + G_{i0} P_0 + \eta \right))^2 \\ &\quad - \left(\frac{\omega_i \xi}{\beta} - G_{0i} P_i \right)^2 \end{aligned} \quad (8)$$

where $\omega_i = \frac{G_{0i}^2}{\sum_{j=1}^N G_{0j}^2}$ is the weighted coefficient and P_{-i}

represents the actions for all players except i . Note that the utility function in [7] only considers the QoS of CUs without guaranteeing the protection of PU. In Equation (8), $\beta \geq 1$ is a constant, and the interference constraint of PU can be easily satisfied by increasing the value of β , because the powers allocated to all CUs are kept at a low level. However, it does not mean that large β might not satisfy the target SINR requirement for all CUs.

3.2. Existence of Nash Equilibrium

Nash Equilibrium is the steady state in the game, in which no player can increase its utility function from unilaterally deviating its action. Mathematically speaking, NE is an action-tuple a'_i , which satisfies the following property

$$U_i(a'_i, a'_{-i}) \geq U_i(a_i, a'_{-i}), \quad a_i \in A_i \quad (9)$$

However, it does not follow that there is a NE existing in every game. Therefore, it becomes necessary to testify the existence of NE.

Theorem 1: $G = [\mathbb{N}, A, \{U_i\}_{i \in \mathbb{N}}]$ is a supermodular game.

Proof: The action space is compact since it is both closed and bounded. In addition, the utility function of player $i \in \mathbb{N}$ is twice differentiable.

$$\begin{aligned} \frac{\partial U_i(\mathbf{P})}{\partial P_i} &= \frac{10}{P_i \times \ln 10} (\gamma - 10 \log_{10} \left(\frac{B}{R} G_{ii} P_i \right) + 10 \log_{10} \\ &\quad \left(\sum_{j=1, j \neq i}^N G_{ij} P_j + G_{i0} P_0 + \eta \right)) + 2 G_{0i} \left(\frac{\omega_i \xi}{\beta} - G_{0i} P_i \right) \end{aligned} \quad (10)$$

$$\frac{\partial^2 U_i(\mathbf{P})}{\partial P_i \partial P_j} = \frac{G_{ij}}{\sum_{j=1, j \neq i}^N G_{ij} P_j + G_{i0} P_0 + \eta} \frac{10^2}{P_i \times (\ln 10)^2} > 0 \quad (11)$$

According to the definition and property of game modes in [7], this game is a supermodular game and there must be at least one NE in this supermodular game.

3.3. Solution to the Game

Since the existence of NE in this game has been proved, we consider the problem of how to identify it. The optimum transmit power or NE can be obtained in such a way that each CU maximizes its corresponding utility function iteratively, which is assured to converge assuming each CU acts in its own interest. Mathematically speaking, the process can be expressed in the following way.

$$P_i^* = \arg \max U_i(P_i, P_{-i}), \quad i \in \mathbb{N} \quad (12)$$

where $P_i^* \in [0, P_{\max}]$.

It should be noted that there is no sufficient guarantee in this game with regard to constraint (5) and (6). First, the protection of PU is not assured if β is not set appropriately. Second, due to the rigid SINR requirement of each CU, the corresponding QoS of some CUs which experience strong interference or deep path loss will be violated. In the next section, we will give details to solve these problems.

4. Joint Rate and Power Allocation Algorithm

In this section, we perform a two-stage processing to

make sure that both the interference constraint of PU and SINR constraint of CUs are satisfied. The transmit power is first allocated to each CU using Equation (12), in which the transmission rate is the same for each CU. If the PU experience harmful interference, the value of β and transmit power in Equation (8) are both updated until the interference introduced to PU is kept at an acceptable level. If the QoS of some CUs are not guaranteed, the transmission rate is adjusted so that the corresponding actual SINR is no less than the target SINR. Besides, it will be demonstrated that the total power consumption is greatly reduced by iteratively update powers based on the modified transmission rate and convergent power first allocated for each CU, while the constraint (5) and (6) can also be satisfied. The whole process can be summarized as follows.

1. Initialize: $\mathbf{P} = (P_{\max}^{(1)}, P_{\max}^{(2)}, \dots, P_{\max}^{(N)})$, $\beta = 2$;
2. Calculate $P_i^* = \arg \max U_i(P_i, P_{-i})$, $i \in \mathbb{N}$;
3. If both the constraints (5) and (6) are satisfied, stop;
4. elseif constraint (5) is not satisfied
5. evoke rate allocation process;
6. elseif constraint (6) is not satisfied
7. $\beta \leftarrow 2 * \beta$, go to 2;
8. end if
9. evoke power allocation process

4.1. Rate Allocation

Rate allocation enables the system to support various data rates by varying the number of bits per second in accordance with the instantaneous SINR. In this case, the actual SINR for i th CU can be rewritten in the following way.

$$SINR_i = \frac{B}{R_i} \frac{G_{ii} P_i}{\sum_{j=1, i \neq j}^N G_{ij} P_j + G_{i0} P_0 + \eta} \geq \gamma \quad (13)$$

where $R_i \in \mathbf{R}$ and \mathbf{R} denotes the set of available transmission rates. Here, $K_i = \frac{B}{R_i}$ represents the spreading gain for i th CU. In this paper, we assume each CU is equipped with the Walsh-Hadamard code and a set of different processing gains denoted as $\mathbf{K} = \{2^n\}$, where n is the positive integer. Therefore, the available transmission rates in \mathbf{R} can be obtained in such a way that B is divided by the corresponding processing gain in \mathbf{K} .

As mentioned before, due to the rigid target SINR constraint, some CUs in bad channel conditions can not satisfy the QoS requirement, in which the transmission rate is the same for each CU and chosen to be the maximum one in \mathbf{R} (or equivalently the minimum processing gain in \mathbf{K}). In order to achieve all CUs' convergence to the target SINR, the transmission rate

should be adjusted according to the convergent power for each CU. Specifically, the adjusted transmission rate is determined in such a way that the corresponding improved SINR is no less than γ . In this case, there may be many available transmission rates which satisfy the above requirement, and the maximum one should be chosen from them in order to achieve the transmission rate as high as possible. If the improved SINR still falls below γ , the corresponding CU will be switched off, for its QoS requirement can not be met. Therefore, the target SINR constraint of all CUs can be satisfied.

4.2. Power allocation

Since the constraint (5) and (6) are both satisfied as discussed before, we consider further reducing the total power consumption for all CUs. Then, the constraint (5) can be expressed in the following way

$$P_i - \sum_{j=1, i \neq j}^N \frac{\gamma G_{ij}}{K G_{ii}} P_j \geq \frac{\gamma(G_{i0} P_0 + \eta)}{G_{ii}} \quad (14)$$

Note that the target SINR is not the same for all CUs in this stage. Let $\mathbf{P} = (p_1, p_2, \dots, p_N)^T$, rewrite (14) with equality in the matrix form, we can obtain

$$(\mathbf{I}_N - \mathbf{F})\mathbf{P} = \mathbf{U} \quad (15)$$

where \mathbf{F} and \mathbf{U} are given by

$$\mathbf{F}_{i,j} = \begin{cases} \frac{\gamma G_{ij}}{K G_{ii}}, & i \neq j, i, j = 1, 2, \dots, N \\ 0, & i = j \end{cases} \quad (16)$$

$$\mathbf{U} = \left(\frac{\gamma(G_{10} P_0 + \eta)}{G_{11}}, \frac{\gamma(G_{20} P_0 + \eta)}{G_{22}}, \dots, \frac{\gamma(G_{N0} P_0 + \eta)}{G_{NN}} \right)^T \quad (17)$$

The Equation (15) can be rewritten in the following form as

$$\mathbf{P}(k+1) = \mathbf{F}\mathbf{P}(k) + \mathbf{U} \quad (18)$$

where $\mathbf{P}(k+1)$ and $\mathbf{P}(k)$ denote the vector of power level at next and current iteration, respectively. Then, the optimal transmit power can be obtained by iteration of [5]

$$P_i(k+1) = \min(P_{\max}, \frac{\gamma}{SINR_i(k)} P_i(k)) \quad (19)$$

Note that the above algorithm terminate with convergent power if $|P_i(k+1) - P_i(k)| \leq \zeta$, where $\zeta > 0$ is a negligibly small error. Based on powers first allocated to all CUs and improved SINR which satisfies the constraint (5), it can be known that the total power consumption will be reduced after second allocation of powers using Equation (19) in which the first allocated powers using Equation (12) are used as initialized powers. The following theorem supports our conclusion.

Theorem 2: Given $P_i^{(1)}$ and corresponding $SINR_i^{(1)}$ satisfying $SINR_i^{(1)} \geq \gamma$, $i \in \mathbb{N}$, then there exists a steady-

Table 1. Simulation parameters.

Number of CUs (N)	9
Noise variance (η)	10^{-4}
Negligible error (ζ)	10^{-4}
Reference distance (d_0)	1m
Path-loss factor (α)	4
Spectrum bandwidth (B)	5.12MHz
Set of processing gains (\mathbf{K})	{16,32,64,128,256}
Maximum power (P_{\max})	1W
Interference limit (ξ)	0.1W
Maximum number of iteration	20

state $P_i^{(2)}$ to achieve $\sum_{i=1}^N P_i^{(2)} \leq \sum_{i=1}^N P_i^{(1)}$ while satisfying both the constraint (5) and (6).

Proof: It can be known from Equation (19) that, if SINR for each CU satisfies the condition $SINR_i(k) \geq \gamma$, then we have

$$\begin{aligned} P_i^{(1)}(k+1) &= \min(P_{\max}, \frac{\gamma}{SINR_i(k)} P_i^{(1)}(k)) \\ &= \frac{\gamma}{SINR_i(k)} P_i^{(1)}(k) < P_i^{(1)}(k) \end{aligned}$$

The iteration will terminate if there is no negligible change in $P_i^{(1)}(k)$ such that $|P_i^{(1)}(k+1) - P_i^{(1)}(k)| \leq \zeta$. That is to say that $SINR_i^{(2)} = \gamma$ when transmit power converges, so that constraint (5) is satisfied. Therefore, the convergent power $P_i^{(2)} = P_i^{(1)}(k+1) \leq P_i^{(1)}(1) = P_i^{(1)}$.

As a result, we have $\sum_{i=1}^N P_i^{(2)} \leq \sum_{i=1}^N P_i^{(1)}$, and the constraint

$$\begin{aligned} (6) \text{ is also satisfied with } \xi_0^{(2)} &= \sum_{i=1}^N G_{0i} P_i^{(2)} \leq \xi_0^{(1)} \\ &= \sum_{i=1}^N G_{0i} P_i^{(1)} \leq \xi. \end{aligned}$$

5. Simulation Results

In this section, we provide numerical results to demonstrate the effectiveness of the proposed algorithm in reduction of the total power consumption while satisfying both the target SINR constraint of CUs and interference constraint of PU. In our simulation, we consider the cognitive radio network placed in a $10\text{m} \times 10\text{m}$ square area, in which transmit nodes are located uniformly and the corresponding receive nodes are random placed within $6\text{m} \times 6\text{m}$ square area centered around them. The specific parameters used in this simulation are listed in Table 1, in which the channel gain can be expressed as $G_{ij} = d_{ij}^{-4}$, where d_{ij} is the distance between j th CU's transmitter and i th CU's

receiver. According to Table 1, the available transmission rates can be obtained with the result of $\mathbf{R} = \{320, 160, 80, 40, 20\}$ kbps.

First, we examine the convergence performance of proposed game model with respect to transmit power and actual SINR for each CU, which are illustrated in Figure 2 (a) and (b), respectively. Figure 2 (a) shows that the transmit power for each CU converges to the steady state after several iterations, in which the transmission rate is 320 kbps and the corresponding processing gain is accordingly 16 for each CU. From Figure 2 (a), we observe that there are 5 CUs still transmitting at the maximum power. This is because these CUs experience strong interference or deep path loss, and the corresponding transmit power should be kept at a high level in order to satisfy the SINR constraint. From Figure 2 (b), we can find that the actual SINR of each CU can not converge to the predefined $\gamma = 16\text{dB}$ at the same time, no matter how high the corresponding transmit power is. This is due to the hard target SINR constraint as mentioned before. Note that, β is set to be 2 and there is no need to update β , because, in this case, the total interference introduced to PU is 0.0112 W which satisfies the constraint (6).

Next, we investigate the convergence performance of the proposed algorithm with respect to transmit power and improved SINR for each CU, which are illustrated in Figure 3(a) and (b), respectively. Figure 3(a) shows that, by adjusting the corresponding transmission rate, more power can be saved after the convergence of (19). To be specific, the total power consumption is greatly reduced from 5.9 W to 0.5408 W. Here, the total interference is 0.0012 W. Note that in Figure 3(a), the inputs are the convergent transmit powers in Figure 2(a) with improved SINR satisfying constraint (5). As for Figure 3(b), the adjusted transmission rates for all CUs are 320, 160, 80, 40, 160, 0, 0, 0 and 160 kbps, which correspond to the processing gain 16, 32, 64, 128, 32, 0, 0, 0 and 32, respectively. It should be noted that 3 CUs are switched off, in which the corresponding improved SINR still falls below γ . As can be seen from Figure 3(b), the improved SINRs for the remainder of CUs all converge to $\gamma = 16\text{dB}$, and we can find that more CUs are permitted to transmit after the adjustment of their corresponding transmission rates compared with Figure 2(b).

In our simulations so far, the constant transmit power of PU P_0 and target SINR γ are assumed. To be more practical, we finally study the impact of different P_0 and γ on the performance of proposed game model in terms of the total power consumption. Figure 4 depicts the total transmit power versus different γ after the convergence of (12), where P_0 is varied at 10W, 20W and 30W, respectively. As can be seen from Figure 4, the total power consumption increases with the increasing

value of P_0 . This is because the interference introduced to each CU increases as P_0 is increased, which means that each CU will increase its transmit power accordingly, in order to satisfy the target SINR constraint. Besides, we also notice that the total power consumption increases with the increasing value of γ . This is due to the fact that the transmit power should be increased so that the corresponding target SINR requirement can be met.

6. Conclusions

In this paper, we have proposed and investigated a joint rate and power allocation algorithm in cognitive radio networks by using game theory. The objective of the proposed algorithm is to minimize the total power consumption, while satisfying the target SINR constraint of CUs and keeping the interference introduced to PU below a given limit. We have analyzed the problem as a super-modular game and obtained the NE which leads to the optimal transmit powers. In order to solve the

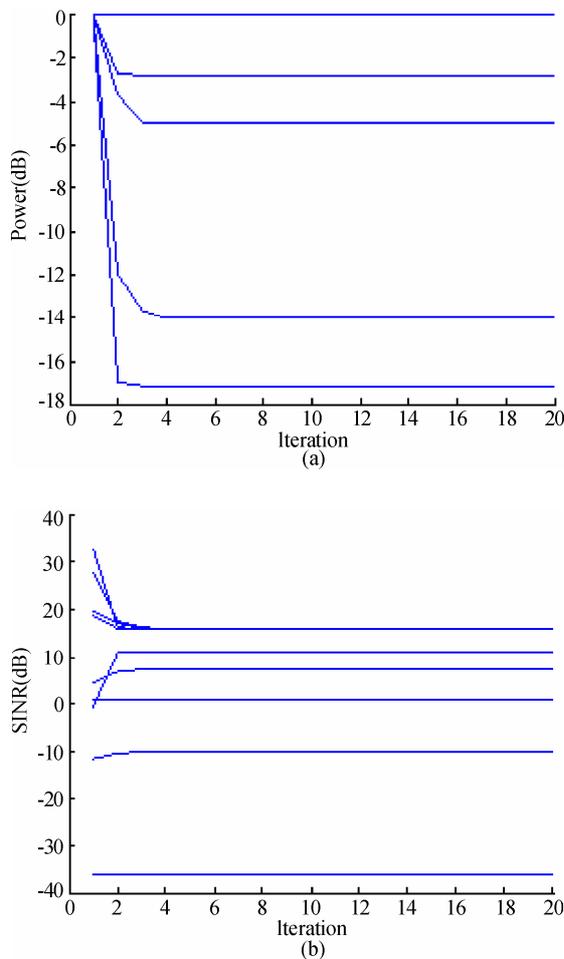


Figure 2. Convergence of transmit power and actual SINR for each CU using Equation (12), where $P_0 = 10W$ and $\gamma = 16dB$. (a) Transmit Power (dB). (b) Actual SINR (dB).

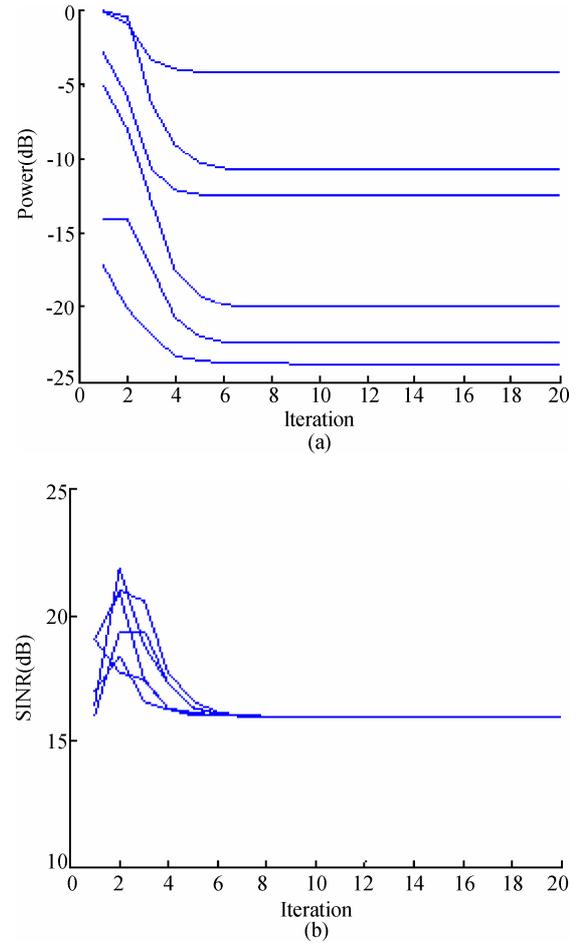


Figure 3. Convergence of transmit power and improved SINR for each CU using Equation (19), where $P_0 = 10W$ and $\gamma = 16dB$. (a) Transmit power (dB). (b) Actual SINR (dB).

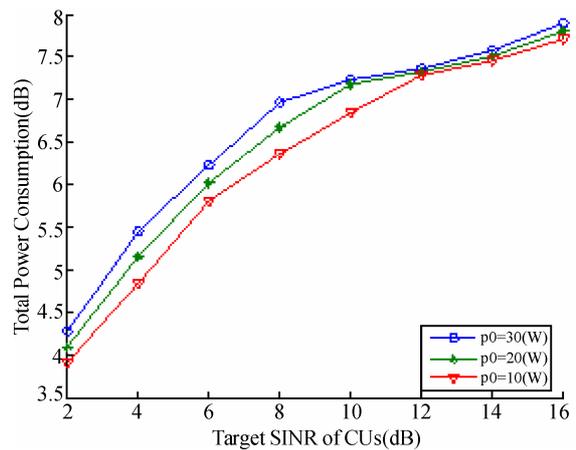


Figure 4. Convergence of total transmit power versus different target SINR γ after the convergence of (12).

inherent γ -divergence problem, the distributed power allocation algorithm with adaptive transmission rate has been introduced, which has been proved to achieve significant improvement in the power efficiency.

Simulation results are shown to confirm the effectiveness of the proposed algorithm.

7. Acknowledgement

This work is supported by the National Science Foundation of China (NSFC), Grant 60772132, and the joint foundation of National Science Foundation of China (NSFC) and Guangdong Province U0635003, and also supported by the Science & Technology Project of Guangdong Province, Grant 2007B010200055.

8. References

- [1] Federal Communications Commission, "Spectrum Policy Task Force," Reports, ET Docket No. 02-135, November 2002.
- [2] J. Mitola and G. Q. Maguire, "Cognitive radio: Making software radios more personal," *IEEE Personal Communications*, Vol. 6, No. 4, pp. 13-18, August 1999.
- [3] J. Mitola, "Cognitive radio: An integrated agent architecture for software defined radio," Tekn. Dr. dissertation, Royal Institute of Technology (KTH), Stockholm, Sweden, 2000.
- [4] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, Vol. 23, No. 2, pp. 201-220, February 2005.
- [5] J. Neel, R. Menon, J. H. Reed, and A. B. MacKenzie, "Using game theory to analyze physical layer cognitive radio algorithms," [Online], available: <http://web.syr.edu/~ejhumpfr/>.
- [6] H. Li, Y. Gai, Z. He, K. Niu, and W. Wu, "Optimal power control game algorithm for cognitive radio networks with multiple interference temperature limits," in *Proceedings of IEEE Vehicular Technology Conference*, pp. 1554-1558, May 2008.
- [7] M. Hayajneh and C. T. Abdallah, "Distributed joint rate and power control game-theoretic algorithms for wireless data," *IEEE Communications Letters*, Vol. 8, No. 8, pp. 511-513, August 2004.
- [8] M. R. Musku, A. T. Chronopoulos, and D. C. Popescu, "Joint rate and power control with pricing," in *Proceedings of IEEE Global Telecommunications Conference*, pp. 3466-3470, November 2005.
- [9] M. R. Musku, A. T. Chronopoulos, and D. C. Popescu, "Joint rate and power control using game theory," in *Proceedings of IEEE Consumer Communications and Networking Conference*, pp. 1258-1262, January 2006.
- [10] L. Guo, P. Wu, and S. Cui, "Power and rate control with dynamic programming for cognitive radios," in *Proceedings of IEEE Global Telecommunications Conference*, pp. 1699-1703, November 2007.
- [11] M. Hong, J. Kim, H. Kim, and Y. Shin, "An adaptive transmission scheme for cognitive radio systems based on interference temperature model," in *Proceedings of IEEE Consumer Communications and Networking Conference*, pp. 69-73, January 2008.
- [12] G. J. Foschini and Z. Miljanic, "A simple distributed autonomous power control algorithm and its convergence," *IEEE Transactions on Vehicular Technology*, Vol. 42, No. 4, pp. 641-646, November 1993.
- [13] IEEE 802.22 Working Group on Wireless Regional Area Networks, <http://www.ieee802.org/22>.
- [14] T. ElBatt and A. Ephremides, "Joint scheduling and power control for wireless ad hoc networks," *IEEE Transactions on Wireless Communications*, Vol. 3, No. 1, pp. 74-85, January 2004.
- [15] T. S. Rappaport, *Wireless communications principles and practice*, Prentice Hall Inc., 1996.

Synchronous Dynamic Adjusting: An Anti-Collision Algorithm for an RF-UCard System

Jichang CAO, Lin SHU, Zhengding LU

School of Computer Science & Technology

Huazhong University of Science & Technology, Wuhan, China

Email: shulin3760@163.com

Received September 27, 2008; revised December 6, 2008; accepted December 8, 2008

Abstract

An RF-UCard system is a contactless smartcard system with multiple chip operating systems and multiple applications. A multi-card collision occurs when more than one card within the reader's read field and thus lowers the efficiency of the system. This paper presents a novel and enhanced algorithm to solve the multi-card collision problems in an RF-UCard system. The algorithm was originally inspired from framed ALOHA-based anti-collision algorithms applied in RFID systems. To maximize the system efficiency, a synchronous dynamic adjusting (SDA) scheme that adjusts both the frame size in the reader and the response probability in cards is developed and evaluated. Based on some mathematical results derived from the Poisson process and the occupancy problem, the algorithm takes the estimated card quantity and the new arriving cards in the current read cycle into consideration to adjust the frame size for the next read cycle. Also it changes the card response probability according to the request commands sent from the reader. Simulation results show that SDA outperforms other ALOHA-based anti-collision algorithms applied in RFID systems.

Keywords: RF-UCard, Anti-collision Algorithm, Synchronous Dynamic Adjusting, RFID, ALOHA, DFSA, BBEI

1. Introduction

Identification is a central concept in user-oriented and ubiquitous computing. Radio Frequency Identification (RFID) is one of the key technologies for identifying physical objects permits remote, non-line-of-sight, and automatic reading. There is a wide variety of RFID products and applications available; the book [1] provides a good overview. A contactless smartcard promises to be a typical instance of the RFID technology, e.g. close-coupling cards (ISO/IEC 10536), proximity cards (ISO/IEC 14443), and vicinity cards (ISO/IEC 15693) [2]. Contactless smartcards often show more powerful processing ability and sufficient storage capacity than RFID tags, which benefits from the card architecture with a microcontroller unit and writeable memories. A Radio Frequency Universal Smart Card (RF-UCard) is a novel contactless smartcard platform with multiple chip operating systems (COS) and multiple applications environment [3]. Multiple COSes from different vendors can coexist on a single card, and additional COSes can be loaded after card issuing. In addition, multiple applications can be hosted by a single COS, and the application can be dynamic downloaded onto or unloaded from the card.

An RF-UCard system is often composed of three main components as shown in Figure 1.

- One or more RF-UCards, held by the users to identify. RF-UCards consist of three layers, the application, the operating system and the physical layers. The application and the operating system layers host multiple applications and COSes respectively. The physical layer includes a microcontroller unit, memories and the coiled antenna. RF-UCards could be either active or passive. Active cards are partly or fully battery powered, have the capability to communicate with other cards, and can initiate a request to the reader. Passive cards, on the other hand, do not have any internal power source but are powered up by the reader.
- One or more readers, made up of a control unit and an RF module. Its main functions are to activate the cards, initiate the communication with the cards, collect the card responses, and transfer data between the back-end server and a card. The reader is usually equipped with a single COS, and could be either mono-functional or multi-functional. The mono-functional reader merely supports the single application, and no the third party is involved in the communication. The multi-functional reader contains several independent applications, but

only one application can be activated by the user at the beginning of the communication.

- A back-end server, which contains various information about RF-UCards and applications.

The reader and RF-UCards communicate over a shared wireless channel. A read process is initiated by the reader that uses radio to broadcast periodically a request command to the RF-UCards. Each valid card¹ within the reader's read field sends its ID to the reader while it receives a request. If only one card responds, the reader can successfully receive the card's ID. When more than one card responds simultaneously, messages from cards will collide and cancel each other out at the reader. This problem is referred to as the "multi-card collision", which is very similar to the "multi-tag collision" in RFID systems. Collisions can defer the transmission delay and lower the identification efficiency and cards often lose their usefulness. Hence, an anti-collision algorithm needs to be devised between the reader and the cards to minimize collisions.

The well known algorithms devised to resolve the multi-tag collision problem in RFID systems can be grouped into two broad types, namely deterministic algorithms and stochastic algorithms [1]. Deterministic algorithms resolve collisions by splitting a set of colliding tags into two subsets and attempt to recognize the subsets one by one. The typical instances of deterministic algorithms are the binary tree algorithm [4-6] and the query tree algorithm [7,8]. Stochastic algorithms are usually based on an ALOHA-like protocol in which the tags send their data at a random time period. The ALOHA-based algorithms include pure ALOHA [9], slotted ALOHA [10], static framed ALOHA [11], and dynamic framed ALOHA [12,13].

An RF-UCard system is much different from an RFID system in identification though they both communicate over a radio channel. In an RFID system, all tags within the read field will send back their responses. However, due to the fact that cards are in general equipped with multiple COSes and applications, whereas the reader always hosts the single COS, only the valid cards will send back their responses in an RF-UCard system. All cards will perform a validity check after receiving a request command. In addition, tags and cards are somewhat different in the arrival mode. Tags are usually attached to the objects and arrive at the read field in a batch mode (e.g. in a supply chain), whereas cards are often held by users and arrive in a single mode. Furthermore,



Figure 1. RF-UCard system architecture.

¹The valid card is the card that the reader's COS has been properly loaded onto it.

most ALOHA-based algorithms applied in RFID systems assume the scenario for tag identification is static, i.e. a set of tags enter the read field and stay there until all tags are identified. No new tag arrives during the identification process. Unfortunately, this scenario is not suitable for RF-UCard systems that the card quantity is dynamic changed since the card arrival occurs randomly. Finally, anti-collision algorithms applied in RFID systems mainly focus on the tag identification but do not care what further to do after the tags have been identified. However, in an RF-UCard system, the anti-collision algorithm needs not only to identify cards, but to process cards. Thus the read time in an RF-UCard system can be divided into two parts: the identification time (i.e. the time needs to identify a card) and the processing time (i.e. the time needs to execute a specific application). Hence, in order to apply the idea involved in ALOHA based algorithms to RF-UCard multiple accesses, the algorithms need to be revised according to the characteristics of an RF-UCard system.

We propose a combinatory anti-collision algorithm, called *synchronous dynamic adjusting algorithm (SDA)* for multi-card collision resolution in RF-UCard systems. SDA employs a two-sided synchronous adjusting scheme that can synchronize to adjust the frame size in the reader side and the response probability in the card sides. We focus our attention on adjusting the frame size and the card response probability by exploiting information obtained from the last read cycle. The estimated card quantity and the new arriving cards in the current read cycle are both taken into consideration to adjust the frame size for the next read cycle. The card response probability changes according to the request commands sent from the reader. These adjusting schemes reduce the collisions and as a result can facilitate card identification with shorter delay and better efficiency. Simulation results show that SDA suppresses the occurrence of collisions and shortens the total read time and delay time while preserving better identification efficiency.

The rest of the paper is organized as follows. Section 2 reviews existing ALOHA based anti-collision algorithms and Section 3 reviews some mathematical tools used for the SDA design. Section 4 gives a detail description to the synchronous dynamic adjusting scheme and the procedure of multi-card collision resolution using SDA. The extensive simulations are conducted in Section 5 to show the performance of SDA versus different parameters, and to further compare SDA with two ALOHA based algorithms. Finally, the conclusions of our analysis are presented in Section 6.

2. ALOHA Based Anti-Collision Algorithms

ALOHA based anti-collision algorithms reduce the occurrence probability of tag collisions since tags transmit at the distinct time. In pure ALOHA, tags randomly select their transmission time and, in slotted ALOHA, tags are limited to transmit only at the beginning

of a time slot with a certain time period. In framed ALOHA, the reader sends the frame size and a tag randomly selects a slot number in the frame for the data transmission. Static framed ALOHA uses a fixed frame size and does not change the size during the tag identification process. On the other hand, dynamic framed ALOHA improves the identification efficiency by dynamically changing the frame size according to the amount of tag responses in the previous read cycle. We here give detailed descriptions about two typical ALOHA based anti-collision algorithms applied in RFID systems.

2.1. Dynamic Framed Slotted ALOHA Algorithm

The dynamic framed slotted ALOHA (DFSA) has been studied extensively and shows the best performance of ALOHA based algorithms. In DFSA, the reader always broadcasts a request command at the beginning of a frame to the tags within the read field, and then waits a certain amount of time slots for tag responses. Tags randomly select a time slot in the frame to send back their IDs. Within a read cycle, the reader can collect the information about the number of the empty slots, the slots occupied by one tag, and the slots occupied by more than one tag. The slot that occupied by one tag means a tag has been successful identified, and the slot that occupied by more than one tag means a collision occurs. One more read cycles needed if collisions occur. For each read cycle, the reader dynamically adjusts the frame size according to the amount of tag responses in the previous cycle. The analysis of DFSA algorithms mainly pays attentions to two primary issues [14]. The first one is how to estimate the tag quantity within the read field according to the responses in the past. The other one is how to determine an optimal frame size for the next read cycle to achieve maximum efficiency. Recently, many researchers focus on these key issues to improve the overall performance of DFSA. As a result, some valuable methods to estimate the tag quantity and to adjust the frame size are proposed [13–17]. Results revealed that the efficiency of DFSA is very dependent on the initial frame size and the maximum efficiency occurs when the frame size equals the number of tags.

2.2. Bi-directional Binary Exponential Index Algorithm

The Bi-directional Binary Exponential Index (BBEI) algorithm [18] is originally inspired from the binary exponential backoff algorithm, which is commonly used to schedule retransmissions after collisions in Ethernet networks [19]. BBEI is based on a slotted ALOHA. Unlike DFSA, it assumes that all tags within the read field respond with a certain probability $p(0 < p \leq 1)$ ².

²Each tag has an initial response probability when it enters the reader's read field.

The reader broadcasts one of three request commands (e.g. EMPTY, SUCCESS, and FAIL) at the beginning of each time slot according to the response results in the previous slot (if the slot is idle, the command is EMPTY; if the slot is occupied by one tag, the command is SUCCESS; and if the slot is occupied by more than one tag, the command is FAIL). While a tag receives the request command, it first adjusts its response probability according to the command type and then sends back its response with the newly probability. The available values for the response probability are the inverses of the binary exponential values, such as $\{1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512, 1/1024\}$. If the received command is EMPTY, the tags multiply its probability by two; if the command is FAIL, the tags divide its probability by two; others, the probability keeps no change.

3. Mathematical Basis

This section reviews some mathematical tools we will use in subsequent sections. The read time is divided into discrete intervals (slots) with fixed length. The duration of a slot is sufficient for a card to send back its response. The number of time slots that the reader needs to wait after broadcasting a request command to cards is called "frame size" and will be denoted by N . The number of cards is usually denoted by n .

3.1. Poisson Process

The Poisson process is a continuous-time counting process which is memoryless and orderly. It applies to many cases where a certain event occurs at different points in time. Consider the card arrivals in a real-world RF-UCard system also occur at different points in time, and are often independent of each other. Thus it is reasonable to assume that the sequence of card arrivals in an RF-UCard system is a Poisson process with the arrival rate λ . Let t_i be the i th arrival time and $\tau_i = t_i - t_{i-1}$ be the i th interarrival time. Owing to the properties of the Poisson process, $\{\tau_i, i \geq 1\}$ is a sequence of independent exponentially distributed random variables with the same distribution $F(t) = 1 - e^{-\lambda t}, t \geq 0$ and the probability density function (pdf) $f(t) = \lambda e^{-\lambda t}, t \geq 0$. As a result, the mean interarrival time can be given by

$$E(\tau) = \frac{1}{\lambda} \quad (1)$$

Let $\{n(t), t \geq 0\}$ be the number of cards that arrived within the interval $(0, t)$, the probability distribution of $n(t)$ depends only on the length of the interval, and can be expressed as

$$P\{(N(t+s) - N(s)) = k\} = P\{N(t) = k\} = \frac{(\lambda t)^k e^{-\lambda t}}{k!}, t \geq 0 \quad (2)$$

Furthermore, we can obtain the expected number of card arrivals within a given interval based on the probability distribution of $n(t)$, that is

$$E(N(t)) = \sum_{k=0}^{\infty} k \cdot P\{N(t) = k\} = \sum_{k=0}^{\infty} k \cdot \frac{(\lambda t)^k e^{-\lambda t}}{k!} = \lambda t \quad (3)$$

3.2. Occupancy Problem

The allocation of cards to time slots within a frame is equivalent to the well known occupancy problem [20] that deals with the random allocation of balls to a number of bins where one is, e.g., interested in the number of filled bins. In the following, we will substitute “balls” and “bins” with “cards” and “slots”.

Given N slots and n cards, the number k of cards in one slot is binomially distributed with parameters n and $\frac{1}{N}$:

$$B_{n, \frac{1}{N}}(k) = \binom{n}{k} \left(\frac{1}{N}\right)^k \left(1 - \frac{1}{N}\right)^{n-k} \quad (4)$$

The number k of cards in a particular slot is called the occupancy number of the slot. The distribution (4) can apply to all N slots, thus the expected value of the number of slots with occupancy number k is given by $a_k^{N,n}$

$$a_k^{N,n} = N B_{n, \frac{1}{N}}(k) = N \binom{n}{k} \left(\frac{1}{N}\right)^k \left(1 - \frac{1}{N}\right)^{n-k} \quad (5)$$

This is a crucial equation because we will use it to estimate the card quantity in SDA and DFSA.

4. Synchronous Dynamic Adjusting Algorithm

The DFSA algorithm only changes the frame size according to the estimated tag quantity to improve the tag identification efficiency. However, as the number of tags becomes much larger than the frame size, the occurrence of tag collisions increases rapidly. This is mainly due to the fact that all tags within the read field send back their responses with the probability 1. On the other hand, the BBEI algorithm merely resorts to changing the tag response probability to reduce collisions. Unfortunately, due to its single-slot property, the total identification time will increase sharply as the number of tags increases. SDA is developed by integrating the ideas involved in DFSA and BBEI that adjusts the frame size and the response probability synchronously. This combinatory scheme will improve the system efficiency by reducing the card collisions. In this section, we give a detail description to the employed adjusting schemes and the procedure of collision resolution using SDA.

4.1. Programming Interface

The programming interface of SDA is both provided by the reader and the RF-UCards. It comprises some communication commands, functions, and local variables, described below.

- *Req-COS*(CID, N, f_T): the command is sent by the reader at the beginning of a frame to initiate the communication with cards within the read field. Three parameters are included that CID, N , and f_T , denote the unique identifier of the reader’s COS, the current frame size, and the traffic intensity of the system in the current read cycle, respectively.
- *Req-app* (UID, AID): the command is sent by the reader to the identified card to execute the specific application. The parameters, UID and AID , denote the card’s identifier and the unique identifier of the application respectively.
- *Res*(UID): the command is sent by a valid card to the reader with its unique identifier (UID).
- *Valid-check*(CID): the function is performed by each card in the read field to validate itself to the reader with reader’s CID .
- *Slot-select* (N): the function is performed by a card to random select a slot from N slots in the frame.
- $\langle c_0, c_1, c_x \rangle$: a triple of numbers that quantify the slots in different states. For a given slot, there are only three possible states: empty (occupied by no card), success (occupied by one card), and collision (occupied by more than one card). c_0, c_1 and c_x denote the number of empty slots, success slots, and collision slots in a frame respectively.
- S_{UID} : a set of card UID s that have just been identified in the current frame. The set is held temporarily by the reader to send the *Req-app* (UID, AID) command.
- *ResProValues*[\cdot]: an array over which a card response probability p can range ($0 < p \leq 1$). Let k be the index of *ResProValues* so that $p = ResProValues[k]$.
- f_T : a flag representing the traffic intensity of the system in a read cycle: loose ($f_T = -1$), moderate ($f_T = 0$), and crowded ($f_T = 1$).
- f_C : a flag representing one of three possible states a card may be in during a read cycle: invalid ($f_C = -1$), sleep ($f_C = 0$), and active ($f_C = 1$).

4.2. Estimation of Card Quantity

A read process of an RF-UCard system consists of multiple continuous read cycles. A read cycle starts at the time that the reader broadcasts a request command, and ends up when the last identified card in the current frame has been processed. The length of a cycle is equal to the current frame size plus the processing times. In order to pick the appropriate frame size N for the (a priori unknown) number of cards n in the read field, we have to estimate n in each read cycle. The estimation of card quantity is a key issue involved in dynamic framed ALOHA algorithms. In SDA, we employ the estimation scheme that studied extensively in most literatures and originally proposed by H. Vogt [13]. The estimation proceeds as following steps.

Step1: Based on the mathematical basis discussed previously (recall (4) and (5)), we can compute the

expected value $\langle c_0, c_1, c_x \rangle$ with already known N and n . In a read cycle with the frame size N , the expected number of empty slots (with occupancy number 0) can be obtained by

$$a_0^{N,n} = NB_{n, \frac{1}{N}}(0) = N(1 - \frac{1}{N})^n \quad (6)$$

Also, the expected number of success slots (with occupancy number 1) can be obtained by

$$a_1^{N,n} = NB_{n, \frac{1}{N}}(1) = N \binom{n}{1} (\frac{1}{N}) (1 - \frac{1}{N})^{n-1} = n(1 - \frac{1}{N})^{n-1} \quad (7)$$

Thus, the expected number of collision slots (with occupancy number >1) is $N - a_0^{N,n} - a_1^{N,n}$. Figure 2 shows a function definition in Java to obtain the expected value of $\langle c_0, c_1, c_x \rangle$ and Table 1 shows some useful expected values that derived from the function *getSlotCount* with given certain N and n . For an extensive experiment, we will compute more expected values of $\langle c_0, c_1, c_x \rangle$ by more possible N and n . Then make an expected value table similar to Table 1.

```
void getSlotCount(int N, int n, double c0, double c1,
double cx)
{
    c0 = N * Math.pow((1-(1.0/N)), n);
    c1 = n * Math.pow((1-(1.0/N)), (n-1));
    cx = N - c0 - c1;
}
```

Figure 2. The function to compute the expected values of $\langle c_0, c_1, c_x \rangle$.

Step2: In each read cycle, the reader will get a read value of $\langle c_0, c_1, c_x \rangle$. The Chebyshev's inequality tells us that the outcome of a random experiment involving a random variable X is most likely somewhere near the expected value of X . Thus we use the distance between the read value and the expected value of $\langle c_0, c_1, c_x \rangle$ to estimate the number of cards n for which the distance becomes minimal. The estimation function denoted by ε_{vd} is defined as

$$\varepsilon_{vd}(N, c0, c1, cx) = \min_n \left| \begin{pmatrix} a_0^{N,n} \\ a_1^{N,n} \\ a_{\geq 2}^{N,n} \end{pmatrix} - \begin{pmatrix} c0 \\ c1 \\ cx \end{pmatrix} \right| \quad (8)$$

Because the current frame size N is always known, after getting a read value of $\langle c_0, c_1, c_x \rangle$, we can compare the read value with the expected value table. According to (8), we can obtain the estimated card quantity n finally.

4.3. Adjusting the Frame Size

The variation of the frame size takes large impacts on the performance of dynamic framed ALOHA algorithms [14, 17]. Results revealed that the maximum performance occurs when the frame size equals the number of tags. However, this result is not suitable again to the collision

resolution in an RF-UCard system. Recall that the card arrivals are a Poisson process, the card quantity within the read field is dynamically changed. The estimation scheme proposed in Section 4.2 just reflects the card quantity within the read field at the beginning of a read cycle while not including the new cards arrived in the current read cycle. Thus in SDA, we will employ a novel adjusting scheme in terms of the estimated card quantity and the new arriving cards to choose an optimal frame size for the next read cycle. Based on the card quantity estimation scheme and (3), the new frame size N^* can be obtained by

$$N^* = (n - c_1) + p \cdot \lambda N \quad (9)$$

where c_1 and p denote the number of identified cards in the current cycle and the initial response probability of cards respectively. Again, λN denotes the expected number of arriving cards in the current cycle (see(3)), and then $p \cdot \lambda N$ denotes the new arriving cards that really respond in the next cycle. In other words, (9) shows the idea that letting the new frame size N^* to be the number of the cards that will respond in the next cycle.

4.4. Adjusting the Response Probability

In BBFI, the tag response probability is changed within a range of the inverses of the binary exponential values. As we all know that a variable with an exponential increment shows the sharp deviations, especially when the variable becomes very large. Thus we limits the card response probability in SDA to be within a range of values with the linear increment, that is *ResProValues* = {1.0/8, 7.0/8, 6.0/8, 5.0/8, 4.0/8, 3.0/8, 2.0/8, 1.0/8}. We also assume that all cards have the same initial response probability $p = \text{ResProValues}[k]$, e.g. if $k=4$, $p=0.5$, and this initial value is also held by the reader.

The adjusting scheme for the card response probability in SDA is based on the previous read results (i.e. the traffic intensity f_T of the system in the previous read cycle) and can be defined as a Java function shown in Figure 3.

```
double getResPro(int k, int fT)
{
    if(fT == -1)
        k--;
    else if(fT == 1)
        k++;
    if(k < 0)
        k = 0;
    else if(k >= ResProValues.length)
        k = ResProValues.length - 1;
    return ResProValues[k];
}
```

Figure 3. The function to compute the card response probability.

4.5. Procedure of SDA Algorithm

Table 1. Some typical expected values of $\langle c_0, c_1, c_x \rangle$ by given certain N and n .

Frame size	5 cards			10 cards			15 cards		
	c_0	c_1	c_x	c_0	c_1	c_x	c_0	c_1	c_x
6	2.4113	2.4113	1.1774	0.969	1.9381	3.0929	0.3894	1.1683	4.4423
7	3.2387	2.6989	1.0624	1.4984	2.4973	3.0043	0.6933	1.7332	4.5735
8	4.1033	2.9309	0.9658	2.1046	3.0066	2.8888	1.0795	2.3132	4.6073
9	4.9944	3.1215	0.8841	2.7715	3.4644	2.7641	1.538	2.8837	4.5783
10	5.9049	3.2805	0.8146	3.4868	3.8742	2.639	2.0589	3.4315	4.5096
Frame size	20 cards			25 cards			30 cards		
	c_0	c_1	c_x	c_0	c_1	c_x	c_0	c_1	c_x
6	0.1565	0.626	5.2175	0.0629	0.3145	5.6226	0.0253	0.1517	5.823
7	0.3207	1.0692	5.6101	0.1484	0.6183	6.2333	0.0687	0.3433	6.588
8	0.5537	1.5819	5.8644	0.284	1.0142	6.7018	0.1457	0.6242	7.2301
9	0.8535	2.1337	6.0128	0.4736	1.4801	7.0463	0.2628	0.9856	7.7516
10	1.2158	2.7017	6.0825	0.7179	1.9942	7.2879	0.4239	1.413	8.1631

A read cycle in SDA proceeds as following five steps.

Step1: The reader initiates a read cycle by broadcasting $Req-COS(CID, N, f_T)$ to all cards within the reader's read field. Because a read process begins with the reader broadcasts the first request command (the time is denoted by 0), no card has arrived at that time and the first read cycle is always wasted. Thus it is reasonable to set the initial frame size to 1 in SDA. Moreover, the initial value of f_T is often set to 0.

Step2: After receiving $Req-COS(CID, N, f_T)$, all cards within the read field perform *Valid-check* (CID) to validate itself to the reader with the reader CID derived from $Req-COS(CID, N, f_T)$. If a card is invalid, it will set itself to the state *invalid* ($f_C = -1$) and then exits the following cycles permanently. Otherwise the cycle proceeds to the next step.

Step3: The valid cards first perform *Slot-select* (N) to generate a random number s uniformly distributed within the range from 0 to $N-1$. Then all valid cards adjust their response probability p by performing $getResPro(k, f_T)$ based on the value of f_T received from the reader. Finally, the cards send back $Res(UID)$ at the s th slot with the newly probability p .

Step4: The reader checks the slot states in the current frame in sequence. If a slot is successful, a card is identified and its UID is appended to S_{UID} for the later processes. After state checking, the reader can observe c_0 empty slots, c_1 successful slots, and c_x collision slots,

where $c_0+c_1+c_x = N$.

Step5: The reader sequential takes the UID from S_{UID} as a parameter to send $Req-app(UID, AID)$ to execute the specific application with the identified card. This step is also called the card processing step. After finishing the process, the card sets itself to the state *sleep* ($f_C = 0$) so that it cannot respond in the following cycles. The read cycle ends up with the finish of processing of all identified cards.

After finishing a read cycle, SDA will perform a synchronous adjusting scheme to optimize the frame size and the card response probability for the next read cycle. The adjusting proceeds as following two steps.

Step1: The reader first performs the card quantity estimation proposed in Section 4.2 to obtain an estimated card number n . Then, the adjusting scheme proposed in Section 4.3 is performed to choose an optimal frame size for the next read cycle.

Step2: The reader also adjusts the traffic intensity flag f_T according to the relationship between c_0 and c_x . Note that a collision slot means it is occupied by at least two cards, thus we develop an adjusting scheme for f_T as: if $c_0 > 2 \cdot c_x$, then $f_T = -1$; if $c_0 < 2 \cdot c_x$, then $f_T = 1$; else $f_T = 0$. This step aims to refresh the value f_T to inform the card to change its response probability in the next cycle.

An example provided in Figure 4 illustrates the read process of SDA. In this example, we set the card arrival rate to 2 and the card response probability to 0.5. We

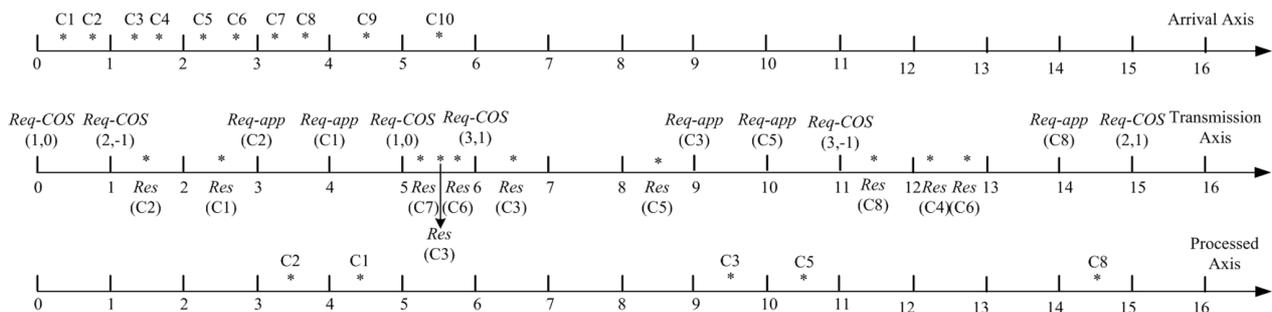


Figure 4. An example of the SDA algorithm.

assume that all arriving cards are valid and execute the same application. Thus the parameters *CID* and *AID* can be omitted in *Req-COS()* and *Req-app()* respectively. There are three time axes: the arrival axis, the transmission axis, and the processed axis. Each time axis is divided into equal slots with fixed length. The arrival axis shows the arrival time of card C1, C2, ..., C10 denoted by the symbol “*”. A sequence of *Req-COS(N, f_T)* and *Req-app (UID)* are shown above the transmission axis (note that all commands are sent at the beginning of a slot), while a sequence of *Res (UID)* are shown below the transmission axis (denoted by the symbol “*”). The successful processed cards are shown above the processed axis (also denoted by the symbol “*”). At $t = 0$, the read process begins and the reader broadcasts *Req-COS(1, 0)*. However, no card arrives at this time and the reader obtains a triple, $\langle 1, 0, 0 \rangle$. According to the adjusting schemes for the frame size and the response probability in SDA, the newly parameters $N=2, f_T=-1$ and $p=0.625$ are derived. The reader initiates the 2nd cycle at $t=1$ with *Req-COS(2, -1)*. At this time, C1 and C2 have arrived and respond in slot 3 and slot 2 respectively. Thus C2 and C1 are identified and then processed in slot 4 and slot 5 sequentially. Again the reader obtains a triple, $\langle 0, 2, 0 \rangle$ and the newly parameters $N=1, f_T=0$, and $p=0.625$ for the 3rd cycle. In the meanwhile of the 2nd cycle, there are other 7 new arriving cards. The read process will continue in this way and the read data derived from the former 5 cycles are shown in Table 2.

5. Simulation and Evaluation

We develop a Java program based on the Eclipse platform to simulate the process of SDA. Our simulations are based on the following scenarios.

- All arriving cards are valid, which is the worst case may be occurred in a real-world RF-UCard system.
- The card arrival follows a Poisson process, and the arrival rate varies from 0.1 to 0.9 with a step of 0.1 and from 1 to 10 with a step of 1.
- The card quantity is finite that enables the simulation to be finished normally and the simulated card set varies from 10 to 200 cards with a step of 10.
- All experiments are simulated 100 times in order to ensure the convergence of simulation results.

5.1. Performance Measures

To evaluate the performance of SDA and other ALOHA-based anti-collision algorithms, we mainly consider the following measures.

- Total read time: this metric is the total time required to identify and process all the cards inside the reader’s read field. We measure the time by the timeslot because each of three mentioned algorithms (SDA, DFSA and BBEI) has a time period for carrying both the reader-to-card signals and the card-to-reader

signals.

- Card delay time: this metric is the average number of timeslots waited by a card in the entire read process. It also reflects the mean sojourn time of each card in the system before being processed.
- Identification efficiency: this metric is the mean number of cards being identified in a timeslot. It equals to the ratio of the total identified cards to the sum of the frame size.
- The number of collisions: this metric is the total number of collision timeslots between the card-to-reader responses. Collisions increase the read time and thus lower the identification efficiency.
- The number of empty timeslots: this metric is equals to the sum of the empty timeslots in each read cycle. More empty timeslots waste more read time and thus also lower the identification efficiency.

5.2. Card Arrival Simulation

We first simulate the card arrivals which follow a Poisson process while varying the arrival rate λ . The theoretical values of the interarrival times for each λ can be obtained from (1). We take 100 cards into considerations to compute the simulated values of the interarrival times. Let T_λ and S_λ denote the theoretical value and the simulated value of the interarrival times respectively. The relative error between S_λ and T_λ denoted by ε_λ can be derived from

$$\varepsilon_\lambda = (S_\lambda - T_\lambda) / T_\lambda \quad (10)$$

Table 3 shows the simulated results when λ varying from 0.2 to 6. The results present that the relative errors are rather little for all λ values, in other words, the theoretical values and the simulated values are in good agreement.

5.3. Performance Evaluation

We then evaluate the impact of the system parameters on the performance of SDA. Three parameters, i.e. the card arrival rate, the card initial response probability and the card quantity, are considered. Although the variation of each of these three parameters will influence the performance, we mainly focus on the impact of the single parameter. Thus we conduct the independent experiments for each parameter under the different scenarios.

Table 2. The read data derived from the example.

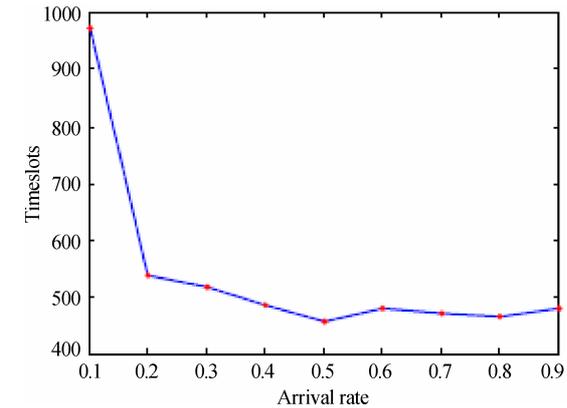
Cycle	1	2	3	4	5
N	1	2	1	3	3
f_T	0	-1	0	1	-1
p	0.5	0.625	0.625	0.5	0.625
c_0	1	0	0	1	1
c_1	0	2	0	2	1
c_x	0	0	1	0	1
S_{UID}	{}	{C2,C1}	{}	{C3, C5}	{C8}

- For the card arrival rate, we set the card quantity n to 100 and the initial response probability p to 0.5.
- For the card initial response probability, we set the card quantity n to 100 and the arrival rate λ to 1.
- For the card quantity, we set the arrival rate λ to 1 and the initial response probability p to 0.5.

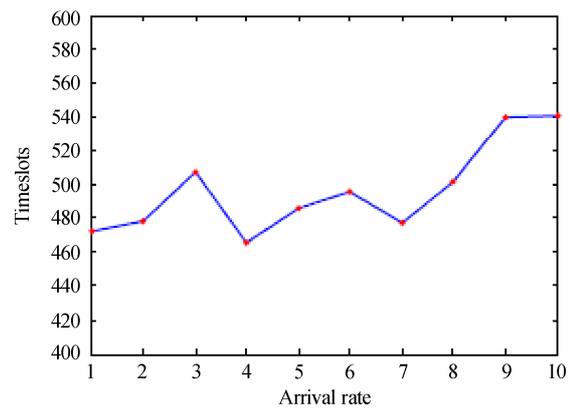
Figure 5 shows the simulation results about the impact of the card arrival rate λ . From Figure 5(a) and (b), we can see that fewer read timeslots required by SDA to complete a read process when λ varies within the range from 0.5 to 2, and the minimal value occurs at $\lambda=0.5$. When λ is less than 0.5, more timeslots required,

Table 3. The mean interarrival times of card arrivals ($n=100$).

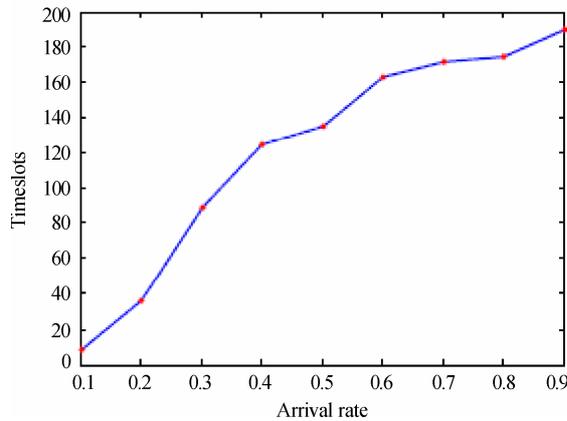
λ	T_λ	S_λ	ϵ_λ
0.2	5	5.0436	0.0087
0.4	2.5	2.5052	0.0021
0.6	1.6667	1.6843	0.0106
0.8	1.25	1.2614	0.0091
1	1	0.9878	-0.0122
2	0.5	0.4984	-0.0032
3	0.3333	0.3343	0.003
4	0.25	0.2534	0.0136
5	0.2	0.1974	-0.013
6	0.1667	0.1641	-0.0156



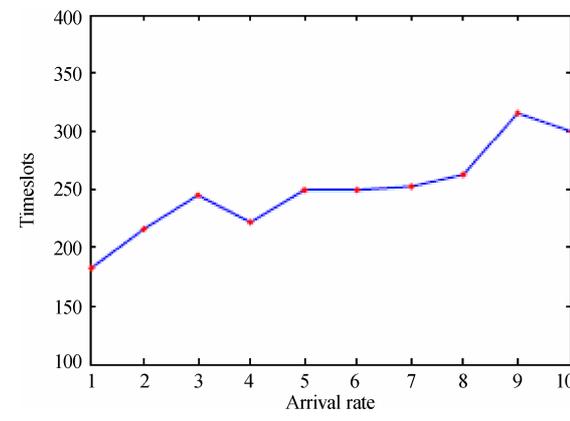
(a) Total read time ($\lambda < 1, p = 0.5, n = 100$)



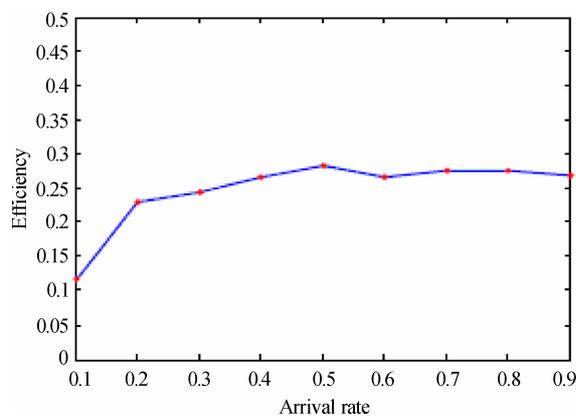
(b) Total read time ($\lambda \geq 1, p = 0.5, n = 100$)



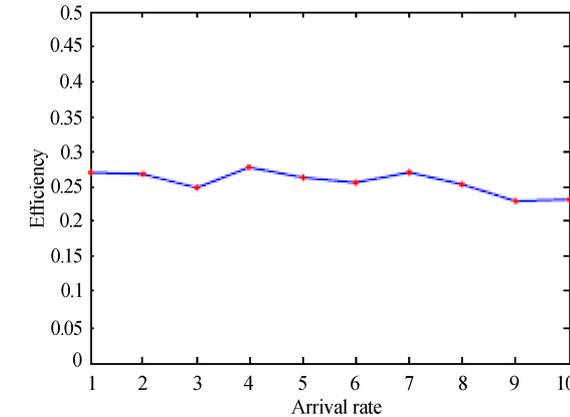
(c) Card delay time ($\lambda < 1, p = 0.5, n = 100$)



(d) Card delay time ($\lambda \geq 1, p = 0.5, n = 100$)



(e) Identification efficiency ($\lambda < 1, p = 0.5, n = 100$)

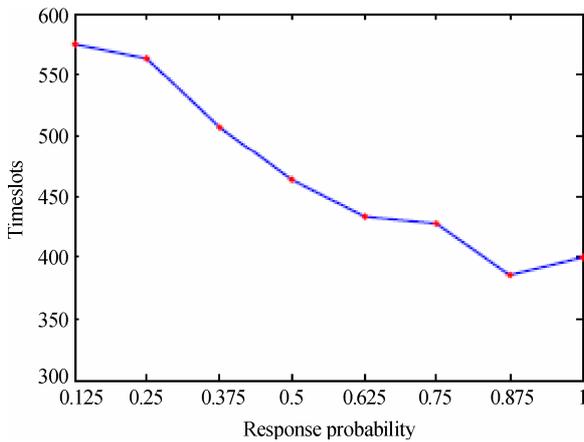


(f) Identification efficiency ($\lambda \geq 1, p = 0.5, n = 100$)

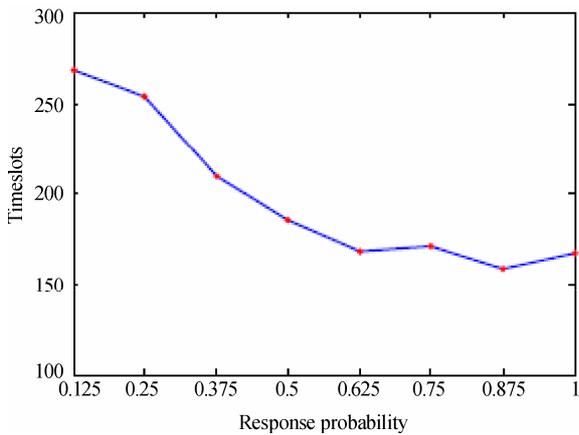
Figure 5. Impact of card arrival rate.

especially the maximal value occurs at $\lambda=0.1$. This is mainly due to the fact that the smaller arrival rate means the larger interarrival time and thus increases the number of empty timeslots in each read cycle. On the other hand, as λ increases, the read process also becomes longer because of the larger arrival rate will enlarge the card quantity rapidly and thus increases card collisions in each read cycle. Figure (c) and (d) illustrate that the card delay time gets longer as λ increases. This is due to the fact that the

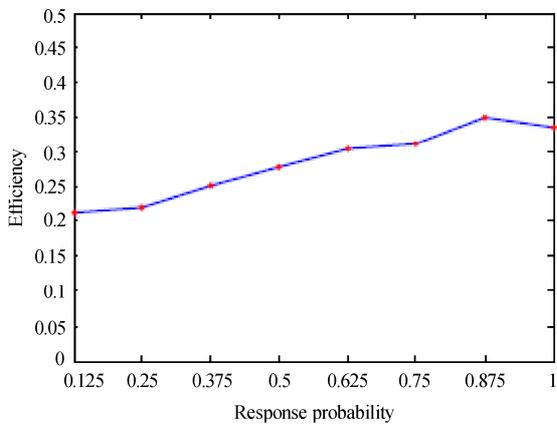
larger arrival rate means the earlier arrival time for each card, while the longer read time further extends the card delay time. The impact of the arrival rate on identification efficiency shown in Figure 5(e) and (f) is very similar to the case of the read time shown in Figure 5(a) and (b). SDA achieves better identification efficiency when λ varies within the range from 0.5 to 2, and the maximal efficiency occurs at $\lambda=0.5$. As λ has larger distance from this range, the lower efficiency occurs. However, the



(a) Total read time ($\lambda = 1, n = 100$)

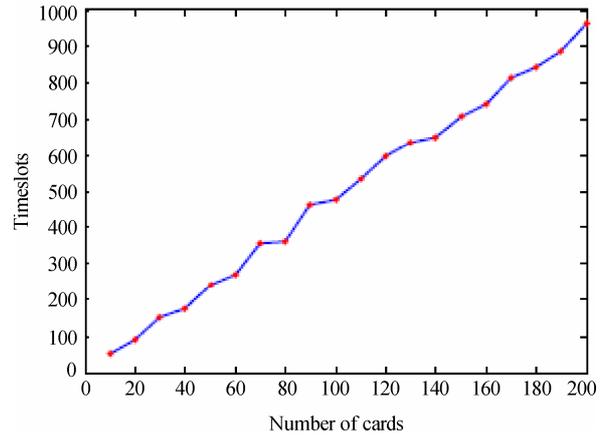


(b) Card delay time ($\lambda = 1, n = 100$)

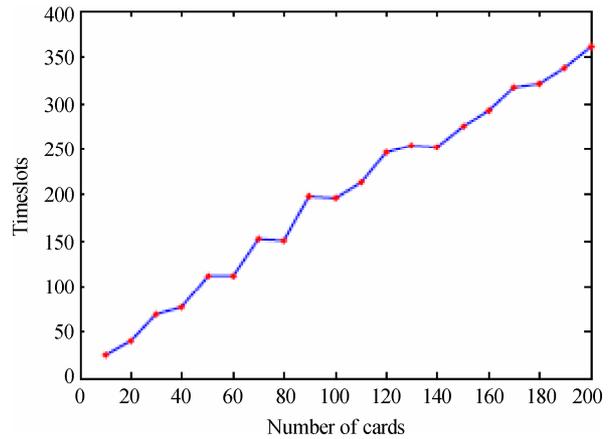


(c) Identification efficiency ($\lambda = 1, n = 100$)

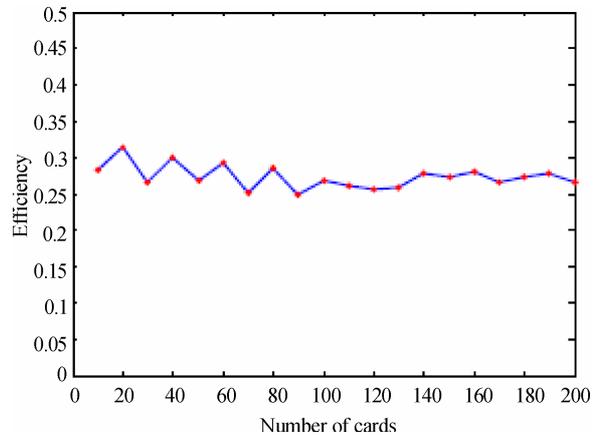
Figure 6. Impact of card initial response probability.



(a) Total read time ($\lambda = 1, p = 0.5$)



(b) Card delay time ($\lambda = 1, p = 0.5$)



(c) Identification efficiency ($\lambda = 1, p = 0.5$)

Figure 7. Impact of card quantity.

deviation of the identification efficiency is rather small for all λ values except 0.1, and the efficiency varies around 0.26.

Figure 6 shows the simulation results about the impact of the card initial response probability p . From Figure 6(a) and (b), we can see that both the total read time and the card delay time get shorter as p increases, especially the shortest time occurs at $p = 0.875$. Also, Figure 6(c) presents that SDA shows the better identification efficiency as p increases. The optimal efficiency achieved when $p = 0.875$ and is close to 0.35. These results are due to the fact that the larger response probability will decrease the number of empty timeslots rapidly as well as not increase card collisions obviously in each read cycle when the arrival rate is 1.0.

Figure 7 shows the simulation results about the impact of the card quantity n . From Figure 7(a) and (b), we can see that both the read time and the card delay time increase linearly with n . However, the read time grows by the larger incremental ratio than the delay time that the former is more than 4.5 times the card quantity while the latter is close to 2 times the card quantity. For example, if there are 100 cards, the required total read

time and card delay time are about 450 and 200 timeslots respectively.

5.4. Performance Comparisons

In order to compare the performance of SDA to DFSA and BBEI, we further develop Java programs to simulate DFSA and BBEI in an RF-UCard system. For DFSA, we use the estimation function ε_{vd} which defined in (8) to estimate the card quantity in the current read cycle. In addition, we set the frame size to be the estimated card quantity to obtain an optimal frame size for the next read cycle. For each algorithm, we set the card quantity to 100 while varying the card arrival rate, whereas we set the arrival rate to 1 while varying the card quantity. In order to carry out an extensive comparison, we take the optimal performance achieved by three algorithms into comparisons under a given simulation scenario.

Table 4 and Figure 8 depict the simulation results while varying the card arrival rate. From Table 4, we find that for each algorithm, collision timeslots are always less than empty timeslots in a read process. However, SDA generates the minimal collision timeslots

Table 4. The average number of timeslots with varying the arrival rate ($n=100$).

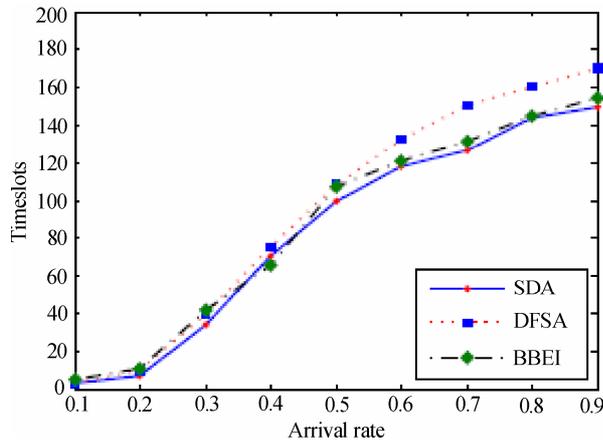
Arrival rate	Algorithm	Collision timeslots	Empty timeslots	Total timeslots
0.2	SDA	41.3	264.1	505.4
	DFSA	55.1	228.8	483.9
	BBEI	50.6	211	461.6
0.4	SDA	72.7	111.5	384.2
	DFSA	75.5	117.2	392.7
	BBEI	81.9	108.2	390.1
0.6	SDA	71.8	123.8	395.6
	DFSA	78.4	129.7	408.1
	BBEI	97.8	102.7	400.5
0.8	SDA	66	103.8	369.8
	DFSA	77.3	129.1	406.4
	BBEI	92.2	102.5	394.7
1	SDA	71.5	120.9	392.4
	DFSA	75.8	141	416.8
	BBEI	98.7	105.5	404.2
2	SDA	69.1	126	395.1
	DFSA	74.2	147.6	421.8
	BBEI	97.7	107.8	405.5
3	SDA	65.2	147.2	412.4
	DFSA	76.2	150.5	426.7
	BBEI	98	118.4	416.4
4	SDA	69.6	125.4	395
	DFSA	78.2	151.1	429.3
	BBEI	98.1	109.8	407.9
5	SDA	68.3	131.4	399.7
	DFSA	78.4	145.5	423.9
	BBEI	99.7	101.4	401.1
6	SDA	70	155.4	425.4
	DFSA	76.9	165.4	442.3
	BBEI	89.4	144.8	434.2

Table 5. The average number of timeslots with varying the card quantity ($\lambda=1$).

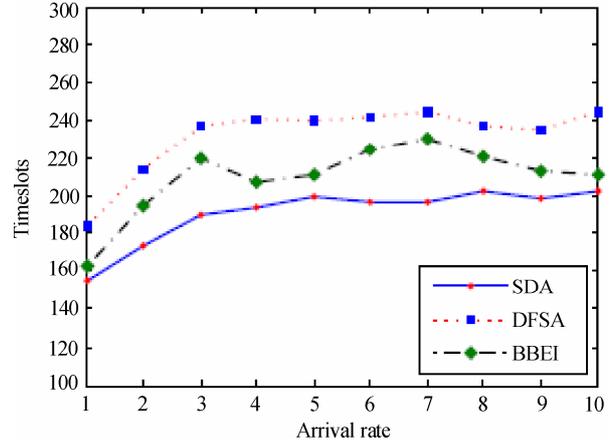
Card quantity	Algorithm	Collision timeslots	Empty timeslots	Total timeslots
20	SDA	13.7	25.5	79.2
	DFSA	12.9	31.4	84.3
	BBEI	15.3	18.8	74.1
40	SDA	26.2	53.3	159.5
	DFSA	30	58	168
	BBEI	37.7	37.6	155.3
60	SDA	42.6	76.4	239
	DFSA	46.5	80.8	247.3
	BBEI	55.5	63.5	239
80	SDA	55	97.1	312.1
	DFSA	59.3	111	330.3
	BBEI	72	92.2	324.2
100	SDA	71.5	120.9	392.4
	DFSA	75.8	141	416.8
	BBEI	98.7	105.5	404.2
120	SDA	92	142.7	474.7
	DFSA	102	170.8	512.8
	BBEI	124.9	127.2	492.1
140	SDA	107.3	169	556.3
	DFSA	112.5	182.8	575.3
	BBEI	129	153.7	562.7
160	SDA	123.4	213.2	656.6
	DFSA	126.9	219.7	666.6
	BBEI	169.7	173	662.7
180	SDA	130.4	219.6	710
	DFSA	145.3	229	734.3
	BBEI	165.4	187	712.4
200	SDA	148.7	229.7	778.4
	DFSA	159.3	263.2	822.5
	BBEI	198.5	213	811.5

in all algorithms because of the synchronous dynamic adjusting scheme we employed for both the frame size and the card response probability. Due to the frame size is always set to 1 during the entire read process, BBEI generates the maximal collision timeslots as well as the minimal empty timeslots in all algorithms. SDA generates fewer empty timeslots than DFSA since the new arriving cards in each read cycle are

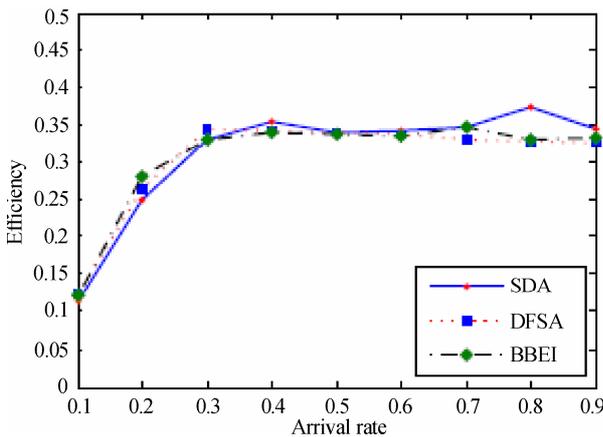
taken into consideration to compute the new frame size. As a result, SDA requires fewer total timeslots than DFSA and BBEI to complete a read process, Figure 8(a) and (b) illustrate that the card delay time of SDA is the shortest of all algorithms, and the gap of delay time between SDA and other algorithms is much larger when the arrival rate is greater than 1. Figure 8(c) and (d) illustrate that the identification efficiency of SDA is



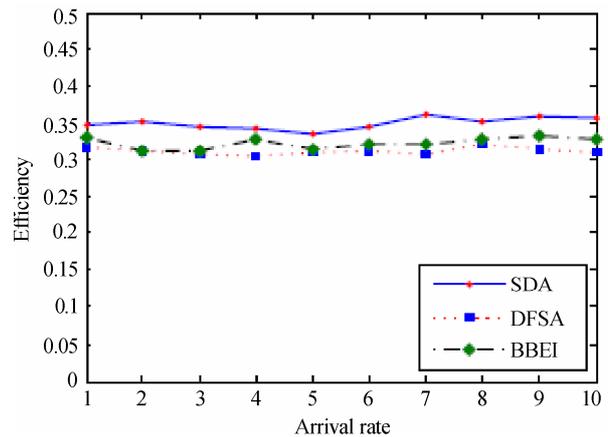
(a) Card delay time ($\lambda < 1, n = 100$)



(b) Card delay time ($\lambda \geq 1, n = 100$)

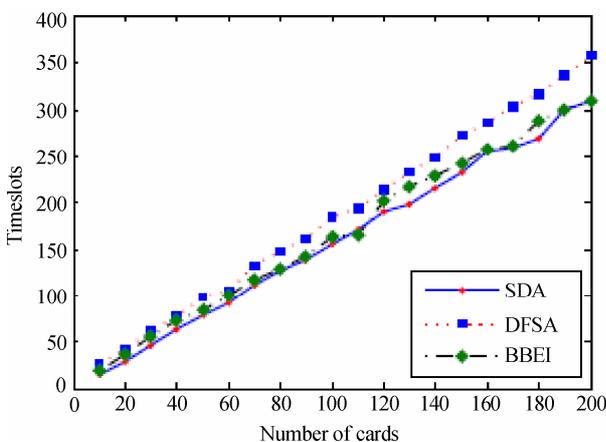


(c) Identification efficiency ($\lambda < 1, n = 100$)

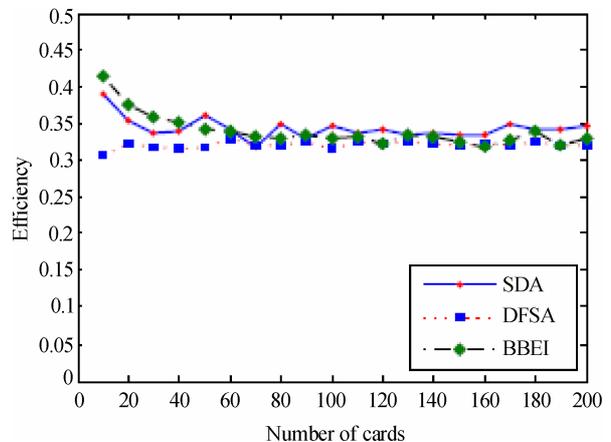


(d) Identification efficiency ($\lambda \geq 1, n = 100$)

Figure 8. Performance comparison with varying the card arrival rate.



(a) Card delay time



(b) Identification efficiency

Figure 9. Performance comparison with varying the card quantity.

the best of all algorithms at high arrival rate, while it is worse than DFSA and BBEI at arrival rate less than 0.3 because of more empty timeslots produced by larger interarrival time. Note that BBEI shows shorter delay time and better identification efficiency than DFSA at most simulation scenarios. These results show a good agreement with the results shown in Table 4 because fewer total timeslots mean better identification efficiency.

Table 5 and Figure 9 show the simulation results obtained by varying the card quantity. As the card quantity increases, the total read time and the card delay time get longer and the number of collision timeslots and empty timeslots gets larger for each algorithm. The results shown in Table 5 are very similar to the results shown in Table 4. SDA generates the minimal collision timeslots when the card quantity is greater than 40. Again, BBEI generates the maximal collision timeslots as well as the minimal empty timeslots. The total timeslots required by SDA is fewer than DFSA and BBEI when the card quantity is greater than 60. By restraining the occurrence of collisions, SDA has shorter delay time than DFSA and the gap gets larger as the card quantity increases. However, the delay time of BBEI is very close to the time of SDA because it generates fewer empty timeslots. All of algorithms show very similar identification efficiency when there are larger cards in the read field. SDA shows a little better efficiency than others when the card quantity is greater than 60. The values vary around 0.35, which is very close to 0.368, the maximal identification efficiency of the framed ALOHA applied in RFID systems.

6. Conclusions

Multi-card collision is a major factor in influencing the efficiency of an RF-UCard system. In this paper, a novel and enhanced multi-card anti-collision algorithm has been proposed and evaluated. Unlike DFSA and BBEI that they merely adjust the frame size or the tag response probability, the proposed SDA algorithm employs a synchronous dynamic adjusting scheme that dynamically adjusts the frame size in the reader and the response probability in cards to maximize the efficiency of card identification and processing. For the reader, the estimated card quantity and the new arriving cards in the current read cycle are both taken into consideration to adjust the frame size for the next read cycle. For a card, it increases or decreases its response probability according to the request commands sent from the reader. This scheme improves the card identification efficiency by reducing both the collision timeslots and the empty timeslots.

The simulation results indicate that the performance of SDA is seriously affected by the variations of the card arrival rate, the card initial response probability and the card quantity. SDA requires shorter read time and achieves better identification efficiency when the arrival rate varies within the range from 0.5 to 2. The card delay

time gets longer as the arrival rate increases while gets shorter as the card initial response probability increases. Moreover, both the read time and the card delay time increase linearly with the card quantity. A simulation based comparison shows that SDA requires shorter read time and card delay time and achieves better identification efficiency than DFSA and BBEI by significantly reducing the collision timeslots and the empty timeslots. The optimal identification efficiency of SDA varies around 0.35, which is very close to 0.368, the maximal identification efficiency of the framed ALOHA applied in RFID systems.

7. References

- [1] K. Finkenzeller, "RFID handbook," 2nd edition, John Wiley and Sons, 2003.
- [2] W. Rankl and W. Effing, "Smart card handbook," 3rd edition, John Wiley and Sons, 2004.
- [3] L. Shu, J. C. Cao, and Z. D. Lu, "CIM: Hardware support for Multi-COS isolation of RF-UCard," in Proceedings 2008 International Conference on Embedded Software and Systems, pp. 595-602, July 2008.
- [4] D. R. Hush and C. Wood, "Analysis of tree algorithm for RFID arbitration," in Proceedings IEEE International Symposium on Information Theory, pp. 107, 1998.
- [5] ISO/IEC 18000-6: Information technology-Radio frequency identification for item management, Part 6, Parameters for air interface communications at 860 MHz to 960 MHz, 2004/Amd 1: 2006.
- [6] J. Myung and W. Lee, "Adaptive binary splitting: A RFID tag collision arbitration protocol for tag identification," *Mobile Networks and Applications*, Vol. 11, No. 5, pp. 711-722, May 2006.
- [7] C. Law, K. Lee, and K. S. Siu, "Efficient memoryless protocol for tag identification," in Proceedings 4th ACM International workshop on discrete algorithms and methods for mobile computing and communications, pp.75-84, August 2000.
- [8] J. Myung and W. Lee, "Adaptive splitting protocols for RFID tag collision arbitration," in Proceedings 7th ACM international symposium on Mobile ad hoc networking and computing, pp. 202-213, 2006.
- [9] N. Abramson, "Development of the ALOHANET," *IEEE Transactions on Information Theory*, Vol. IT-31, pp. 119-123, March 1985.
- [10] L. G. Roberts, "ALOHA packet system with and without slots and capture," ARPA Network Information Center, Stanford Institute, Menlo Park California, ASS Note 8 (NIC 11290), June 1972.
- [11] F. C. Schoute, "Control of ALOHA signalling in a mobile radio trunking system," in Proceedings International Conference on Radio Spectrum Conservation Techniques, pp. 38-42, 1980.
- [12] F. C. Schoute, "Dynamic frame length ALOHA," *IEEE Transactions on Communications*, Vol. 31, No. 4, pp. 565-568, April 1983.
- [13] H. Vogt, "Efficient object identification with passive

- RFID tags,” in Proceedings International Conference on Pervasive Computing, Springer-Verlag, Vol. 2414, pp. 98–113, 2002.
- [14] W. T. Chen and G. H. Lin, “An efficient anti-collision method for tag identification in a RFID system,” *IEICE Transaction on Communications*, Vol. E89-B, No.12, pp. 3386–3392, December 2006.
- [15] J. Zhai and G. N. Wang, “An anti-collision algorithm using two-functioned estimation for RFID tags,” in Proceedings International Conference Computational Science and its Applications, Vol. 3843, pp. 702–711, May 2005.
- [16] S. R. Lee, S. D. Joo, and C. W. Lee, “An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification,” in Proceedings 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, pp. 166–172, July 2005.
- [17] B. Zhen, M. Kobayashi, and M. Shimizu, “Framed ALOHA for multiple RFID objects identification,” *IEICE Transaction on Communications*, Vol. E88-B, No. 3, pp. 991–999, March 2005.
- [18] S. S. Yu, Y. Zhan, and Y. H. Wang, “RFID anti-collision algorithm based on bi-directional binary exponential index,” in Proceedings the IEEE International Conference on Automation and Logistics, Jinan, China, pp. 2917–2921, August 2007.
- [19] S. A. Tanenbaum, “Computer Networks,” 4th edition, Prentice Hall PTR, 2002.
- [20] R. Motwani and P. Raghavan, “Randomized algorithms,” Cambridge University Press, 1995.

Performance Analysis of Sub-Rating for Handoff Calls in HCN

Xiaolong WU¹, Min HE¹, Fei WANG², Jun ZHENG³, Emma REGENTOVA⁴, Guoshun HAO⁵

¹Department of CECS, ² Department of EE, California State University, Long Beach, CA, USA

³ Department of Computer Science, New Mexico Institute of Mining and Technology, New Mexico, USA

⁴ Department of Electrical and Computer Engineering, University of Nevada, Las Vegas, Nevada, USA

⁵National Lab of Software Development Environment, Beijing University of Aeronautics and Astronautics, Beijing, China

Email: {xwu3, mhe, fwang3}@csulb.edu

Received October 3, 2008; revised December 28, 2008; accepted December 31, 2008

Abstract

Hierarchical Cellular Networks (HCN) offer more efficient channel utilization and better quality of service (QoS) under the high tele-traffic condition compared to the single-tier system. One of the important measures of QoS in HCN as in any single-tier system is the handoff dropping rate. Although the existing approaches such as guard channel and queuing can reduce forced termination probability, they also result in higher new call blocking probability. The channel sub-rating strategy has found to be an effective technique to reduce the handoff force termination probability while preserving the new call blocking probability in a single-tier system. In this paper, we propose a new call admission control scheme for HCN based on the channel sub-rating. Analytic models based on 1-D Markov process in microcell and 2-D Markov process in macrocell are developed. Experimental results show that our scheme achieves lower blocking and forced termination probabilities compared to the traditional guard channel scheme. The effect of channel sub-rating on the voice quality degradation is also studied. Results demonstrate that we can establish a good balance between the forced termination probability and the voice quality degradation by varying the number of sub-ratable full-rate channels.

Keywords: Sub-Rating, HCN, Handoff, Degradation Ratio

1. Introduction

Cellular networks has experienced a rapid growth of the number of subscribers during the past decades, which places high demands on the system capacity [1–4]. The cell size can be reduced to accommodate more mobile users in a given area and higher frequency re-uses [5]. However, chances of mutual interferences and cell-boundary crossings (*handoffs*) in small-cell systems are high [6]. The encountered handoffs can degrade one of the essential performance characteristics of cellular networks, i.e., forced termination probability—probability of dropping an active call. Handoff calls limit the call handling capacity of a cellular system due to higher delays and limited bandwidth source. One way of controlling the increase of traffic, while preserving the frequency reuse advantage of small-cell systems is to consider hierarchical cellular network (HCN) [7–8]. In HCN, cells of different sizes are organized into separate layers to provide high coverage and capacity over a given service area. For example, in the two-layer (micro/

macro) hierarchical structure [9], several small cells called microcell with a radii of a few hundred meters are organized as a microcell layer which is overlaid by a large macrocells whose radius is a few kilometers.

In HCN, upon entering the system, the mobile terminals can select the service layer based on their mobility or traffic patterns. For example, in the speed-insensitive selection mechanism [10], all new and handoff calls are first directed to the microcells; if there is no channel available in the microcells, calls can be overflowed to the macrocell. On the contrary, in the speed-sensitive strategy, the mobile terminals traveling at high speeds (fast-mobility subscribers) are normally serviced by the macrocell, whereas low-mobility subscribers are handled by the microcells [4,9–12]. Under the speed-sensitive layer selection strategy, the *overflow* operation can be used to transfer calls to the other layer if there is no sufficient resource in the current service layer [13–15].

For call admission control (CAC) in HCN, handoff call handling is particularly important because generally the forced termination of an ongoing call is considered

less desirable than blocking of a new call. Thus, handoff calls have higher priority than new calls during the call admission control process. Several approaches have been proposed for handling handoff calls in HCN. Guard channels [3,9,16] are reserved for serving the handoff calls in both microcells and macrocells. Blocked handoff calls [1] are queued before there is any free channel available to reduce the force termination probability. Lagrange and Goldewski [10] proposed that only handoff calls can overflow from the microcell to the macrocell. Chung and Lee [11] proposed a mobility-dependent call admission control scheme that new calls are accepted according to some mobility-based acceptance probability while handoff calls are admitted when there are free channel available.

These existing schemes can effectively reduce the handoff call forced termination probability, but at the cost of the increment of new call blocking probability. In [5], a new channel assignment scheme called the sub-rating is introduced to address handoff call problem in the Personal Cellular System (PCS). Sub-rating means that an occupied full-rate channel can be temporarily divided into two channels at the half of the original rate: one to serve the existing call and the other to serve the handoff request. By using the channel sub-rating, it is shown that the forced termination probability is greatly reduced with a slight increase of the blocking probability of new calls. The channel sub-rating scheme was used in [17] to improve the performance of handoff packets for the cellular radio network. The proposed scheme which combines fixed channel assignment, sub-rating, and directed retry, gives a significant improvement in the QoS of the cellular radio network. In [18], Yamanaka and Shimohara applied channel sub-rating for handoff calls in CDMA cellular networks. However, in all of the aforementioned studies, the channel sub-rating is only used in a single-tier system. Even some preliminary results about the utilization of sub-rating in HCN were reported in [19], this paper gives a complete investigation of the sub-rating strategy for handoff calls in HCN. A general two-tiers HCN system model is considered in this paper. Performance models are developed for the microcell layer and macrocell layers respectively. Two important QoS merits, new call blocking probability and handoff call forced termination probability, are computed to evaluate the proposed call admission control scheme. Meanwhile, the degraded voice quality due to the half-rate channel is also investigated to study the effects of the sub-rating on the QoS. Furthermore, the partial sub-rating by decreasing the number of sub-ratable full-rate channel is taken into consideration to balance the forced termination probabilities and the voice quality.

The rest of the paper is organized as follows. The proposed CAC strategy with channel sub-rating and unidirectional call overflow is introduced in Section 2. In Section 3, the system model and performance analysis of micro/macro layers are described with details. Numerical results are presented in Section 4. Finally, concluding remarks are given in Section 5.

2. Call Admission Control With Sub-Rating

The HCN system supports both fast-mobility users and low-mobility users, e.g. pedestrians. We use the speed-sensitive selection strategy, i.e., the macrocells handle the fast-mobility users while the microcells serve the low-mobility users. In the following, we describe the proposed CAC scheme with the channel sub-rating. Notice that some other techniques such as queuing [5] are orthogonal to the proposed scheme thus not investigated here.

The scheme is illustrated in Figure 1. In each microcell, slow new or handoff calls will be serviced with a full-rate channel if the total number of calls currently being serviced does not exceed the full capacity of the microcell. Otherwise, handoff requests are handled by using sub-rating that is an occupied full-rate channel but temporarily divided into two channels at a half of the original rate: one to serve the existing call and the other to serve the new incoming slow handoff request. To reduce the blocking probability of slow new calls, blocked slow new calls are allowed to overflow to the overlaying macrocell.

In a macrocell, fast new or handoff calls and overflowed new slow calls will be served with a full-rate channel if the number of calls exiting does not exceed the full capacity of the macrocell. Otherwise, new calls, including fast new calls and overflowed slow new calls will be blocked, while handoff calls including fast handoff calls and slow handoff calls incurred by the overflowed slow new calls are handled based on the sub-rating, similar to those at the microcell layer. The overflow of the fast calls from macrocell to microcell is not allowed because it will incur frequent handoff of the fast calls in the microcell layer due to small size of a microcell. Takeback is also not used as it needs the system to continuously monitor the available resource in the microcell layer [13].

3. System Model and Performance Analysis

3.1. System Model

For the analysis purpose, we introduce the system model of the two-tier HCN with the following assumptions.

- A macrocell in the macrocell layer is overlaid completely by N microcells in the microcell layer and there is no macrocell-only or microcell-only area.
- Cells in the same layer have the same shape and size.
- Each microcell contains C_m channels and each macrocell contains C_M channels.
- The fast-mobility calls are first directed to a macrocell, while slow-mobility calls are first directed to a microcell.
- The velocity of mobile terminals is assumed not to change greatly during its call lifetime.
- The total new call arrival to a cluster formed by a macrocell

and its associated N microcells follows a Poisson process with the mean arrival rate λ_n .

- The portion of slow new calls in total call arrivals to a cluster is a .
- The slow/fast handoff call arrival to a given cell at each layer is assumed to be an independent Poisson process with the mean arrival rates $\lambda_{sh}^m / \lambda_{fn}^M$, respectively.
- The overflowed slow new calls from the microcell layer arrive at the overlaying macrocell in accordance with a Poisson process with the mean rate λ_{sno}^m ; and the slow handoff calls resulting from the overflowed slow new calls arrive at the macrocell in accordance with a Poisson process with the mean rate λ_{sh}^M .
- The call holding time in a macrocell or microcell is exponentially distributed with the mean rate $1/\mu$.
- The dwell time is the mobile residence time in a given cell. For a slow mobile, its dwell times are exponentially distributed with the mean rate $1/\gamma_s^m$ and $1/\gamma_s^M$ in the microcell and macrocell respectively. The dwell time of a fast mobile in the macrocell is exponentially distributed with the mean rate $1/\gamma_f^M$.

Since the arrival process of the overflowed calls is generally not Poisson distributed, above assumptions are an approximation made for analytic tractability. Therefore, the slow and fast new call arrival rates can be obtained as

$$\lambda_{sn}^m = \frac{1}{N} a \lambda_n; \quad \lambda_{fn}^M = (1-a) \lambda_n \tag{1}$$

Also we can obtain the channel holding time of slow calls in a microcell (or macrocell) which is exponentially distributed with the mean rate of $1/\mu_s^m = 1/(\mu + \gamma_s^m)$ or $1/\mu_s^M = 1/(\mu + \gamma_s^M)$. In the same manner, we can find that the channel holding time for fast calls in a macrocell is also exponentially distributed with mean $1/\mu_f^M = 1/(\mu + \gamma_f^M)$.

Based on the system model, we can perform the performance analysis of the proposed call admission control strategy in the two-layer HCN. In the analysis, we assume that all full-rate channels are sub-ratable and the model can be easily modified for partial sub-rating. The analysis is carried out separately for the microcell and macrocell layers.

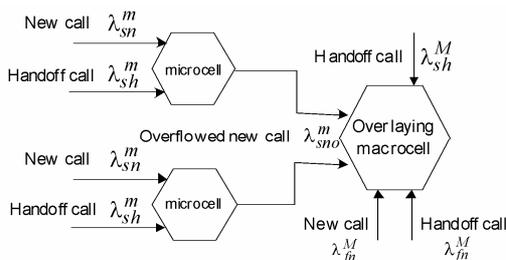


Figure 1. CAC scheme with channel sub-rating and unidirectional calls overflow.

3.2. Performance Analysis of the Microcell Layer

Based on the above system model, the microcell layer in the two-layer HCN system can be modeled as a one-dimension Markov process.

The state transition diagram of the Markov process is shown in Figure 2. In this diagram, a state is defined as the number of calls in progress in a microcell. For $0 \leq i \leq C_m$, $s(i)$ represents that there are i full-rate busy channels being occupied in a microcell. If there are less than C_m busy channels, the arrived slow call (either a new call or a handoff call) is allocated one of those free channels. The Markov process moves from current state $s(i)$ to the next state $s(i+1)$ with the rate $\lambda_{sn}^m + \lambda_{sh}^m$. The transition rate from state $s(i+1)$ to $s(i)$ is given by $(i+1)\mu_s^m$.

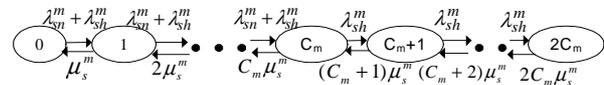


Figure 2. State transition diagram for the microcell.

For $C_m < i \leq 2C_m$, $s(i)$ represents that there are $2C_m - i$ full-rate busy channels, and $2(i - C_m)$ half-rate busy channels. When all full-rate channels in the microcell layer are busy, the arrived slow new call will be blocked while the slow handoff calls will be served by sub-rating a full-rate busy channel into two half-rate channels. If $i = 2C_m$, all channels are sub-rated, any arrived slow hand-off call will be blocked. The Markov process moves from $s(i)$ to $s(i+1)$ with the rate λ_{sh}^m . The transition rate from state $s(i+1)$ to $s(i)$ is given by $(i+1)\mu_s^m$. The steady state probability π_i^m of state $s(i)$ in a microcell is given as follows

$$\pi_i^m = \begin{cases} \frac{\lambda_{sn}^m + \lambda_{sh}^m}{i! \mu_s^m} \pi_{i-1}^m, & 0 \leq i \leq C_m \\ \frac{\lambda_{sh}^m}{i! \mu_s^m} \pi_{i-1}^m, & C_m < i \leq 2C_m \end{cases} \tag{2}$$

After the normalization condition that $\sum_{i=0}^{2C_m} \pi_i^m = 1$, it yields

$$\pi_0^m = \left[\sum_{i=0}^{C_m} \frac{(\lambda_{sn}^m + \lambda_{sh}^m)^i}{i! (\mu_s^m)^i} + \sum_{i=C_m+1}^{2C_m} \frac{(\lambda_{sn}^m + \lambda_{sh}^m)^{C_m} (\lambda_{sh}^m)^{i-C_m}}{i! (\mu_s^m)^i} \right]^{-1} \tag{3}$$

From Equations (2) and (3), the respective probabilities of the new and handoff slow calls being blocked in a microcell, P_{snb}^m and P_{shb}^m , are expressed as

$$P_{snb}^m = \sum_{i=C_m+1}^{2C_m} \pi_i^m; \quad P_{shb}^m = \pi_{2C_m} \tag{4}$$

According to the CAC scheme described in Section 2, the new slow calls that are blocked at the microcell layer

are not dropped but overflow to the overlaying macrocell (see Figure 2). Based on Equation (3), the aggregate mean arrival rate λ_{sno}^m of the overflowed slow new call from N microcells to the overlaying macrocell can be determined as

$$\lambda_{sno}^m = NP_{snb}^m \lambda_{sn}^m \quad (5)$$

3.3. Performance Analysis of the Macrocell Layer

Based on the above assumptions, the total call flow to an individual macrocell can be divided into the following four portions: 1) fast new calls with the arrival rate of λ_{fn}^M ; 2) fast handoff calls with the arrival rate of λ_{fh}^M ; 3) overflowed slow new calls with the arrival rate of λ_{sno}^m ; 4) handoff calls resulting from the overflowed slow new calls with the arrival rate of λ_{sh}^M .

The call state process in the macrocell layer can be modeled by a two-dimensional Markov process as shown in Figure 3. For each state (i, j) , i represents the number of fast calls including the fast new and handoff calls, and j is the number of slow calls including the overflowed slow new calls and the incurred slow handoff calls in the macrocell layer. The maximum channel capacity of a macrocell is $2C_M$, which means $0 \leq i + j \leq 2C_M$. In the state transition diagram, λ_{fn}^M and λ_{fh}^M cause the state

change along the horizontal direction while λ_{sno}^m and λ_{sh}^M alter the state change along the vertical direction.

Following the state transition diagrams in Figure 3 and 4, the equilibrium equations can be derived as follows

$$(\lambda_{fn}^M + \lambda_{fh}^M + \lambda_{sno}^m + \lambda_{sh}^M) \pi_{0,0} = \mu_f^M \pi_{1,0} + \mu_s^M \pi_{0,1} \quad (6)$$

$$(\delta_A \lambda_{fn}^M + \lambda_{fh}^M + \lambda_{sno}^m + \lambda_{sh}^M + i \mu_f^M) \pi_{i,0} = (i+1) \mu_f^M \pi_{i+1,0} + \mu_s^M \pi_{i,1} + (\delta_B \lambda_{fn}^M + \lambda_{fh}^M) \pi_{i-1,0}, \quad 0 < i < 2C_M \quad (7)$$

$$\lambda_{fh}^M \pi_{2C_M-1,0} = 2C_M \mu_f^M \pi_{2C_M,0} \quad (8)$$

$$(\lambda_{fn}^M + \lambda_{fh}^M + \delta_C \lambda_{sno}^m + \lambda_{sh}^M + j \mu_s^M) \pi_{0,j} = \mu_f^M \pi_{1,j} + (j+1) \mu_s^M \pi_{0,j+1} + (\delta_D \lambda_{sno}^m + \lambda_{sh}^M) \pi_{0,j-1}, \quad 0 < j < 2C_M \quad (9)$$

$$\lambda_{sh}^M \pi_{0,2C_M-1} = 2C_M \mu_s^M \pi_{0,2C_M} \quad (10)$$

$$(\lambda_{fn}^M + \lambda_{fh}^M + \lambda_{sno}^m + \lambda_{sh}^M + i \mu_f^M + j \mu_s^M) \pi_{i,j} = (i+1) \mu_f^M \pi_{i+1,j} + (j+1) \mu_s^M \pi_{i,j+1} + (\lambda_{sno}^m + \lambda_{sh}^M) \pi_{i,j-1} + (\lambda_{fn}^M + \lambda_{fh}^M) \pi_{i-1,j}, \quad 0 < i + j < C_M \quad (11)$$

$$(\lambda_{fn}^M + \lambda_{sh}^M + i \mu_f^M + j \mu_s^M) \pi_{i,j} = (i+1) \mu_f^M \pi_{i+1,j} + (j+1) \mu_s^M \pi_{i,j+1} + (\lambda_{sno}^m + \lambda_{sh}^M) \pi_{i,j-1} + (\lambda_{fn}^M + \lambda_{fh}^M) \mu_s^M \pi_{i-1,j}, \quad i + j = C_M \quad (12)$$

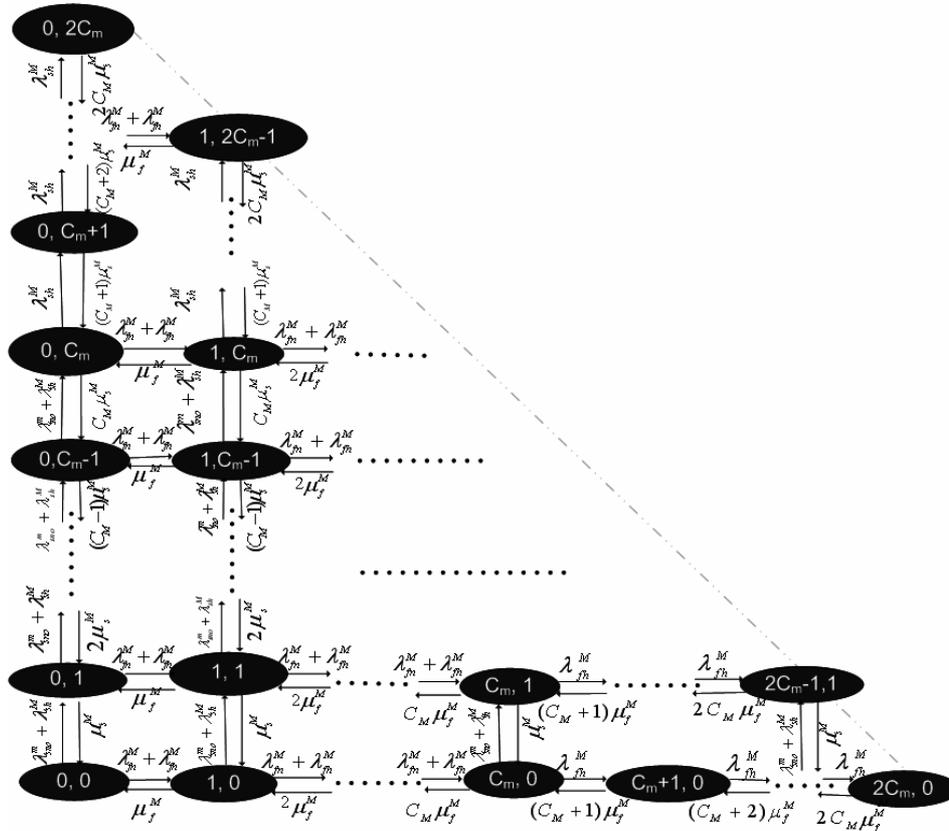


Figure 3. State transition diagram for a macrocell.

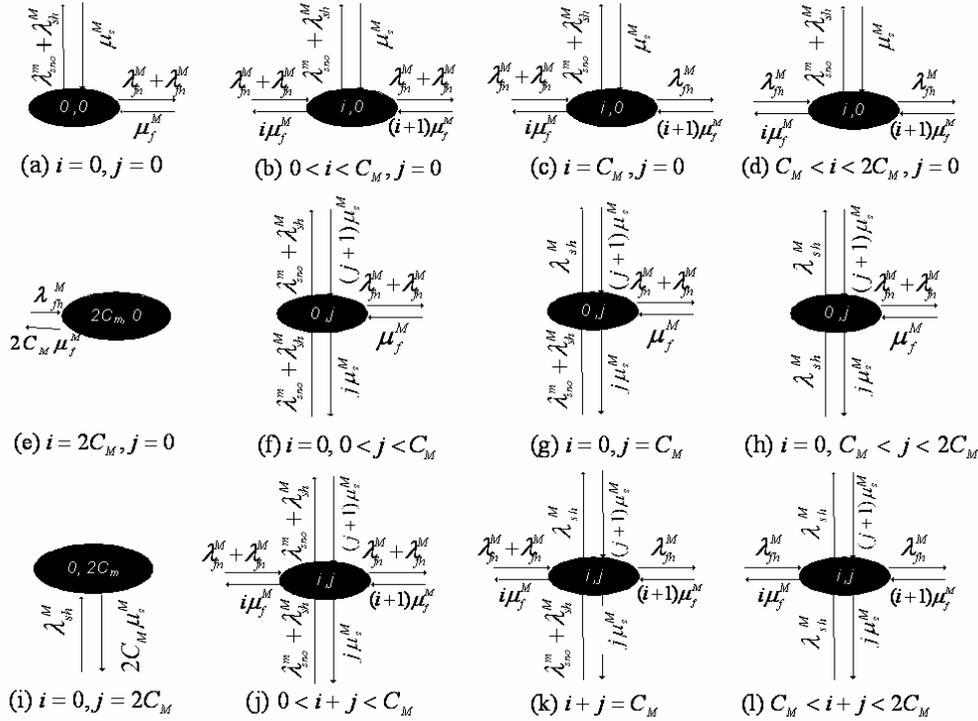


Figure 4. Specific states in the state transition diagram.

$$\begin{aligned}
 & (\lambda_{fh}^M + \lambda_{sh}^M + i\mu_f^M + j\mu_s^M) \pi_{i,j} = (i+1)\mu_f^M \pi_{i+1,j} \\
 & + (j+1)\mu_s^M \pi_{i,j+1} + \lambda_{sh}^M \pi_{i,j-1} + \lambda_{fh}^M \pi_{i-1,j}, \quad C_M < i+j < 2C_M
 \end{aligned} \tag{13}$$

where $\delta_A = 1$ for $0 < i < C_M$, $\delta_B = 1$ for $0 < i \leq C_M$, $\delta_C = 1$ for $0 < j < C_M$, and $\delta_D = 1$ for $0 \leq j \leq C_M$.

We use the Gauss-Seidel iterative method to solve the above linear equations and compute the steady-state probabilities $\pi_{i,j}$. The probabilities of fast new calls and overflowed slow new calls blocked in the macrocell layer, P_{fnb}^M and P_{sno}^M , are then expressed as

$$P_{fnb}^M = P_{sno}^M = \sum_{C_M \leq i+j \leq 2C_M} \pi_{i,j}^M \tag{14}$$

The blocking probabilities of the fast handoff calls and the overflowed slow handoff calls, P_{fsh}^M and P_{sho}^M , are expressed as

$$P_{fsh}^M = P_{sho}^M = \sum_{i+j=2C_M} \pi_{i,j}^M \tag{15}$$

From (14) and (15), the total new and handoff slow call blocking probabilities including microcell and macrocell, P_{snb} and P_{shb} can be obtained respectively as

$$P_{snb} = P_{snb}^m P_{sno}^M; \quad P_{shb} = P_{shb}^m + P_{snb}^m (1 - P_{sno}^M) P_{sho}^M \tag{16}$$

The arrival rates of slow and fast handoff calls can be obtained as in [5] using following three expressions.

$$\lambda_{sh}^m = \frac{\gamma_s^m (1 - P_{snb}^m)}{\mu + \gamma_s^m P_{shb}^m} \lambda_{sn}^m, \quad \lambda_{sh}^M = \frac{\gamma_s^M (1 - P_{sno}^M)}{\mu + \gamma_s^M P_{sho}^M} \lambda_{sno}^m, \quad \lambda_{fh}^M = \frac{\gamma_f^M (1 - P_{fnb}^M)}{\mu + \gamma_f^M P_{fnb}^M} \lambda_{fn}^M \tag{17}$$

Since the handoff rates and the steady state probabilities are mutually related, an iterative algorithm is employed here to compute them.

3.4. The Degradation Ratio

The sub-rating scheme is efficient under the limited channel resources because more handoff calls can be served. However, we need to consider the degradation of the voice quality due to the employment of sub-rating. In the following, the degradation ratios of voice quality for the microcell and macrocell layers are evaluated.

The expected number $E[H_{busy}^m]$ of the busy channels in the microcell layer is expressed as

$$E[H_{busy}^m] = \sum_{i=0}^{2C_m} i \pi_i^m \tag{18}$$

The expected number $E[H_{sub}^m]$ of the sub-rated channels in the microcell layer is obtained as follows

$$E[H_{sub}^m] = \sum_{i=C_m+1}^{2C_m} 2(i - C_m) \pi_i^m \tag{19}$$

The degradation ratio of the voice quality in the microcell layer, DR_m , is the portion of the call holding time that the mobile user experiences the degraded voice

quality [20]. The mean degradation ratio of the voice quality, $E[DR_m]$, can be expressed as

$$E[DR_m] = E\left[\frac{H_{sub}^m}{H_{busy}^m}\right] = \sum_{i=C_m+1}^{2C_m} \frac{2(i-C_m)}{i} \pi_i^m \quad (20)$$

Similarly the mean degradation ratio of the voice quality in the macrocell layer, $E[DR_M]$, can be obtained as

$$E[DR_M] = \sum_{(i+j)=C_M+1}^{2C_M} \frac{2(i+j-C_M)}{(i+j)} \pi_{i,j}^M \quad (21)$$

4. Numerical Results

In this section, numerical results derived from the model developed in Section 3 are presented to quantify the performance of the proposed scheme. The results are compared to the scheme with guard channel and unidirectional call overflow proposed in [9], referred as Shan03 in the plots. If otherwise not specified, the parameters and their values used for the experiments are those shown in Table 1 [11,21], where C_g and C_G are the numbers of guard channels reserved in the microcell and macrocell for Shan03 scheme.

Table 1. Parameters and their values.

Parameter	Value	Parameter	Value
$1/\mu$	110s	N	4
$1/\gamma_s^m$	150s	C_g	1
$1/\gamma_s^M$	300s	C_G	1
$1/\gamma_f^M$	60s	C_m	5
a	0.8	C_M	10

4.1. Case 1: Full Sub-Rating

In this part, we present the experimental results generated from the model developed in Section 3 where all the full-rate channels are considered to be sub-ratable.

Figure 5 plots the slow/fast new call blocking probabilities and slow/fast handoff call forced termination probabilities under various total new call arrival rates for the proposed scheme and Shan03 Scheme. For Shan03 scheme, we investigate the performance with different guard channel numbers. It can be seen from Figures 5(b) & (d) that the slow/fast handoff call forced termination probabilities of the proposed scheme are almost zero and also much lower than those of Shan03, even in the case that two guard channels are reserved for handoff calls. Figures 5(a) & (c) reveal that when the number of guard channel is set to one, the new fast call blocking probability of Shan03 are almost the same as that of the proposed scheme while the new slow call blocking probability of Shan03 is slightly higher than that of the proposed scheme when the number of guard channel is set to two. The reason is that the proposed scheme allows the micro-macro overflow of the new slow calls when

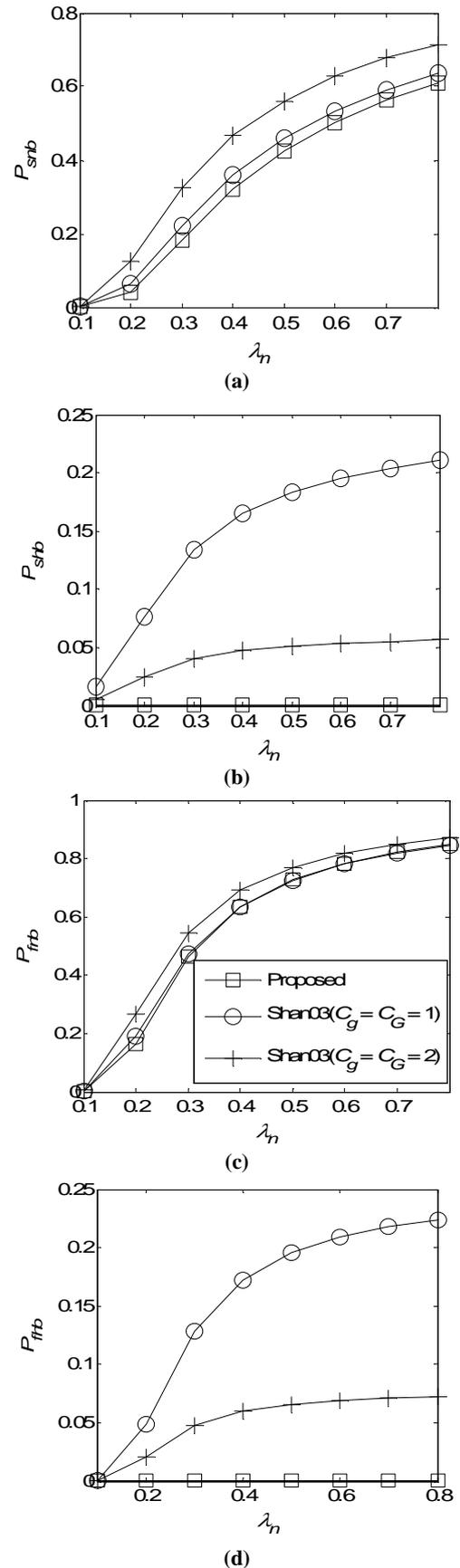


Figure 5. Performance comparison of the proposed scheme with the Shan03 scheme, P_{snb} , (b) P_{sfb} , (c) P_{fnb} , (d) P_{fhb} .

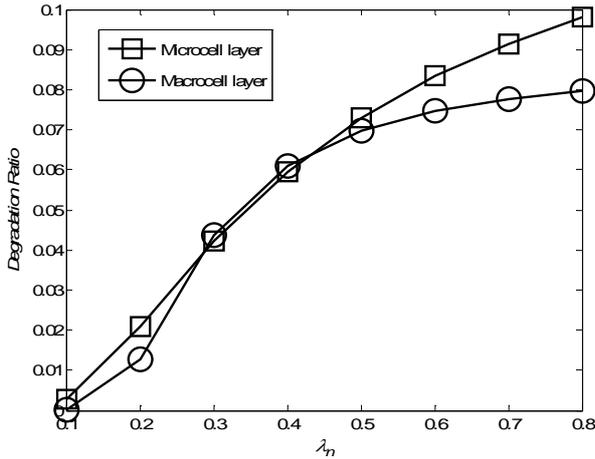


Figure 6. The degradation ratios of the microcell and macrocell layers.

there is no free channel in the microcell. For Shan03 scheme, although reserving more guard channels can greatly reduce the handoff call forced termination probability, the new call blocking probability also increases significantly. The results demonstrate that the proposed scheme achieves better performance for both new and handoff calls than Shan03 scheme in all cases. By employing channel sub-rating strategy, the proposed CAC scheme can massively reduce the probability of forced termination of handoff calls while maintaining the new call blocking probability as low as possible.

Under the channel sub-rating strategy, when calls are in progress, the mobile user is expected to experience the degraded voice quality due to the half-rate channel.

Figure 6 plots the corresponding degradation ratios of the voice quality in the microcell and macrocell layers of Figure 5. Results indicate that the slow and fast mobility subscribers experience the degraded voice quality for a very short period of the total channel holding time when the total new call arrival rate is 0.1 calls/second. As the total new call arrival rate increases, the degradation ratios increase correspondingly. When the total new call arrival rate reaches 0.8 calls/second, the degradation ratios for the microcell and macrocell are about 10% and 8%, respectively. To reduce the degradation ratio, we can decrease the number of full-rate channel that can be sub-rated which in turn increases the handoff call forced termination probability. Next we will investigate partial sub-rating to balance the forced termination probabilities and the voice quality.

4.2. Case 2: Partial Sub-Rating

In this part, we present the experimental results when only partial full-rate channels can be sub-rated. The number of sub-ratable full-rate channels is denoted as C_{sm}/C_{sM} for the microcell/macrocell.

Figure 7 plots the slow and fast handoff call forced termination probabilities and degradation ratios for the microcell and macrocell layers under various C_{sm} and

C_{sM} . The total new call arrival rate is set to 0.5 calls/second. It can be seen that the handoff call forced termination probability drops rapidly as the number of sub-ratable full-rate channels increases. On the other hand, the degradation ratios for the microcell and macrocell layers become higher with the increment of the number of sub-ratable full-rate channels. Therefore to meet the QoS requirements of the system on the handoff call forced termination probability and degradation ratio of the voice quality, the number of sub-ratable full-rate channels must be carefully chosen to achieve the best tradeoff.

In Figure 8, we compare the performance of proposed scheme under partial sub-rating with that of Shan03 scheme for various total new call arrival rate, where C_{sm} , C_{sM} , C_g and C_G are both set to 1. The results demonstrate that the proposed scheme outperforms the guard channel-based Shan03 scheme. Even in this case, only one full-rate channel is allowed to be sub-ratable.

5. Conclusions

A call admission control scheme for HCN has been proposed in this paper which incorporates the channel sub-rating and one-way call overflow. By sub-rating the occupied channels to create new half-rate channels for the handoff calls, the forced terminations of handoff calls are expected to be greatly reduced. Meanwhile, the micro-to-macro overflow was introduced to reduce the blocking probability of slow new calls. We developed analytic models to evaluate the performance of the proposed scheme. Experimental results have shown that the proposed scheme based on channel sub-rating achieves better performance for both new and handoff calls compared to the scheme based on guard channels.

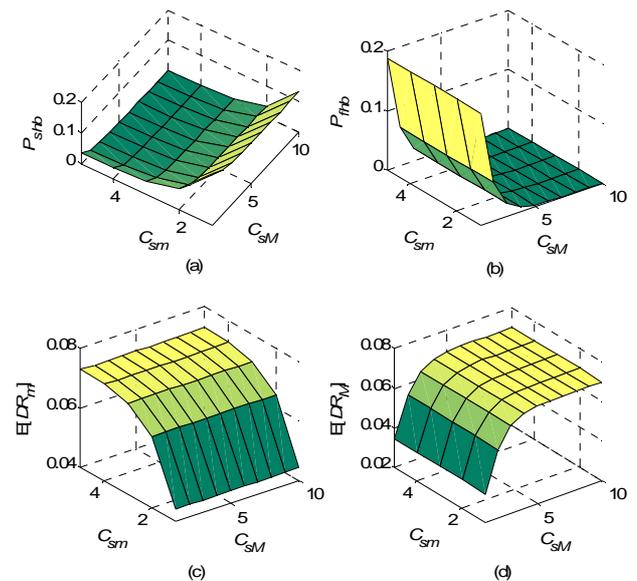


Figure 7. Performance metrics of the proposed scheme under various C_{sm} and C_{sM} , (a) P_{sfb} , (b) P_{fhb} , (c) $E[DR_m]$, (d) $E[DR_M]$.

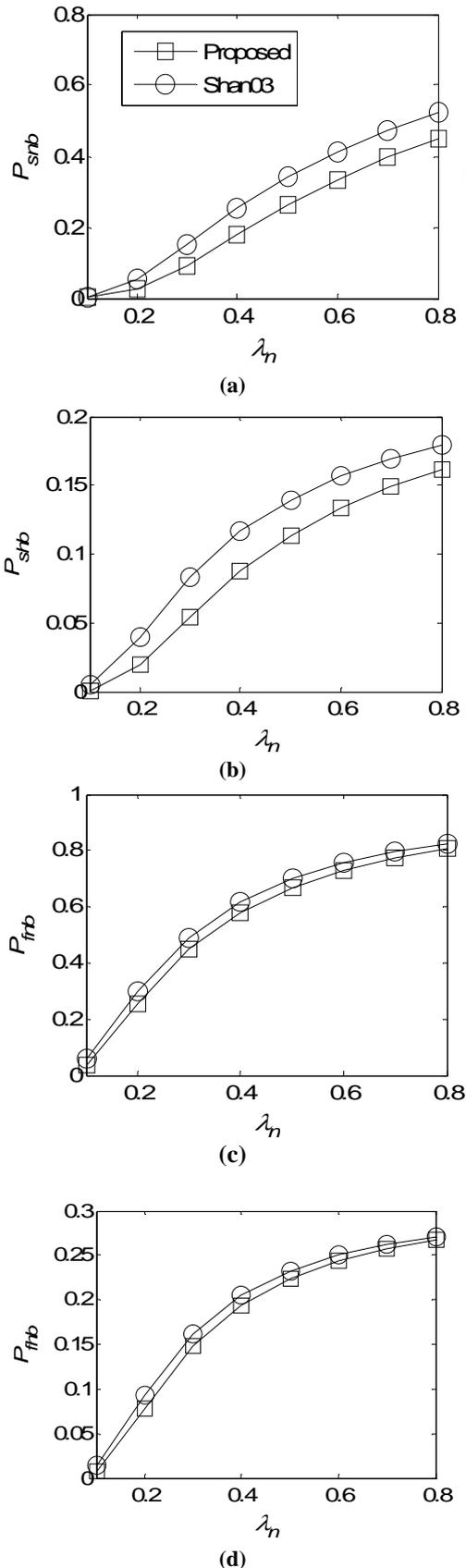


Figure 8. Performance comparison of the proposed scheme under partial sub-rating with Shan03 scheme for various λ_n , $C_{sm}=C_{sM}=C_g=C_G=1$, (a) P_{snb} , (b) P_{shb} , (c) P_{fnb} , (d) P_{fhb} .

The penalty of the sub-rating is the degradation of voice quality during the time when the channels are sub-rated to accommodate the handoff calls. Our results indicated that there is a tradeoff between handoff call forced termination probability and degradation ratio of the voice quality when the channel sub-rating is employed. By assigning the appropriate number of the sub-ratable full-rate channels, we can achieve good balance between the handoff call forced termination and the voice quality degradation.

6. References

- [1] D. Hong and S. S. Rappaport, "Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and no-protection handoff procedure," *IEEE Transactions on Vehicular Technology*, Vol. 35, pp. 77–92, 1986.
- [2] Y. S. Hwang, Y. N. Han, and Y. H. Kim, "Performance analysis of mixed voice/data service in a macrocell-based PCS network," *IEICE Transactions on Fundamentals*, Vol. E81-A, pp. 1136–1144, 1998.
- [3] X. Liu and A. O. Fapojuwo, "Performance analysis of hierarchical cellular networks with queuing and user retrials," *International Journal of Communication Systems*, Vol. 19, pp. 699–721, 2006.
- [4] S. S. Tang and W. Li, "Modeling and analysis of hierarchical cellular networks with bidirectional overflow and take-back strategies under generally distributed cell residence times," *Telecommunication Systems*, Vol. 32, pp. 71–91, 2006.
- [5] Y. B. Lin and A. R. Noerpel, "The sub-rating channel assignment strategy for PCS hand-offs," *IEEE Transactions on Vehicular Technology*, Vol. 45, pp. 122–130, 1996.
- [6] K. Maheshwari and A. Kumar, "Performance analysis of microcellization for supporting two mobility classes in cellular wireless networks," *IEEE Transactions on Vehicular Technology*, Vol. 49, pp. 321–333, 2000.
- [7] C. Hartmann and O. Schelegelmilch, "Hierarchical cell structures with adaptive radio resource management," in *Proceedings IEEE Vehicular Technology Conference*, pp. 1764–1771, 2000.
- [8] X. Yang and M. Guizani, "Optimal paging load balance with total delay constraint in macrocell-microcell hierarchical cellular networks," Vol. 5, No. 8, pp. 2202–2209, 2006.
- [9] W. H. Shan and P. Z. Fan, "Performance evaluation of a hierarchical cellular system with mobile velocity-based bidirectional call-overflow scheme," *IEEE Transactions on Parallel and Distributed System*, Vol. 14, pp. 72–83, 2003.
- [10] X. Lagrange and P. Godlewski, "Teletraffic analysis of a hierarchical cellular network," in *Proceedings IEEE Vehicular Technology Conference*, pp. 882–886, 1995.
- [11] S. Chung and J. Lee, "Mobility-dependent call admission control in hierarchical cellular networks," *Computer Communications*, Vol. 25, pp. 700–713, 2000.

- [12] S. P. Shung and J. C. Lee, "Mobility-dependent call admission control in hierarchical cellular networks," *Computer Communications*, Vol. 25, pp. 700–713, 2000.
- [13] G. Boggia, P. Camarda, and N. Di Fonzo, "Teletraffic analysis of hierarchical cellular communication networks," *IEEE Transactions on Vehicular Technology*, Vol. 52, pp. 931–946, 2003.
- [14] B. Jabbari and W. F. Fuhrmann, "Teletraffic modeling and analysis of flexible hierarchical cellular networks with speed-sensitive handoff strategy," *IEEE Journal on Selected Areas in Communications*, Vol. 15, pp. 1539–1548, 1997.
- [15] Y. Zhang and B. H. Song, "Handoff counting in hierarchical cellular system with overflow scheme," *Computer Networks*, Vol. 46, pp. 541–554, 2004.
- [16] X. Liu and A. O. Fapojuwo, "Resource management for handoff traffic in hierarchical cellular networks," in *Proceedings 24th International, Computing, and Communications Conference*, pp. 615–620, 2005.
- [17] M. J. Rakhee, "A subrating channel assignment scheme for cellular radio network with directed retry," *Computers & Operations Research*, Vol. 32, pp. 2407–2417, 2005.
- [18] S. Yamanaka and K. Shimohara, "Performance analysis of CDMA cellular networks with channel sub-rating," *International Journal of Communication Systems*, Vol. 19, pp. 317–333, 2005.
- [19] X. Wu, J. Zheng, E. Regentova, and Y. Jiang, "Analysis of the effect of channel sub-rating in unidirectional call overflow scheme for call admission in hierarchical cellular networks," in *Proceedings Vehicular Technology Conference*, pp. 1265–1269, 2007.
- [20] S. H. Wie, J. S. Jang, B. C. Shin, and D. H. Cho, "Handoff analysis of the hierarchical cellular system," *IEEE Transactions on Vehicular Technology*, Vol. 49, pp. 2027–2036, 2000.
- [21] S. H. Lee and J. S. Lim, "Performance analysis of channel allocation schemes for supporting multimedia traffic in hierarchical cellular systems," *IEICE Transactions on Communications*, Vol. E86-B, pp. 1274–1285, 2003.

A Priority Queuing Model for HCF Controlled Channel Access (HCCA) in Wireless LANs

Reza GHAZIZADEH¹, Pingzhi FAN¹, Yi PAN²

¹Keylab of Information Coding and Transmission, Institute of Mobile Communications,
Southwest Jiaotong University, Chengdu, China

²Department of Computer Science, Georgia State University, Atlanta, GA, USA

Emails: r_ghazizadeh@yahoo.com, p.fan@ieee.org, pan@cs.gsu.edu

Received June 15, 2008; revised December 22, 2008; accepted December 31, 2008

Abstract

Recently, there has been a rapid growing interest in new applications requiring quality of service (QoS) guarantees through wireless local area networks (WLAN). These demands have led to the introduction of new 802.11 standard series to enhance access medium supporting QoS for multimedia applications. However, some applications such as variable bit rate (VBR) traffic address some challenges in the hybrid coordination function (HCF) nominated to provide QoS. This paper presents a novel priority queuing model to analyze a medium access in the HCF controlled channel access (HCCA) mode. This model makes use of a MAP (Markovian Arrival Process)/PH (Phase Type)/1 queue with two types of jobs which are suitable to support VBR traffic. Using a MAP for traffic arrival process and PH distribution for service process, the inclusion of vacation period makes our analysis very general and comprehensive to support various types of practical traffic streams. The proposed priority queuing model is very useful to evaluate and enhance the performance of the scheduler and the admission controller in the HCCA mechanism.

Keywords: QoS, HCCA, Priority Queues, Matrix-Geometric Method, MAP/PH/1

1. Introduction

Increasing demands to access to network in hotspots areas at airports, hotels, coffee shops have led wireless local area network (WLAN) to be a key technology for high-speed local access in public and private areas. Furthermore, in the near future, WLAN will play a key role within the hybrid wireless systems and also it is the best candidate to connect home devices to wireless networks. Therefore, it should be able to allow users to ubiquitously access a large variety of services. On the other hand, demands for new applications such as real time traffic, multimedia video and voice over IP are increasing rapidly. These applications have created need for QoS support.

However, IEEE802.11 is unsuitable for multimedia applications to support QoS in the MAC layer. Therefore, IEEE802.11 working group has been developing a new protocol, IEEE802.11e, which will be able to provide QoS features. IEEE802.11e introduces the hybrid coordination function comprised of two medium access mechanisms: contention-based channel access referred to

as *enhanced distributed channel access* (EDCA) and controlled channel access referred to as *HCF controlled channel access* (HCCA) [1].

Although contention-based channel access is very simple and robust for best effort traffic, it can not provide QoS guarantees easily. These can be achieved with the polling-based medium access through the HCCA. The HCCA provides a *hybrid coordinator* (HC) with ability to assign a contention free time interval during contention period and contention free period to packet transmission. Therefore, *transmission opportunity* (TXOP) and service interval (SI) are very important parameters to provide QoS guarantees. A reference scheduler calculates these parameters with the reservation information achieved through the negotiation with the end users. Using average values, such as average packet length and average data rate, to compute transmission parameters cause some challenges to QoS support in VBR traffic. Therefore, modification of the scheduler to provide such traffic is very crucial. In this way, analytical system analysis is very useful to improve and develop the system.

This paper presents a priority queuing model to analyze a medium access in the HCCA. According to the HCCA characteristics, the model is based on a MAP/PH/1 queue with vacation and time-limited service in the presence of two priority levels. Some important performance measures are presented after solving the queuing model through the matrix-analytic method [2,3]. To our best knowledge, although there are some papers which have investigated the HCCA and the EDCA through simulation [4–9], and the EDCA analytically [10–12], there is only one published analytical work for the HCCA [13], where a queuing model without priority levels has been developed. However, priority is a key point to separate various traffic streams with different QoS requirements, especially for real-time and non-real-time traffic. In addition, to manage resources, use bandwidth efficiently and provide QoS for various traffic streams, it is necessary that different traffic streams appear in the different statistic or dynamic priority levels in the system. Therefore, IEEE802.11e introduces four priority levels for eight groups of traffic streams. Consequently, to modify and investigate the performance of the scheduler and the admission controller, priority analytical model in the presence of prioritized traffic is very helpful. In [14], a priority model in a medium without a vacation period and time-limited service was introduced. That model can not be applied to the HCCA medium access which is based on a vacation period and time-limited service. This paper will present a priority queuing model to analyze medium access in the HCCA by making use of an MAP/PH/1 queue with two types of jobs which are suitable to support vast practical traffic streams.

The rest of the paper is organized as follows: in Section 2, we briefly describe the HCCA mechanism, phase type (PH) distribution and discrete Markovian arrival process (DMAP); our proposed model is presented in Section 3; the related performance measures are analyzed in Section 4; numerical and simulation results are given in Section 5; and finally conclusions are drawn in Section 6.

2. HCCA and System Parameters

2.1. HCCA

IEEE802.11e/HCCA is a polling-based medium and centralized scheduling which is controlled by the HC. Each station that requires a strict QoS support is allowed to send QoS requirement packets to the HC and the HC assign a corresponding transmission opportunity to the station. The HC can start a polling period at any time during a contention period after the medium remains idle for at least *point coordination function* (PCF) inter-frame space interval. Each station can transmit a sequences of data packets separated by a short inter-frame space during own TXOP allocated by the HC in a contention free period. Therefore, as it has shown in Figure 1, a sequence of transmission opportunities will be assigned

to the stations during each SI. Consequently, each station is polled once per SI and allowed to transmit its packets until its TXOP duration elapses. Uplink and downlink TXOPs are initiated by the scheduler in the HC and end when there is no packet in the queues for transmission or TXOP duration expires. To provide QoS, each station manages QoS control field added to the legacy frames. Consequently, the scheduler receives separated reservation information of different traffic streams to calculate an aggregated service schedule. Some of this information is the mean data rate, delay bound, maximum burst size, minimum physical rate, user priority and peak data rate. The scheduler, first of all calculates the maximum service interval according to the delay bound for each traffic stream. Then, it selects the smallest service interval among all the maximum service intervals corresponding with the traffic streams as a service interval for all stations. The scheduler after that determines TXOP for each traffic stream according to the negotiated reservation information. Allocated TXOP for each station is sum of all TXOPs of station's traffic streams. TXOP of *j*th station that has made *n* traffic streams is computed as follows.

$$TXOP_j = \sum_{i=1}^n \max\left(\frac{\lceil \frac{SI \times \rho_i}{L_i} \rceil \times L_i}{R_i} + o, \frac{M_i}{R_i} + o\right) \quad (1)$$

where *R* denotes the minimum physical data rate, *L* and *M* represents the nominal and the maximum size of packet respectively, ρ denotes the mean data rate and *O* represents the overhead due to the physical and MAC headers, acknowledgment and polling frames. According to the service interval duration, the number of active stations and the TXOP duration in each station, an admission controller manages the number of active stations to provide QoS.

2.2. Discrete Markovian Arrival Process (DMAP)

Consider a discrete time Markov chain with a transition matrix *D* and two sub-stochastic matrix D_0, D_1 , where $D = D_0 + D_1$. Suppose at the time *t*, the Markov chain is in the state *i*, $1 \leq i \leq n$. Then, at the time epoch *t*+1 with probability $(D_k)_{ij}$, $k=0,1$ arrival process enters state *j*, $1 \leq j \leq n$ and starts a batch of *k* arrivals. Therefore, D_0 corresponds to a transition matrix with no arrival and D_1 corresponds to a transition matrix with only one arrival per time slot.

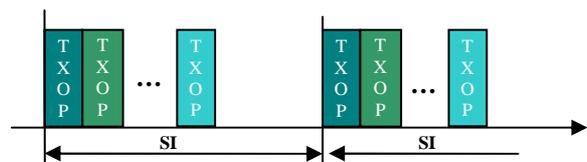


Figure 1. TXOP allocation in contention free period.

2.3. Phase Type (PH) Distribution

Phase type distributions can approximate many of general distributions encountered in queuing systems. Therefore, it is appropriate in service time modeling [2]. Consider an $m+1$ states Markov chain with one absorbing state, and initial probability vector $[\beta, \beta_{m+1}]$ with $m+1$ elements where β is a row vector with m element while $\beta e + \beta_{m+1} = 1$ (e is column vector of one). Let the transition probability matrix of the Markov chain be $P = \begin{bmatrix} S & S^0 \\ 0 & 1 \end{bmatrix}$.

where S is a sub-stochastic transition matrix, $I-S$ nonsingular (I is identity matrix) and $S^0 = e - S e$. The absorbing Markov chain can initialize from any states according to the initial vector and gets absorbed to the absorbing state. Therefore, the time to absorption in such a Markov chain is said to have phase type distribution which is represented with the pair (β, S) .

3. Proposed Priority Queuing Model

To clarify the model, suppose that there is one station communicating with the HC in the system. The HC can be considered as a single server which serves queues (high and low priority) of the station no more than T slots (maximum TXOP duration is divided to T slots) during each SI. In the view of the station, as soon as T time slots is used up or queues become empty, the server goes on a vacation (i.e. the server serves other stations or becomes idle until the next visit). Hence, as it is illustrated in Figure 2, the minimum vacation duration is subtraction of the SI and the maximum TXOP duration. It is assumed that the HC is a server which serves priority queues in non-preemptive priority case during a TXOP period. In a non-preemptive case, no service interruption is applied upon arrival of a high priority packet when a low priority packet is being served. To analyze the discrete time Markov chain (DTMC) describing the queuing model, arrival process, service process and vacation model are defined. The arrival process is modeled by a discrete Markovian arrival process (DMAP) to allow correlation among the inter-arrival times within packets (within each priority and between two priorities packets) and support various types of traffic streams, especially VBR traffic which generate packets in random arrival intervals. On the other

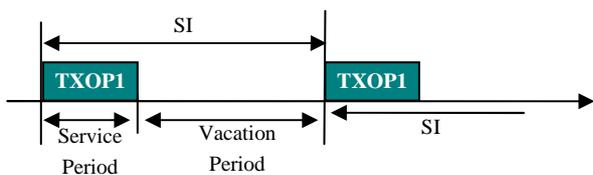


Figure 2. Service and vacation periods for one station in HCCA.

hand, it is obvious that the packet transmission time is corresponding to the service time which depends on the packet length while the channel data rate is fixed. Therefore, to support various packet length distributions and make the model of service process more general and comprehensive, a phase type (PH) distribution is proposed for a service process model. Consequently, the introduced priority queuing model is based on a MAP/PH/1 queue with vacation and time-limited service. The proposed model is based on the work of [14] which makes use of matrix-geometric solution for analysis priority queues without vacation and time limitation in service.

Some of the notations and symbols which will be used throughout the paper are introduced as follows: e is a column vector of one (with appropriate order equals to the number of columns of the matrix or to the vector length that it is multiplied with), $e_T(v)$ represents a column vector of zeroes with T length except at the v th position that is one, e' is the transpose of e vector, I_k denotes an identity matrix of dimension $k \times k$ and H_k represents $H_k = \begin{bmatrix} 0 & I_{k-1} \\ 0 & 0 \end{bmatrix}$.

3.1. Arrival Process

The arrival process is modeled by a discrete Markovian arrival process (DMAP). The DMAP, an extension of the Markov modulated Bernoulli process, can support many types of traffic flows such as VBR traffic generating variable packets length in variable inter-arrival periods. Suppose there are two independent types of traffic corresponding to two priorities where each traffic source is able to generate only one packet per time slot. Hence, each traffic flow will have two sub-stochastic matrices (Section 2.2) and consequently there are four sub-stochastic matrixes ($D_{00}, D_{11}, D_{12}, D_2$) corresponding to join packet arrival (high priority and low priority packet arrival) where D_{00} denotes a transition matrix with no packet arrival, D_{11} is a transition matrix when one high priority packet arrives, D_{12} represents a transition matrix when one low priority packet arrives, D_2 is a transition matrix when two high and low priority packet arrive (one of each priority packet) and also $D = D_{00} + D_{11} + D_{12} + D_2$ where D represents stochastic matrix. The arrival rate is $\lambda_i = \psi(D_{1i} + D_2)$ $i=1,2$ where $\psi = \psi e$ and $\psi e = 1$ (e is a column vector of one).

As mentioned above, four sub-stochastic matrices can be expressed by the sub-stochastic matrices of both traffic streams. Suppose $D_0(1)$ and $D_1(1)$ are the transition matrices in the high priority traffic when no packet and one packet arrives at a time slot, respectively. Furthermore assume, $D_0(2)$ and $D_1(2)$ are the transition matrices in the low priority traffic when no packet and one packet arrives at the time slot respectively.

In this case, $D_{00} = D_0(1) \otimes D_0(2)$, $D_{11} = D_1(1) \otimes D_0(2)$, $D_{12} = D_0(1) \otimes D_1(2)$ and $D_2 = D_1(1) \otimes D_1(2)$, where \otimes is the Kronecker product.

3.2. Service Process

The service process is corresponding to the transmission time. The total transmission time of a frame is sum of transmission time of data packet, its necessary headers added by the MAC and physical layer, ACK, and short inter-frame space (SIFS). We assume that the channel data rate, ACK, SIFS durations and header size are fixed. Hence, the service time of a packet can be considered as a random variable which varies only with the length of the packet. Consequently, to generalize the model and support different packet length distributions, we consider phase type distribution for both high priority and low priority service processes. Let (β_1, S_1) and (β_2, S_2) denote PH distribution for high priority and low priority services, respectively where S_1, S_2 are transition matrices of dimensions $m_1 \times m_1, m_2 \times m_2$, respectively and β_1, β_2 represent initial vectors. $S_1^0 = e - S_1 e$, $S_2^0 = e - S_2 e$ are transition to absorption vectors for the high priority and low priority services, respectively.

3.3. Vacation Model

In the service period, whenever there is no packet in the queues or the TXOP duration expires, the server enters a vacation period. Therefore, the vacation duration depends on the service duration. A vacation with the maximum duration begins whenever the server visits the station at the first slot of TXOP and the station has no packet to transmit. Consequently, a vacation model can be represented by a (δ_i, V) phase type distribution while the Markov chain can initialize from any states depending on the vacation duration. Therefore, the initial vector, the transition matrix and the transition to absorption vector will be $\delta_i = e_{SI}(i)$ $1 \leq i \leq T+1$, $V = H_{SI}$ and $V^0 = e - V e$, respectively.

3.4. State Space and Transition Matrix of the DTMC

In this subsection, we introduce state space and the transition matrix of the discrete time Markov chain (DTMC). The state space can be divided into two main groups that are vacation and service state spaces. Each of these states are described by the number of the packets in the high priority and low priority queues (i_1, i_2) , the phase of the arrival process (j) , the phase of the high priority or low priority service processes (k_1, k_2) , the phase of the vacation (l) and the phase of the TXOP (t) . Therefore, the states can be expressed as follows.

$$\begin{aligned}
 d^v &= \{(i_1, i_2, j, l), 0 \leq i_1 \leq Q; 0 \leq i_2 \leq Q; j = 1, 2, \dots, n; l = 1, \dots, r\} \\
 d_0^s &= \{(0, i_2, j, k_2, t), 1 \leq i_2 \leq Q; j = 1, 2, \dots, n; t = 1, \dots, T; k_2 = 1, 2, \dots, m_2\} \\
 d_1^s &= \{(i_1, i_2, j, k_2, t), 1 \leq i_1 \leq Q; 1 \leq i_2 \leq Q; j = 1, 2, \dots, n; t = 1, \dots, T; k_2 = 1, 2, \dots, m_2\} \\
 d_2^s &= \{(i_1, i_2, j, k_1, t), 1 \leq i_1 \leq Q; 0 \leq i_2 \leq Q; j = 1, 2, \dots, n; t = 1, \dots, T; k_1 = 1, 2, \dots, m_1\}
 \end{aligned} \tag{2}$$

where Q is the buffer size in the number of packets (high priority and low priority), d^v denotes the vacation states while the number of high priority and low priority packets in the queues are i_1 and i_2 respectively and the packet arrival is in j th phase as well as vacation is in l th phase, d_0^s represents service state space when there are only low priority packets in the system. Therefore, a low priority packet is being served while the service is in the phase K_2 at t th time slot and the packet arrival is in j th phase as well. In the same way, d_1^s is service state space when there is at least one high priority packet in the system and a low priority packet is being served, and d_2^s represents service state space when there is at least one high priority packet in the system and a high priority packet is being served.

The transition matrix of the discrete time Markov chain can be expressed as follows.

<i>probability of remaining in vacation</i>	<i>probability of switching from vacation to low priority service</i>	<i>probability of switching from vacation to high priority service</i>
<i>probability of switching from low priority service to vacation</i>	<i>probability of remaining in low priority service</i>	<i>probability of switching from low priority service to high priority service</i>
<i>probability of switching from high priority service to vacation</i>	<i>probability of switching from high priority service to low priority service</i>	<i>probability of remaining in high priority service</i>

Figure 3. General form of the transition probability sub matrix.

B_{00}^1 contains the state transitions increasing the number of packets in the low priority queue by one and remaining the high priority queue empty while there is at least one packet in the low priority queue before transition. One can easily compute the possible transitions such above discussions

$$B_{00}^1 = \begin{bmatrix} D_{12} \otimes V & e'_1(T) \otimes D_{12} \otimes V^0 \beta_2 \\ e_T(T) \otimes D_{12} \otimes e_{m_2} \delta & H_T \otimes D_{12} \otimes S_2 \end{bmatrix} \quad (9)$$

B_{00}^{-1} represents state transitions in which the number of packets in the low priority queue decreases by one and at least one packet is in the low priority queue before transition as well as high priority queue remains empty. These transitions happen when no packet arrives and the processing of a low priority packet is completed, and a) the processing of another one begins, b) the TXOP duration expires and a vacation period begins.

$$B_{00}^{-1} = \begin{bmatrix} 0 & 0 \\ e_T(T) \otimes [D_{00} \otimes S_2^0 \delta] & H_T \otimes D_{00} \otimes S_2^0 \beta_2 \end{bmatrix} \quad (10)$$

$B_{01}, B_{10}, A_{-1}, A_0, A_1$ and their elements can be computed in the same manner. The block matrices are given in Appendix A.

4. Performance Measures

According to the structure of P matrix, its steady state distribution vector can be obtained by applying the matrix-geometric method. Let probability steady state distribution vector be

$x = [x_0 \ x_1 \ x_2 \ \dots \ x_Q]$ where $x_i = [x_{i0} \ x_{i1} \ x_{i2} \ \dots \ x_{iQ}]$,
 $x_{ij} = [x_{ij}^v \ x_{ij}^1 \ x_{ij}^2]$ and $x_{ij}(k) = [x_{ij}^1(k) \ x_{ij}^2(k) \ \dots \ x_{ij}^T(k)]$
 $0 \leq i \leq Q, 0 \leq j \leq Q, k = 1, 2$ where $x_{ij}^n(k)$ is the probability that the number of packets in the high priority and in the low priority queues are i and j respectively while type k packet ($k = 1$: high priority, $k = 2$: low priority) is being served at the n th time slot of the TXOP period.

Using balanced equations ($x = xP, xe = 1$) and the matrix-geometric method, the steady state vector x can be calculated. For more details of how to find out steady state vector, readers can refer to [2].

4.1. Queue Length Distribution

Let $f_h(l)$ ($f_l(l)$) be the probability that there are l high priority packets (low priority packets) in the queue. The length of the high priority queue will be l if there are l high priority packets in the system and the server is not on the processing of the high priority packet (i.e. is on vacation or in the processing of the low priority packet) or, $l+1$ high priority packet are in the system while one high priority packet is being served.

$$f_h(L=l) = \begin{cases} x_{00} + \sum_{j=1}^Q x_{0j}^v + \sum_{j=1}^Q x_{0j}(2) + \sum_{j=0}^Q x_{1j}(1) & l=0 \\ \sum_{j=0}^Q x_{lj}^v + \sum_{j=0}^Q x_{lj}(2) + \sum_{j=0}^Q x_{l+1j}(1) & 1 \leq l \leq Q \end{cases} \quad (11)$$

$$f_l(L=l) = \begin{cases} x_{00} + \sum_{i=1}^Q x_{i0} + x_{01}(2) & l=0 \\ \sum_{j=1}^Q x_{il} + x_{0l}^v + x_{0l+1}(2) & 1 \leq l \leq Q \end{cases} \quad (12)$$

Probability of the queue length at the end of the TXOP duration can be calculated in the similar manner.

4.2. Packet Loss Rate

Packet loss occurs whenever a new packet arrives and the target buffer is full. These conditions can happen during service processing (at any time slots of the TXOP duration) or vacation.

The high priority packet will be lost when the number of packets in the high priority queue is Q (regardless of the number of packets in the low priority queue) and a new high priority packet arrives (by itself or together with a low priority packet) ($D_{11} + D_2$). Therefore, the packet loss probability will be sum of all possible probabilities among vacation and service period satisfying above conditions. As an example,

$\lambda_1^{-1} \sum_{j=0}^Q \sum_{k=1}^{T-1} x_{Qj}^k(1) [(D_{11} + D_2) \otimes (S_1)]$ shows sum of the probabilities in which the server stays on the high priority processing while a new high priority packet arrives and the other mentioned conditions has been satisfied. Consequently the high priority Packet loss rate which is normalized with the high priority arrival rate (λ_1) is expressed as follow.

$$P_{Lh} = \lambda_1^{-1} \sum_{j=0}^{Q_2} x_{Q_1j}^v [(D_{11} + D_2) \otimes (V + V^0 \beta_1)] \\ + \lambda_1^{-1} \sum_{j=0}^{Q_2} \sum_{k=1}^{T-1} x_{Q_1j}^k(1) [(D_{11} + D_2) \otimes (S_1)] \\ + \lambda_1^{-1} \sum_{j=0}^{Q_2} x_{Q_1j}^T(1) [(D_{11} + D_2) \otimes e_{m_1} \delta] \\ + \lambda_1^{-1} \sum_{j=0}^{Q_2} \sum_{k=1}^{T-1} x_{Q_1j}^k(2) [(D_{11} + D_2) \otimes (S_2)] \quad (13) \\ + \lambda_1^{-1} \sum_{j=0}^{Q_2} x_{Q_1j}^T(2) [(D_{11} + D_2) \otimes e_{m_2} \delta] \\ + \lambda_1^{-1} \sum_{j=0}^{Q_2} \sum_{k=1}^{T-1} x_{Q_1j}^k(2) [(D_{11} + D_2) \otimes (S_2^0 \beta_1)] \\ + \lambda_1^{-1} \sum_{j=0}^{Q_2} x_{Q_1j}^T(2) [(D_{11} + D_2) \otimes S_2^0 \delta]$$

The other elements are given in Appendix B. Finally, the probability vector after elapsing n+1 time slot will be

$$z^{n+1} = z^n P_h \quad (20)$$

where $z^0 = z$. To reduction of computation, the set of following equation can be used.

$$\begin{cases} z_0^{n+1} = z_0^n \tilde{B}_{00} + z_1^n \tilde{B}_{10} \\ z_i^{n+1} = z_i^n \tilde{A}_0 + z_{i+1}^n \tilde{A}_{-1} \end{cases} \quad (21)$$

Finally, let W_{HT} be the probability that the waiting time of high priority packet is less than or equal to T, then

$$W_{HT} = z_0^T P_h \quad (22)$$

where z_0^T is the probability vector that the arrived high priority packet finding no packet a head of it after T slot.

The low priority access delay is calculated in Appendix C.

5. Numerical and Simulation Results

In this section, first we provide a simple example of wireless multimedia communications to demonstrate how can apply the computational algorithm.

It is assumed that the wireless network can transmit a fixed size data block during one time slot, and each packet is segmented into a number of data blocks. Suppose a station can transmit voice and video traffic. Furthermore, the priority of voice traffic is higher than the video traffic. Voice traffic is modeled by an ON/OFF source as depicted in Figure 4a. Therefore, $D_0(1)$ and $D_1(1)$ can be calculated as follows [15].

$$D_1(1) = R * D \quad , \quad D_0(1) = (I - R) * D$$

$$D = \begin{bmatrix} 1 - \gamma_1 & \gamma_1 \\ \mu_1 & 1 - \mu_1 \end{bmatrix} \quad R = \begin{bmatrix} 0 & 0 \\ 0 & A \end{bmatrix}$$

where A is the probability of the packet arrival per time slot. Now assume that the voice packet length is fixed and is three times more than data block size. Therefore,

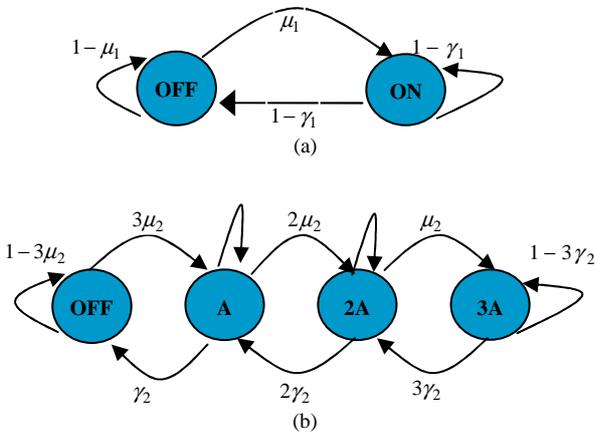


Figure 4. ON/OFF traffic model for Voice and VBR traffic.

$$S_1 = H_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad S_1^0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \beta_1 = [1 \quad 0 \quad 0]$$

The VBR traffic is modeled by three independent ON/OFF sources as showed in Figure 4b. D , $D_0(2)$ and $D_1(2)$ can be easily calculated like voice traffic matrices. Readers to find more details can refer to [15]. If we assume that the maximum video packets size is 8 times more than data block size, and video packets size follow log-normal distribution with a probability mass function (p_v) in terms of the number of data blocks such a following example [3]

$$p_v = [0.002 \quad 0.153 \quad 0.427 \quad 0.286 \quad 0.099 \quad 0.025 \quad 0.006 \quad 0.002]$$

The S_2, S_2^0 and β_2 will be

$$S_2 = H_8 \quad S_2^0 = e - S_2 e$$

$$\beta_2 = [0.002 \quad 0.006 \quad 0.025 \quad 0.099 \quad 0.286 \quad 0.427 \quad 0.153 \quad 0.002]$$

Let SI and TXOP durations be 100 and 10 slots, respectively. Then,

$$V = H_{100} \quad , \quad V^0 = e - V e \quad \delta_i = e_i(100) \quad 1 \leq i \leq 11 \quad T=10$$

Using the above information, one can easily find out system performance through the introduced model.

Now, according to the 802.11e and characteristics of the applied traffic streams which are described as follow, the numerical results obtained from the analytical model are compared with simulation results. We analyze the queue length and the access delay distribution as well as the packet loss rate for the high priority and the low priority packets. Similar to above example, one can easily match the proposed model with the introduced traffic and system parameters.

It is assumed that the voice traffic is handled with a higher priority than video traffic. The voice traffic is modeled by an ON/OFF source which generates 160 octet message periodically with a bit rate 64 kb/s during active period. The CBR video traffic has only a ON state and always stays in that. The VBR traffic is modeled by three independent ON/OFF sources with the mean bit rate 200 Kb/s. however, the PhFit program [16] can be used to find out the phase type distribution of service times of real traffic. Table 1 summarizes the different traffic used for the analytical analysis and simulations. It is assumed that the queue buffer size is seven and the channel data rate is 12Mbps.

Simulations are performed using program which is written in C++ medium. There are two queues in each station, and the server processes packets in each of the queues in the FIFO fashion. There are ten stations which are communicating with the access point. All stations enjoy ON/OFF voice traffic as high priority traffic. Five

stations generate the CBR video traffic and the others send the VBR video traffic as low priority traffic. Arrivals to the queues are independent of whether the server is in service or on vacation. The TXOPs duration are calculated through Eqn (1), according to the traffic information in each station. Each station can transmit its data during its TXOP period.

Figure 5 shows cumulative distribution function (CDF) of the high priority and the low priority queue length for VBR, CBR video and voice traffic streams. Although the CBR packet arrival rate is much larger than that of the VBR traffic stream, the queue length in the CBR traffic is less than that one in the VBR traffic considerably. As it is obvious from the mentioned figure, the probability that the length of the queue get less than or equal to six is about 98 percent for the CBR traffic while that is about 74 percent for the VBR traffic. It means that most of the time the VBR packets remain in the queue and unable to be transmitted. Therefore, the packet loss goes up and lots of packets drop. Simulation and numerical results show that the packet loss rate is about 28 percent for the VBR traffic while it is about 0.6 percent for the CBR video traffic. Consequently, although there is enough bandwidth to support QoS guarantee in the VBR video traffic but the scheduler is unable to use it.

The CDF of the queue length at the end of the TXOP for all traffic, plotted in Figure 6, confirms created challenges through the VBR traffic. Therefore, the modification of the scheduling algorithm and introduction of a dynamic scheduler to adapt with the bursty arrivals are unavoidable. Dynamic conditions in the scheduler can be obtained through adjusting the TXOP and the SI durations based on the packet queue length statistics. Scheduler can get the information from the stations and find the optimal TXOP and SI through the employing the model to maintain an empty queue at the end of TXOP duration.

Figure 7 shows CDF of the access delay and packet blocking in the high priority and the low priority traffic through analysis and simulations. It is observed that all packets in the CBR video traffic experience access delay less than about 35 ms, while only 23 percent of the VBR video packets experience such a cumulative access delay. Although there is enough bandwidth to serving the VBR traffic, the scheduler does not have essential flexibility to support of bursty arrival rate. Consequently, the queue will be full and 28 percent of arrived packets are blocked and dropped.

Finally, from Figures 5-7, it can be readily seen that the validation of analytical model is confirmed by the numerical results obtained from analytical model and the simulation results under the same conditions.

Table 1. Description of different traffic streams.

Application	Arrival rate(Kb/s)	Packet size(byte)
Voice	64	160
VBR video	200	660
CBR video	3200	910

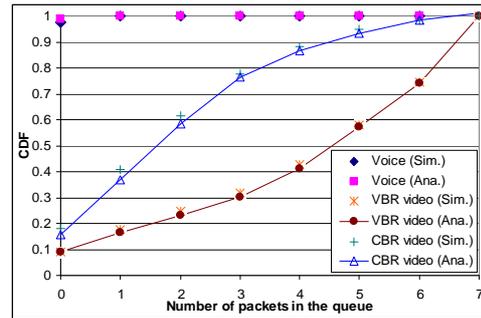


Figure 5. CDF of packet queue length for different traffic streams (Simulation (Sim.) & Analytical (Ana.)).

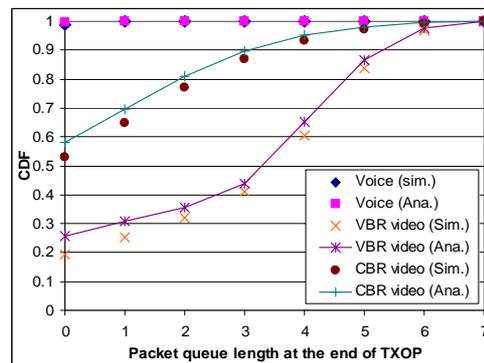


Figure 6. CDF of packet queue length for different traffic streams at the end of TXOP (Simulation (Sim.) & Analytical (Ana.)).

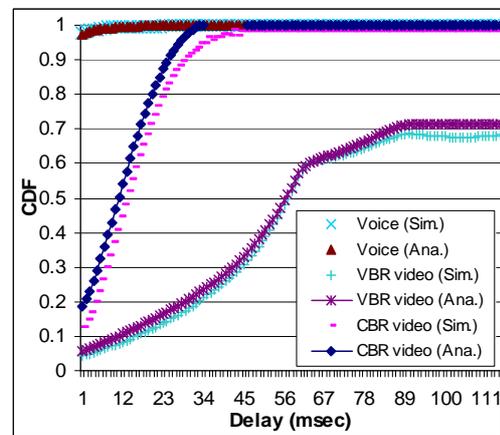


Figure 7. CDF of access delay for different traffic streams (Simulation (Sim.) & Analytical (Ana.)).

6. Conclusions

The scheduling algorithm which is introduced by HCCA /IEEE802.11e to support QoS in multimedia applications enjoys separated queues with specified priority levels and transmission opportunity according to the traffic streams characteristics. The transmission opportunity is found out based on the mean values. Therefore, some multimedia traffic streams such as VBR traffic address

some challenges in this medium. Consequently, adapting algorithms to new conditions in order to provide desired QoS is on the focus of researchers. To investigate and improve the scheduler, analytical model is very useful. This paper introduced a priority queuing model for the HCCA. Using of the MAP/PH/1 queue makes the model more comprehensive and provides it to support different practical traffic streams. The important performance measures in the high priority and the low priority queues are calculated which enable us to investigate the effect of the SI and the TXOP durations on QoS guarantees and find out the optimal TXOP values according to the queue length and the access delay statistics to provide QoS. It is shown by the numerical and the simulation results that the analytical model is quite accurate, and thus useful in the practical system design and performance evaluation.

7. Acknowledgments

This work was supported in part by the National Science Foundation of China (NSFC) under the Grant No. 90604035, the 111 Project under the grant No.111-2-14, the National 863 High-Tech R&D Program under the grant No.2007AA01Z228.

8. References

- [1] IEEE 802.11e/D13.0, Part 11, Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Medium access control (MAC) enhancements for quality of service (QoS), draft supplement to IEEE 802.11 Std, January 2005.
- [2] M. F. Neuts, "Matrix geometric solutions in stochastic models—an algorithmic approach," John Hopkins University Press Baltimore, MD, 1981.
- [3] J. A. Zhao, B. Li, X. Cao, and I. Ahmad, "Matrix-analytic solution for DMAP/PH/1 priority queue," *Queueing Systems Journal*, Springer Netherlands, Vol. 53, No. 3, pp. 127–145, July 2006.
- [4] S. Mangold, S. Choi, G. R. Hiertz, O. Klein, and B. Walke, "Analysis of IEEE 802.11e for QoS support in wireless LANs," *IEEE Wireless Communications*, pp. 40–50, December 2003.
- [5] P. Ansel, Q. Ni, and T. Turetli, "FHCF: A Simple and an efficient scheduling scheme for IEEE 802.11e wireless LAN," *Mobile Networks and Applications*, No. 11, pp. 391–403, April 2006.
- [6] A. Griolo, M. Macedo and M. Nunes, "A Scheduling Algorithm for QoS Support in IEEE802.11e Networks," *IEEE Wireless Communications*, pp. 36–43, June 2003.
- [7] A. Iera, A. Molinaro, G. Ruggeri, and D. Tripodi, "Improving QoS and throughput in single and multihop WLANs through dynamic traffic prioritization," *IEEE Network*, pp. 35–44, June 2005.
- [8] N. Vaidya, A. Dugar, S. Gupta, and P. Bahl, "Distributed fair scheduling in a wireless LAN," *IEEE Transactions on Mobile Computing*, Vol. 4, No. 6, pp. 616–629, November 2005.
- [9] H. Zhai, X. Chen and Y. Fang, "How well can the IEEE 802.11 wireless LAN support quality of service?" *IEEE Transactions on Wireless Communications*, Vol. 4, No. 6, pp. 3084–3093, November 2005.
- [10] Z. Kong, D. Tsang and B. Bensaou, "Performance analysis of IEEE802.11e contention-based channel access," *IEEE Journal on Selected Areas in Communications*, Vol. 22, No. 10, pp. 2095–2106, December 2004.
- [11] Y. Xiao, "Performance analysis of priority schemes for IEEE 802.11 and IEEE 802.11e wireless LANs," *IEEE Transactions on Wireless Communications*, Vol. 4, No. 4, pp. 1506–1515, July 2005.
- [12] H. Zhu and I. Chlamtac, "Performance analysis for IEEE 802.11e EDCF service differentiation," *IEEE Transactions on Wireless Communications*, Vol. 4, No.4, pp. 1779–1788, July 2005.
- [13] M. M. Rashid and E. Hossain, "Queuing analysis of 802.11e HCCA with variable bit rate traffic," *IEEE International Conference on Communications*, Vol. 10, pp. 4792–4798, 2006.
- [14] A. S. Alfa, "Matrix geometric solution of discrete time MAP/PH/1 priority queue," *Naval Research Logistics*, Vol. 45, pp. 23–50, July 1997.
- [15] C. Blondia and O. Casals, "Performance analysis of statistical multiplication of VBR sources," *Conference of the IEEE Computer and Communications Societies*, pp. 828–838, 1992.
- [16] Horvath and M. Telek, "Phfit: A general phase type fitting tool," in *proceedings of 12th performance TOOLS*, Lecture Notes in Computer Science, Vol. 2324, London, UK, April 2002.

Appendix B

State probability vectors in the high priority access delay (Section 4.3)

$$\begin{aligned}
 z_{0H}^1 &= \lambda^{-1} \sum_{j=1}^{Q_2} x_{0j}^v (D_{11} + D_2) \otimes V^0 \\
 z_{0H}^k &= \lambda_1^{-1} \sum_{j=1}^{Q_2} x_{0j}^{k-1} (2)(D_{11} + D_2) \otimes S_2^0 \\
 &\quad + \lambda_1^{-1} \sum_{j=0}^{Q_2} x_{1j}^{k-1} (1)(D_{11} + D_2) \otimes S_1^0 \quad 2 \leq k \leq T \\
 z_i^v &= \lambda_1^{-1} \sum_{j=0}^{Q_2} x_{ij}^v (D_{11} + D_2) \otimes V \\
 &\quad + \lambda_1^{-1} \sum_{j=0}^{Q_2} x_{ij}^T (1)(D_{11} + D_2) \otimes e_{m_1} \delta \\
 &\quad + \lambda_1^{-1} \sum_{j=0}^{Q_2} x_{i+1j}^T (1)(D_{11} + D_2) \otimes S_1^0 \delta \\
 &\quad + \lambda_1^{-1} \sum_{j=0}^{Q_2} x_{ij}^T (2)(D_{11} + D_2) \otimes (S_2^0 \delta + e_{m_2} \delta) \\
 z_{iL}^k &= \lambda_1^{-1} \sum_{j=0}^{Q_2} x_{ij}^{k-1} (2)(D_{11} + D_2) \otimes S_2 \quad 2 \leq k \leq T \\
 z_{0H}^1 &= \lambda_1^{-1} \sum_{j=0}^{Q_2} x_{ij}^v (D_{11} + D_2) \otimes V^0 \beta_1 \\
 z_{iH}^k &= \lambda_1^{-1} \sum_{j=1}^{Q_2} x_{ij}^{k-1} (1)(D_{11} + D_2) \otimes S_1 \\
 &\quad + \lambda_1^{-1} \sum_{j=0}^{Q_2} x_{i+1j}^{k-1} (1)(D_{11} + D_2) \otimes S_1^0 \beta_1 \\
 &\quad + \lambda_1^{-1} \sum_{j=0}^{Q_2} x_{ij}^{k-1} (2)(D_{11} + D_2) \otimes S_2^0 \beta_1 \quad 2 \leq k \leq T \quad 1 \leq i < Q_1
 \end{aligned}$$

Block matrices of transition matrix P_h

$$\begin{aligned}
 \tilde{A}_0 &= \begin{bmatrix} I \otimes V & 0 & e_1'(T) \otimes I \otimes V^0 \beta_1 \\ \tilde{A}_0^{Lv} & H_T \otimes I \otimes S_2 & H_T \otimes I \otimes S_2^0 \beta_1 \\ \tilde{A}_0^{Hv} & 0 & H_T \otimes I \otimes S_1 \end{bmatrix} \\
 \tilde{A}_0^{Lv} &= e_T(T) \otimes [I \otimes e_{m_2} \delta + I \otimes S_2^0 \delta] \\
 \tilde{A}_0^{Hv} &= e_T(T) \otimes I \otimes e_{m_1} \delta \\
 \tilde{B}_{00} &= \begin{bmatrix} I \otimes V & 0 & e_1'(T) \otimes I \otimes V^0 \\ \tilde{B}_{00}^{Lv} & H_T \otimes I \otimes S_2 & H_T \otimes I \otimes S_2^0 \\ \tilde{B}_{00}^{Hv} & 0 & H_T \otimes I \end{bmatrix} \\
 \tilde{B}_{00}^{Lv} &= e_T(T) \otimes [I \otimes e_{m_2} \delta + I \otimes S_2^0 \delta] \\
 \tilde{B}_{00}^{Hv} &= e_T(T) \otimes I \otimes e_{m_1} \delta
 \end{aligned}$$

$$\tilde{A}_{-1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ e_T(T) \otimes I \otimes S_1^0 \delta & 0 & H_T \otimes I \otimes S_1^0 \beta_1 \end{bmatrix}$$

Appendix C

Access delay Distribution for the low priority packets (Section 4.3)

The arriving low priority packet will reach the head of its queue and will be ready to transmit if all the low priority packets ahead of it are served. Since the low priority packet is unable to be transmitted while there are high priority packets in the system, the arriving low priority packet has to wait for completion of the transmission of all high priority and low priority packets which are in the system and those high priority packets which will enter during the period of the time that the arriving packet moves towards head of queue. Therefore, the number of packet ahead of arriving low priority packet at its arrival time is all high priority and low priority packets in the system (including any high priority packet which might have arrived jointly with it). Suppose y defines initial probability vector in the low priority access delay.

$$y = [y_0 \ y_1 \ \dots \ y_{Q_1}] \quad \text{where} \quad y_i = [y_{i0} \ y_{i1} \ \dots \ y_{iQ_2}]$$

$$y_{ij} = [y_{ij}^v \ y_{ijL}^1 \ \dots \ y_{ijL}^T \ y_{ijH}^1 \ \dots \ y_{ijH}^T]$$

where y_{ij}^v , represents probability of an arriving low priority packet finding i high priority and j low priority packet in the system with the server on vacation and $y_{ijL}^k (y_{ijH}^k) \ 1 \leq k \leq T$ is the probability of an arriving low priority packet finding i high priority and j low priority packet in the system with the server in the low priority (high priority) processing at the k th slot of TXOP. All the probability vectors can be calculated in a similar manner which has discussed in the Section 4.3.

$$y_{00}^v = \lambda_2^{-1} \{x_{00}[D_{12} \otimes V + V^0 \delta] + x_{10}^T(1)[D_{12} \otimes S_1^0 \delta]$$

$$+ x_{01}^T(2)[D_{12} \otimes S_2^0 \delta] + \sum_{k=2}^T x_{10}^{k-1}(1)[D_{12} \otimes S_1^0 \delta_k]$$

$$+ \sum_{k=2}^T x_{01}^{k-1}(2)[D_{12} \otimes S_2^0 \delta_k \}$$

$$y_{0j}^v = \lambda_2^{-1} \{x_{0j+1}^T(2)[D_{12} \otimes S_2^0 \delta] + x_{0j}^T(2)[D_{12} \otimes e_{m_2} \delta] + x_{1j}^T(1)[D_{12} \otimes S_1^0 \delta] + x_{0j}^T[D_{12} \otimes V]\} \quad 1 \leq j < Q_2$$

$$y_{0jL}^1 = \lambda_2^{-1} \{x_{0j}^v[D_{12} \otimes V^0 \beta_2]\} \quad 1 \leq j < Q_2$$

$$y_{0jL}^k = \lambda_2^{-1} \{x_{0j}^{k-1}[D_{12} \otimes S_2]\} \quad 1 \leq j < Q_2 \quad 2 \leq k \leq T$$

$$y_{1j}^v = \lambda_2^{-1} \{x_{1j}^v[D_{12} \otimes V] + x_{1j}^T(1)[D_{12} \otimes e_{m_1} \delta]$$

$$+ x_{1j}^T(1)[D_2 \otimes S_1^0 \delta] + x_{1j}^T(2)[D_{12} \otimes S_2^0 \delta]$$

$$+ x_{2j}^T(1)[D_{12} \otimes S_1^0 \delta] + x_{0j+1}^T(1)[D_2 \otimes S_2^0 \delta]$$

$$+ x_{0j}^v[D_2 \otimes V] + x_{1j}^T(2)[D_{12} \otimes e_{m_2} \delta]$$

$$+ x_{0j+1}^T(2)[D_2 \otimes e_{m_2} \delta]\} \quad 0 \leq j < Q_2$$

$$\tilde{A}_{-1}^0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ e_T(T) \otimes (D_{00} + D_{12}) \otimes S_1^0 \delta & 0 & H_T \otimes (D_{00} + D_{12}) \otimes S_1^0 \beta_1 \end{bmatrix}$$

$$\begin{cases} y_0^{n+1} = y_0^n \tilde{B}_{00} + y_1^n \tilde{B}_{10} \\ y_1^{n+1} = y_0^n \tilde{B}_{01} + y_1^n \tilde{A}_0 + y_2^n \tilde{A}_{-1} \\ y_i^{n+1} = y_{i-1}^n \tilde{A}_1 + y_i^n \tilde{A}_0 + y_{i+1}^n \tilde{A}_{-1} \quad i \geq 2 \end{cases}$$

Suppose $y^0 = y$. Then, $y^{n+1} = y^n P_l$

To reduction of computation, the set of following equation can be used.

Finally, let W_{LT} be the probability that the waiting time of low priority packet is less than or equal to T. Then, $W_{LT} = y_{00}^T P_l$.

Gateways Placement in Backbone Wireless Mesh Networks

Maolin TANG

Queensland University of Technology, Brisbane, QLD, Australia

Email: m.tang@qut.edu.au

Received October 6, 2008; revised November 5, 2008; accepted December 11, 2008

Abstract

This paper presents a novel algorithm for the gateway placement problem in Backbone Wireless Mesh Networks (BWMNs). Different from existing algorithms, the new algorithm incrementally identifies gateways and assigns mesh routers to identified gateways. The new algorithm can guarantee to find a feasible gateway placement satisfying Quality-of-Service (QoS) constraints, including delay constraint, relay load constraint and gateway capacity constraint. Experimental results show that its performance is as good as that of the best of existing algorithms for the gateway placement problem. But, the new algorithm can be used for BWMNs that do not form one connected component, and it is easy to implement and use.

Keywords: Gateway Placement, Backbone Wireless Mesh Networks

1. Introduction

A wireless mesh network is an ad hoc communication network that is made up of wireless communication nodes organized in a mesh topology. It allows for continuous connections and reconfiguration around broken or blocked paths by hopping from node to node until the destination is reached. In a wireless mesh network communication nodes can connect to each other via multiple hops and they are not mobile. The infrastructure that supports a mesh wireless network is a wireless router network, or backbone wireless mesh network (BWMN).

A BWMN consists of a collection of wireless mesh routers, each of which can communicate with other wireless mesh routers and clients. Each mesh router forwards packages on behalf of other mesh routers and clients. The wireless mesh routers are generally not mobile. The clients connect to the wireless network through a wireless mesh router and they do not have restriction on mobility.

Gateway placement is an important problem in the design of BWMNs. It determines network points, or gateways, through which a BWMN communicates with other networks. The objective is to minimize the total number of gateways subject to Quality-of-Service (QoS) constraints. There are three popular QoS constraints in the design of BWMNs: *delay constraint*, *relay load constraint* and *gateway capacity constraint*.

A BWMN is a multi-hop network where significant delay occurs at each hop due to contention for the

wireless channel, packet processing, and packet queuing. The delay is therefore a function of the number of communication hops between the mesh router and its gateway [1]. The delay constraint requires that the maximal number of hops from any mesh router to a gateway is not greater than a given number R . In the placement of BWMN gateways, it is important to guarantee the throughput for individual traffic flows. Since it is assumed in this research that a BWMN has multiple communication channels, which allow interfering wireless links operate on different communication channels concurrently, the bottleneck on throughput is therefore reduced to the load on the link individual links between wireless routers as relay load L . In addition, the throughput of a BWMN depends on the bandwidth and processing speed of the gateways. Thus, a gateway has a capacity S , which is measured by the maximal number of mesh routers that it can serve.

This paper presents a new algorithm, namely incremental clustering algorithm, for the gateway placement problem. Compared with existing algorithms for the gateway placement problem, the new algorithm has the following advantages: first, it guarantees to find a gateway placement satisfying all the constraints; second, it has competitive performance; third, it can be used for the BWMNs that does not form a connected component; fourth, it is easy to implement and use.

The remaining paper is organized as follows. The following section formulates the BWMN gateway placement problem, which is followed by a discussion of related work. Then, we discuss our incremental clustering

algorithm and show our experimental results on our incremental clustering algorithm. Finally, we conclude the incremental clustering algorithm.

2. Problem Formulation

A BWMN can be represented by a directed graph $G = (V, E)$. Each node $v = \langle x, y, r \rangle \in V$ represents a mesh router, where x and y are the x-coordinate and y-coordinate of the location of v and r is the radius of the circular transmission range of v . Arc $\langle v_i, v_j \rangle \in E$ if and only if mesh router v_j is in the transmission range of mesh router v_i , or $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq r_i$, where $v_i = \langle x_i, y_i, r_i \rangle$ and $v_j = \langle x_j, y_j, r_j \rangle$. Note that $\langle v_i, v_j \rangle \in E$ does not implies $\langle v_j, v_i \rangle \in E$ because the radiuses of their transmission range may be different.

A mesh cluster is a set of vertices $C \subseteq V$. A mesh cluster has a cluster head $h \in C$. The nodes in C and the arcs between them define a *cluster graph* $G_C = (C, E_C)$, where an arc $\langle v_i, v_j \rangle \in E_C$ if and only if $v_i \in C$, $v_j \in C$, and $\langle v_i, v_j \rangle \in E$. A mesh cluster is connected if and if only the corresponding cluster graph is connected. The radius of a mesh cluster r_C is defined as the maximal shortest distance between from the mesh cluster head h and the nodes in C . The delay constraint is translated into an upper bound R on the mesh cluster radius.

The *shortest path spanning tree* is a spanning tree of G_C , $T(G_C)$, which is formed by composing the shortest paths from the cluster head h to all the other nodes in C . The nodes at i^{th} level of the shortest path spanning tree have i hops to the cluster head h . The depth of $T(G_C)$ is denoted $d(T(G_C))$. Let v be a node in $T(G_C)$. The number of nodes in the subtree rooted v is denoted $\pi(v)$.

Given a BWMN represented by a directed graph $G = (V, E)$, a delay constraint R , a relay load constraint L and a gateway capacity constraint S , the BWMN gateway placement problem is to find a set of connected clusters $\{C_1, C_2, \dots, C_n\}$ and their corresponding clusters' shortest path spanning trees such that n is minimal subject to

- (a) $C_1 \cup C_2 \cup \dots \cup C_n = V$;
- (b) $|C_k| \leq S$, where $1 \leq k \leq n$;
- (c) $d(C_k) \leq R$, where $1 \leq k \leq n$;
- (d) $\forall v \in T(G_{C_k}), \pi(v) \leq L$.

The shortest path spanning trees give a gateway placement solution where the roots represent the mesh router where a gateway is placed and the links specify the communication topology.

Condition (a) guarantees that a BWMN gateway placement solution covers all mesh routers; Condition (b) ensures that the gateway capacity constraint S is satisfied; Condition (c) enforces that the delay constraint R is met; Condition (d) makes sure that the relay load constraint L is respected.

3. Related Work

From the computational point of view, the gateway placement problem is conjectured as an NP-hard problem as it can be transformed into the minimum dominating set problem [1], which has been proven to be NP-complete [2]. Thus, optimal algorithms are not suitable for solving the problem as they would lead to combinatorial explosion in the search space when the problem size is large. Because of the reason, all existing algorithms for the gateway placement problem are heuristic or approximation ones.

In [3], Wong, *et al.* addressed two gateway placement problems: one is to optimize the communication delay and another is to optimize the communication cost. Although the algorithms can be extended to consider delay, relay load, and gateway capacity constraints, they can only be used for BWMNs that form a connected component and require at least two hops for communication between at least one pair of nodes.

The algorithm proposed by Bejerano in [4] uses a clustering technique and performs the gateway placement problem in four stages: select cluster heads, assign each node to an identified cluster satisfying the delay constraint, break down the clusters that do not satisfy the relay load constraint or the gateway capacity constraint, and finally select gateways to reduce the maximum relay load. However, the algorithm does not have competitive performance because of the following two reasons: first, when identifying cluster heads and assigning mesh routers to the identified cluster heads, the algorithm does not make use of global information about the BWMN; second, splitting a cluster without considering re-assigning those mesh routers to existing clusters may create some unnecessary clusters and therefore increases the number of clusters significantly.

In [5], Chandra, *et al.* explored the placement problem of Internet Transit Access Points (ITAPs) in wireless neighborhood networks under three wireless link models, and for each of the wireless link models, they developed algorithms for the placement problem based on neighborhood layouts, user demands, and wireless link characteristics. The placement problem is similar to the gateway place of BWMN. However, their algorithms consider only one constraint, that is, users' bandwidth requirements.

The work closest to ours is the algorithm proposed by Aoun, *et al.* in [1], which transforms the gateway placement problem into the minimum dominating set problem and adopts a recursive dominating set algorithm to tackle the minimum dominating set problem. The algorithm considers the delay, relay load and gateway constraints and has better performance than the Wong's algorithms, the Bejerano's algorithms, and the Chandra's algorithms. However, it has the following deficiencies: first, it can be used for those BWMNs that form a connected component; second, it needs to set the initial

radius size properly; otherwise, it would not create satisfactory results.

4. The Incremental Clustering Algorithm

4.1. Preliminaries

In this paper, the *transitive closure* of a directed graph $G = (V, E)$ is a directed graph $G^+ = (V, E^+)$ such that for $\forall \langle u, v \rangle \in E^+$ if and only if there exists a non-null path from u to v . The *n-step transitive closure* of a directed graph $G = (V, E)$ is a directed graph $G^n = (V, E^n)$ such that for $\forall \langle u, v \rangle \in E^n$ if and only if there exists a non-null path from u to v and the length of the path is less than or equals to n . Figure 1 shows a BWMN graph. The transitive closure and the 2-step transitive closure are displayed in Figure 2 and Figure 3 respectively.

A BWMN graph $G = (V, E)$ can be represented by an $n \times n$ adjacency matrix $A = [a_{ij}]_{n \times n}$, where

$$a_{ij} = \begin{cases} 1, & \text{if } v_i, v_j \in V \text{ and } \langle v_i, v_j \rangle \in E; \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

For example, for the BWMN graph shown in Figure 1, its adjacency matrix is shown in Equation 2. The adjacent matrix representations for its transitive closure and its 2-step transitive closure are displayed in Equation 3 and Equation 4 respectively.

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (2)$$

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (3)$$

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (4)$$



Figure 1. A BWMN graph G.

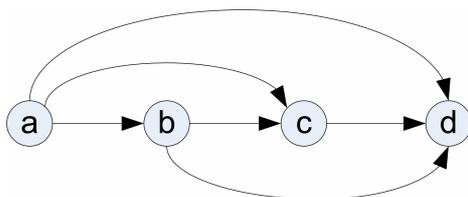


Figure 2. The transitive closure of G.

4.2. Algorithm Description

The incremental clustering algorithm solves the gateway placement problem by iteratively and incrementally identifying gateways and assigning mesh routers to identified gateways.

Algorithm 1 is the algorithm description.

Algorithm 1 Incremental clustering algorithm

```

while  $U \neq \emptyset$  do
    construct a BWMN graph from  $U$ ;
    build the  $R$ -step transitive closure from the BWMN graph;
    identify gateways from the  $R$ -step transitive closure;
    assign mesh routers in  $U$  to identified gateways subject to the  $R, L$  and  $S$  constraints;
    remove the assigned mesh routers from  $U$ .
end while
    
```

In Algorithm 1, U is the set of mesh routers; R, L and S represent the delay constraint, relay load constraint and the gateway capacity constraint, respectively.

The incremental clustering algorithm is an iterative one. In each iteration, it starts with constructing a BWMN graph from the current unassigned mesh router set U , and then builds the R -step transitive closure from the BWMN graph, and then identifies gateways based on the R -step transitive closure, and finally assigns mesh routers to the identified gateways and removes the assigned mesh routers from U . The process is repeated until U is empty. By the time U is empty, every mesh router has been assigned to a gateway. This algorithm is incremental as it incrementally identifies gateways and assigns mesh routers to identified gateways, rather than identifying all gateways and assigning all mesh routers to the gateways in one step. Since the construction of a BWMN graph has been already introduced in the previous subsection and the algorithm for building an R -step transitive closure is well-known, we focus on discussing how to identify gateways from the R -step transitive closure and how to assign mesh routers to identified gateways in the following.

It can be observed that the i^{th} row of the R -step transitive closure is a cluster, representing a set of mesh routers that can be covered by the i^{th} mesh router, where $1 \leq i \leq |U|$. The i^{th} mesh router is the head of the mesh cluster. The mesh router clusters can be classified into *covered clusters* and *uncovered clusters*. A mesh cluster is a uncovered one if there exists one mesh router in the mesh cluster that is not present in the other mesh clusters; Otherwise, the mesh cluster is a covered one. It can be observed that there is one and only one mesh router that cannot be covered by the other mesh cluster in a uncovered cluster, which is the head.

For each uncovered mesh cluster, at least one gateway is needed as the head of the mesh cluster cannot use any mesh router in other mesh clusters as its gateway because it cannot be covered by any other mesh routers

in the other clusters. Thus, we select the head of a uncovered cluster as a gateway. However, sometimes there is no uncovered mesh cluster (we will give an example when illustrating the algorithm later). If this is the case, we select the head of the mesh cluster that covers the maximal number of mesh routers as a gateway. Algorithm 2 is the algorithm for identifying gateway.

Algorithm 2 Identifying gateways

```

for  $i=1$  to  $|U|$  do
  if the corresponding mesh router cluster of the  $i^{th}$ 
  row of the  $R$ -step transitive closure is a uncovered
  mesh cluster
  then
    the head of the mesh router cluster is selected
    as a gateway;
  end if
end for
if no uncovered mesh cluster was found then
  find a mesh router cluster has the maximal size;
  the head of the mesh router cluster is selected as a
  gateway.
end if

```

Once gateways have been identified using the technique described above, we assign as many mesh routers as possible to those identified gateways subject to the delay, relay load, and gateway capacity constraints to minimize the total number of gateways. Algorithm 3 is the algorithm for assigning mesh routers to identified gateways.

Algorithm 3 Assigning mesh routers to identified gateways

```

for each gateway  $g$  do
  for  $h=0$  to  $R$  do
    for any mesh router that is covered by  $g$  and the
    shortest distance to  $g$  is  $h$  do
      if not violating any of the constraints then
        assign the mesh router to  $g$ ;
        remove the mesh router from the other gateways,
        if any;
      end if
    end for
  end for
end for

```

4.3. Algorithm Analysis

The incremental clustering algorithm is iterative. In each iteration, the algorithm identifies at least one gateway, assigns at least one mesh router to an identified gateway and therefore the number of unassigned mesh routers decreases by one. Thus, the algorithm terminates after at most $n - 1$ iterations, where n is the total number of mesh routers. In addition, an assignment is accepted only when it does not violet the constraints. So, it is guaranteed that the algorithm generates a feasible solution when it terminates.

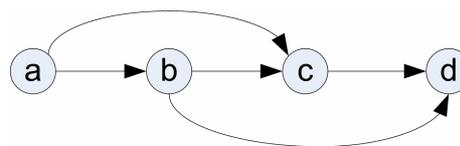


Figure 3. The 2-step transitive closure of G .

Assume that $G = (V, E)$ is the BWMN graph of a BWMN gateway placement problem. The computational complexity of the incremental clustering algorithm in the worst case is $O(R \times |V|^3 + |E| \times |V|^2)$. The following is the proof.

In the worst case, the algorithm iterates $|V|$ times. In each of the iterations, the algorithm identifies only one gateway and assigns only one mesh router (the mesh router at the gateway) to the gateway. Thus, the algorithm builds the BWMN graph $|V|$ times. It takes $O(|V|^2)$ time to build a BWMN graph that has $|V|$ nodes (it is assumed that the adjacent matrix representation is used.). Thus, the total computational complexity for building BWMN graphs is $O(|V|^3)$. It takes $O(R \times |V|^2)$ to construct an R -step transitive closure. In the worst case, the R -step transitive closure needs to be constructed $|V|$ times. Thus, the total computational complexity for constructing R -step transitive closures is $O(R \times |V|^3)$. In addition, given an R -step transitive closure, it takes $O(|E| \times |V|)$ time to identify a gateway in an iteration. Thus, the total computational complexity for identifying gateways is $O(|E| \times |V|^2)$. It takes at most $O(|V|)$ to assign a mesh router to a gateway (it needs to remove the assigned mesh router from the other mesh router clusters). Thus, the total computational complexity for assigning $|V|$ mesh routers is $O(|V|^2)$ in the worst case. Thus, the computational complexity in the worst case is $O(|V|^2 + R \times |V|^3 + |E| \times |V|^2 + |V|^2) = O(R \times |V|^3 + |E| \times |V|^2)$.

4.4. Algorithm Illustration

This section uses an example to illustrate how the incremental clustering algorithm works. The BWMN gateway placement problem is given in a BWMN graph shown in Figure 4. In the BWMN there are nine mesh routers that may have different coverage radiuses. For example, the coverage radius of mesh router v_8 is larger than that of mesh router v_9 . As a result, mesh router v_8 can cover mesh router v_9 , but not the other way around. Figure 5 is the matrix representation of the BWMN graph shown in Figure 4.

For this BWMN gateway placement problem, we assume that the delay constraint $R = 2$, the relay load constraint $L = 2$, the gateway capacity constraint $S = 3$. In other words, for this BWMN gateway placement problem we need to find a solution such that the maximum hop from any mesh router to its gateway must not exceed 2, every mesh router must not relay packets for more than 2 mesh routers, and each gateway must not serve for more than 3 mesh routers.

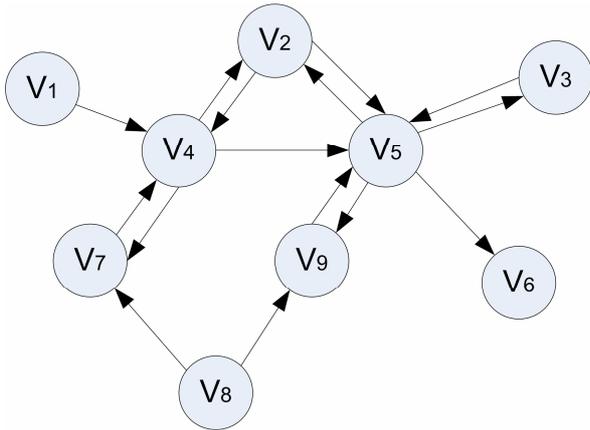


Figure 4. The BWMN graph.

The algorithm starts with finding the 2 transitive closure of the BWMN graph. Figure 6 displays the matrix representation of the 2-step transitive closure of the BWMN graph.

Then, the algorithm identifies gateways using the procedure described in Algorithm 2. Since the mesh router clusters corresponding to the 1th and the 8th rows of the 2-step transitive closure are the only uncovered mesh router clusters, v_1 and v_8 are identified as gateways. The algorithm then uses the procedure described in Algorithm 3 to assigns mesh routers in U to v_1 and v_8 as many as possible subject to the R , L and S constraints. The assigning procedure starts with v_1 .

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Figure 5. The matrix representation of the BWMN graph.

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Figure 6. The matrix representation of the 2-step transitive closure of the BWMN graph.

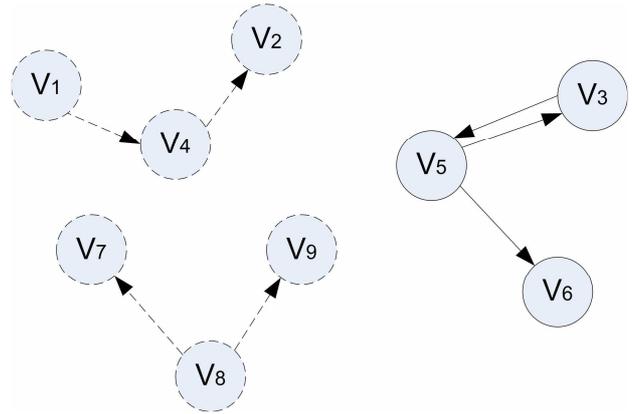


Figure 7. The intermediate state of the BWMN graph.

It considers all the mesh routers that can be covered by v_1 according to the information given in the 2-step transitive closure in Figure 6 in the descending order of the number of hops from the mesh router to v_1 . As a result, v_1 , v_4 and v_2 are assigned to gateway v_1 in the order. The assigning procedure then uses the same idea to assign mesh routers to v_8 , v_7 and v_9 to gateway v_8 . Figure 7 shows the state after this iteration of identifying gateways and assigning mesh routers. In the figure, the components drawn in broken lines represent the assigned mesh routers and the components drawn in solid lines represent the mesh routers that have not been assigned to any gateway.

Since there are still some mesh routers that have not been assigned to any gateway, the algorithm repeats the above process. It creates a BWMN graph for the remaining mesh routers and then generates a 2-step transitive closure of the BWMN graph. Figures 8 and 9 show the matrix representation of the BWMN and the 2-step transitive closure of the BWMN graph, respectively.

From the 2-step transitive closure of the BWMN graph, the algorithm identifies gateways using the procedure described in Algorithm 2. Since all the mesh router clusters are covered ones, the mesh router cluster that has the largest size, which is v_5 , is selected as a gateway. The algorithm then assigns the rest mesh routers to gateway v_5 . Figure 10 shows the final placement result. As displayed in the figure, three gateways are needed to be placed.

5. Experimental Results and Discussions

This section evaluates the performance of the incremental clustering algorithm by comparing it with three top algorithms for the gateway placement problem by simulation. The three top algorithms are the weighted recursive algorithm proposed by Aoun, *et al.* in [1], the iterative greedy algorithm proposed by Bejerano in [4], and an augmenting algorithm similar to those proposed by Wong, *et al.* in [3] and by Chabdra, *et al.* in [5]. The performance of the four algorithms are evaluated and

compared in terms of the delay constraint, the relay load constraint, and the gateway capacity constraint respectively.

We have developed a MATLAB program to randomly generate gateway placement problems. All the generated gateway placement problems have 200 mesh routers on a 10×10 plane. The connection radius is 1.0, and the minimum distance between any pair of mesh routers is 0.5. We have use the program to generate 30 instances for each of the set-ups, and have used the four algorithms to solve the gateway placement problems. The performance of the algorithms is evaluated by the average number of gateways of the 30 runs for each of the set-ups in each of the evaluations.

The implementations of the weighted recursive algorithm, the iterative greedy algorithm, and the augmenting algorithm used in the evaluations is the ones used in [1] and is kindly provided by Mr Bassam Aoun and Prof. Raouf Boutaba. However, the program used for randomly generating test problems is different from the one used in [1]. Given a parameter n , the test problem generator used in [1] randomly creates a test problem that contains up to n mesh routers, but the test problem generator used in our experiments randomly creates a test problem that has exactly n mesh routers, which makes the experimental results more accurate and reliable.

$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

Figure 8. The matrix representation of the BWMN graph.

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

Figure 9. The matrix representation of the 2 transitive closure of the BWMN graph.

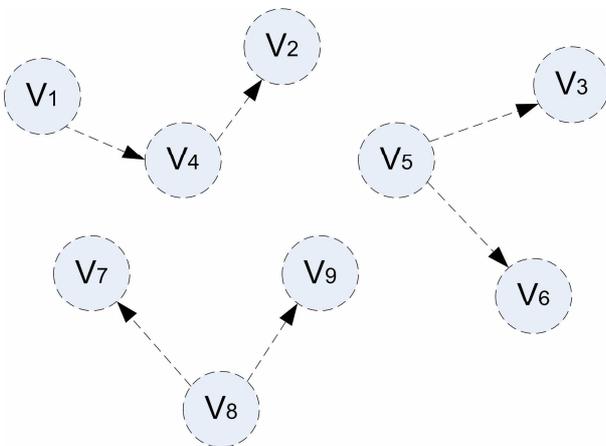


Figure 10. The solution.

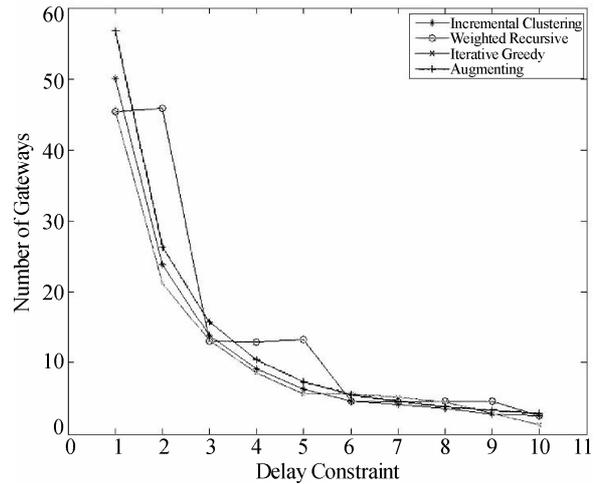


Figure 11. Comparison of the effects of the hop constraint on the four algorithms. $L = \text{NaN}$. $S = \text{NaN}$.

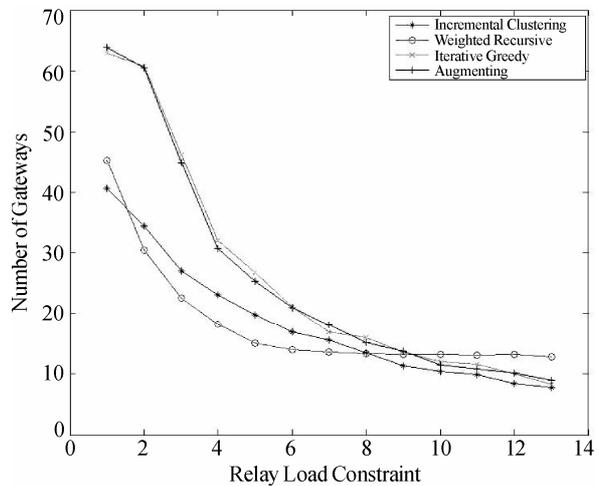


Figure 12. Comparison of the effects of the link capacity Constraint on the four algorithms. $R = 8$. $S = \text{NaN}$.

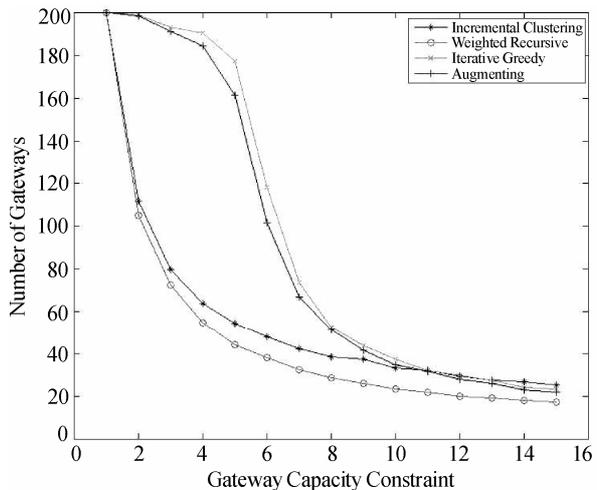


Figure 13. Comparison of the effects of the gateway constraint on the performance of the four algorithms. $R = 8$. $L = \text{NaN}$.

5.1. Effects of the Delay Constraint

The effects of the delay constraint on the performance of the four algorithms are evaluated in this section. In the evaluation, the relay load constraint and the gateway capacity constraint are relaxed. The values of the delay constraint vary from 1 to 10. Figure 11 shows the evaluation results.

It can be seen from the figure that the performance of the incremental clustering algorithm is similar to that of the iterative greedy algorithm and the augmenting algorithm, but it is better than that of the weighted recursive algorithm, under the delay constraints.

5.2. Effects of the Relay Load Constraint

This section evaluates the effects of the relay load constraint on the performance of the four algorithms. In this evaluation, the link capacity constraint varies from 1 to 13, the gateway capacity constraint is relaxed, and the delay constraint is fixed to 8. Figure 12 illustrates the evaluation results.

The evaluation results show that the performance of the incremental clustering algorithm is better than that of the iterative greedy algorithm and the augmenting algorithm. It also outperforms the weighted recursive algorithm when the relay load constraint is 1 and when the relay load constraint is greater than 8. But, it is not as good as that of the weighted recursive algorithm when the link capacity is between 2 and 8. In overall, the performance of the incremental clustering algorithm is as good as that of the weighted recursive algorithm, which is the best among the existing gateway placement algorithms, under the relay load constraints.

5.3. Effects of the Gateway Capacity Constraint

The effects of the gateway capacity constraint on the performance of the four algorithms are studied in this section. In this evaluation, we test the performance of the four algorithms when the gateway capacity constraint varies from 1 to 15. The delay constraint is set to 8 and the relay load constraint is relaxed. Figure 13 shows the performance of the four algorithms in relation to the gateway capacity constraint.

The figure shows that that the performance of the weighted recursive algorithm is the best among the four algorithms. The performance of the incremental clustering algorithm is similar to that of the weighted recursive algorithm, and it is better than that of the iterative algorithm and the augmenting algorithm when the gateway capacity constraint is tight. When the gateway capacity constraint is relaxed, the performances of the recursive clustering and assignment algorithm, the iterative greedy algorithm, and the augmenting algorithm

are close to each other.

6. Conclusions

This paper has presented a new algorithm for the gateway placement problem. Different from existing algorithms for the gateway placement problem, this new algorithm incrementally identifies gateways and assigns remaining mesh routers to the identified gateways. By incrementally identifying gateways, the new algorithm can exploit the dynamically generated information about the distribution of unassigned mesh routers; By incrementally assigning mesh routers to a gateway, the new algorithm can fully explore mesh router assignment options and therefore benefit to reduce in the number of gateways. Experimental results have shown that in overall the performance of the new algorithm is as good as that of the best of the three top algorithms, and sometimes it outperforms the best algorithm.

In addition to its good performance, the new algorithm has the following good advantages: first, it guarantees to find a gateway placement satisfying all the constraints; second, it has competitive performance; third, it can be used for the BWMNs that does not form a connected component; fourth, it is easy to implement and use.

7. Acknowledgement

The author would like to thank Mr Bassam Aoun and Prof. Raouf Boutaba for providing the source code used in their research on the gateway placement problem in [1].

8. References

- [1] B. Aoun, R. Boutaba, Y. Iraqi, and G. Kenward, "Gateway placement optimization in wireless mesh networks with QoS constraints," *IEEE Journal on Selected Areas in Communications*, Vol. 24, No. 11, pp. 2127–2136, November 2006.
- [2] M. R. Garey and D. S. Johnson, "Computers and intractability; A guide to the theory of NP-completeness," New York, NY, USA: W. H. Freeman & Co., 1990.
- [3] J. Wong, R. Jafari, and M. Potkonjak, "Gateway placement for latency and energy efficient data aggregation," in *Proceedings IEEE LCN*, pp. 490–497, 2004.
- [4] Y. Bejerano, "Efficient integration of multihop wireless and wired networks with QoS constraints," *IEEE/ACM Transactions on Networking*, Vol. 12, No. 6, pp. 1064–1078, December 2004.
- [5] R. Chandra, L. Qiu, K. Jain, and M. Mahdian, "Optimizing the placement of Internet TAPs in wireless neighborhood networks," in *Proceedings IEEE ICNP*, pp. 271–282, 2004.

Parallel-Transmission: A New Usage of Multi-Radio Diversity in Wireless Mesh Network

Yun HU¹, Shoubao YANG, Qi ZHANG, Peng ZHANG

Department of Computer Science, University of Science and Technology of China, Hefei, Anhui, China

Emails: ¹geajy@mail.ustc.edu.cn

Received September 5, 2008; revised December 1, 2008; accepted December 31, 2008

Abstract

To fully utilize the diversity of multi-radio, a new parallel transmission method for wireless mesh network is proposed. Compared with conventional packet transmission which follows “one flow on one radio”, it uses the radio diversity to transmit the packets on different radios simultaneously. Three components are presented to achieve parallel-transmission, which are control module, selection module and schedule module. A localized selecting algorithm selects the right radios based on the quality of wireless links. Two kinds of distributed scheduling algorithms are implemented to transmit packets on the selected radios. Finally, a parallel-adaptive routing metric is presented. Simulation results by NS2 show that this parallel-transmission scheme could enhance the average throughput of network by more than 10%.

Keywords: Wireless Mesh Network, Radio Diversity, Parallel Transmission, Scheduling Algorithm

1. Introduction

Wireless local network (WLAN) is widely implemented today to provide hot spot coverage. Wireless Mesh Network (WMN) [1,2], a new wireless architecture is emerging as a latest trend in development because of its lower cost and wider coverage. A WMN is made up of Mesh Routers (MRs) and Mesh Clients (MCs). MRs with less mobility form the backbone of WMN, provide access to Internet for MCs. MCs might be mobile or stationary, and they can be linked to MRs directly or with the help of other clients. Some of MRs work as gateways. As a result, all the covered equipments can be linked to Internet in several hops.

Recently the tremendous popularity of wireless systems has led to the commoditization of RF transceivers (radios) whose prices have fallen dramatically. The use of two or more radio modules in a device is becoming economically feasible. On the other hand, MRs and MCs must content for single channel to forward packets in Single-Radio WMN. The channel contention leads to a low network capacity and non-predictable network delay. Due to the physical limitation it is very hard to improve the performance through protocol redesign. Therefore, to further improve the flexibility of WMN, a MR is usually equipped with multiple radios. BelAir [20] reports the capacity of 3 modes (single-radio, dual-radio and multi-radio) of

WMN in its white paper. It's shown that multi-radio mode has the best performance. Bahl [3] shows wireless systems using multiple radios in a collaborative manner dramatically improve performance and functionality over the traditional single radio wireless systems, that is popular today in terms of energy management, capacity enhancement, mobility management, channel failure recovery, and last-hop packet scheduling behavior. On the basis of this work, a great number of related researches have been done in terms of routing, medium access control, channel assignment etc [4-8,12].

Most works on MR-WMN are based on “one flow transmission on one radio”, that is to say the system will select a best radio to use when transmission occurs. [5] selects the best link quality corresponding to one radio. Some other researches focus on channel assignment which tries to make the network with lower interference and higher capacity [13,14]. As node only selects the best radio, mostly, some radios are free. Links between two nodes are always more than one in multi-radio environments, and they work on different channel, so packets could be parallel transmitted on them. It will enhance the throughput of transmission. To make all the radios collaboratively work on one node, utilizing the radios' diversity will be the main point of this paper. We provide a parallel-transmission method here, meanwhile, a parallel-adaptive routing metric is provided when mesh nodes are selecting routes. The contributions of this

paper are as follows:

- It's the first provision of adopting parallel transmission to advance the multi-radio utilization.
- A Parallel-transmission model proposed, which makes parallel-transmission operate as on one radio.
- The selecting algorithm operating on the model is localized and the selection is based on the quality of wireless links. This algorithm adapts to the asymmetry of wireless environments.
- Two distributed scheduling algorithms are presented for our model in different ways. Algorithm 1 transmits multiple, possibly erroneous copies of a given frame on selected-radios. Algorithm 2 transmits packets on different radios with random probability.
- A parallel-adaptive routing metric is introduced for routing selection.

Some simulation experiments are carried to show that parallel transmission could exactly enhance the performance of wireless mesh networks. Three key performance metrics are analyzed in this paper. Our two parallel algorithms both could improve the throughput by 10-50%, and the delay could exactly fit the transmission of some special applications such as VoIP and so on. Meanwhile, the retransmission probability reduces using scheduling algorithm 1.

The rest of the paper is organized as follows. In Section 2, we present a review of related work in this area. Motivations of this paper are provided in section 3. Section 4 describes our parallel transmission mechanism. And in section 5, an adaptive routing is proposed. Section 6 shows the simulation results. Finally, section 7 concludes this paper.

2. Related Work

2.1. Media Access Control for Multi-Radio System

Several researchers have submitted extra mechanisms to improve the performance of multi-radio wireless network using multi-radio diversity. For wireless local area network, the Multi-Radio Diversity (MRD) wireless system is presented in [9], which uses path diversity to improve loss resilience. It incorporates two techniques to recover from bit errors and reduce the loss rates observed by higher layers, without consuming much extra bandwidth. One is frame combining, and the other is request-for-acknowledgement (RFA).

In multi-hop wireless network, another multi-radio unification protocol (MUP) is introduced in [6]. MUP conceals multiple NICs from layers above it by presenting a single virtual interface, and then MUP periodically monitors the channel quality on each interface to each of its neighbors. When it is time to send a packet to a neighbor, it selects the right interface to forward the packet on. This means that at one time MUP only allows one interface to work and the other interfaces are out of work. It's a low utility ratio of multi-radio.

2.2. Routing in MR-WMN

As a simply implemented metric, hop count is widely

used in wired network routing protocols. However, minimal hop count couldn't be equal to good link quality because of the complexity of wireless link. Most researches try to achieve a good routing metric to character the quality of wireless link. [10] presents the expected number of transmission (ETX) which is measured as follows:

$$ETX = \frac{1}{d_f * d_r} \quad (1)$$

d_f (forward delivery ratio): probability that a data packet successfully arrives at the recipient.

d_r (reverse delivery ratio): probability that the ACK packet is successfully received.

Analysis done in [11] shows that ETX has the best performance compared to hop count, RTT and PktPair in static multi-hop wireless networks. Based on ETX, [5] provides a new routing metric called WCETT which could character the diversity of multi-channel. It is based on the idea of MUP. Now, most routing designing in based on the above work. On one hand, many routing metrics [5,10] provide better characterization of link state. On the other hand, routing protocols for multi-radio environments [5,6] use these metrics to find a transmission path. However, all of them have the same disadvantage, which is the low utilization of multi-radio.

To fully utilize the diversity of multi-radio, a parallel-transmission scheme is provided. This parallel-transmission is adaptive to multi-hop wireless mesh network.

3. Motivation

3.1. Parallel Transmission

Two interfaces working on different frequency can forward the packet simultaneously as it is regarded as no self-interference. To achieve more capacity, the utilization of diversity of multi-radio and multi-channel becomes a key point. Many channel assignment algorithms [14-16] try to find the optimization assignment using both centralized and distributed methods. Multi-channel and multi-radio comes in order to decrease the self-interference under the transmission range and enhance throughput of end-to-end transmission. However, the mere usage of diversity of multi-radio for preventing inter-flow or intra-flow interference is not enough. This paper proposes a new usage of the diversity of multi-radio, which is called parallel transmission. Figure 1 shows that the two transmitting nodes are equipped with dual-radio and they work on different channels with no interference. Thus the packets on link1 and link2 can transmit simultaneously just as P1 and P2 do. The existing researches only considered the parallel transmitting between links in which the transmitting node pairs are different.

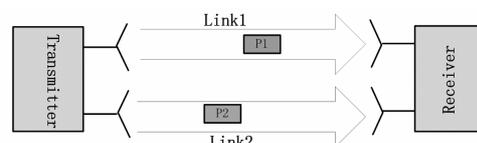


Figure 1. Parallel transmission between two nodes.

Utilizing radio diversity to improve network performance can be considered in both the MAC design and in routing selection. In MAC design, parallel transmission control strategy is a supposed scheme. Packets in the transmission queue are divided to transmit from different links between the transferring node pair according to the link performance. In routing path selecting, a node equipped with multi-radio could be seen as multi- hosts and which is called layer-2 routing. In this paper, the parallel strategy is considered in MAC design.

3.2. Parallel Transmission Adaptive Routing Metric

Mesh routing metrics such as ETX [10], ETT, and WCETT [5] are mostly link characterized. A unicast routing is to select a path formed by several links which have the best performance. Thus the selection is on the link instead of on the node. The attention should be focused on the parallelized links when parallel transmission occurs. Since we could get performance of each link, one unique metric should be integrated to character quality of the parallel transmission links. Both the cost and time decreasing caused by parallelization should be considered. A routing metric adaptive parallel transmission achieving by mathematics analyzing is presented latter in section 5.

4. Parallel Transmission System

4.1. Parallel Transmission System Model

Figure 2 shows the model of parallel transmission. The left part is the protocol stack. Parallel transmission mechanism is implemented at the link layer called parallel-adaptive MAC. It does the collaboration of multiple interfaces and exposes a *virtual MAC address* in place of the multiple physical MAC address. Thus, from the application perspective it operates as if there is only a single wireless network interface.

The parallel-adaptive MAC includes three components: control module (V-MAC), selection module, and schedule module. Selection module selects the right interfaces for each flow using a localized algorithm. Schedule module will transmit packets using these selected interfaces. V-MAC is designed to do the unification work such as exposes the *virtual MAC address* to the upper layer. Since the transmission mode is changed, transmission quality between two nodes has been motivated. A parallel-adaptive metric is proposed in routing layer.

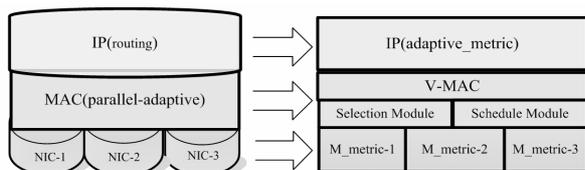


Figure 2. Parallel Transmission System Model.

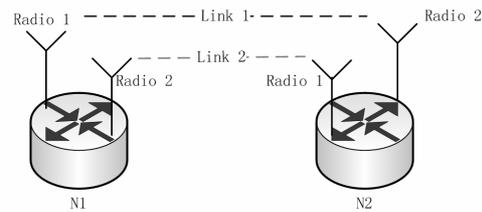


Figure 3. A transmission example.

For each node, a *virtual MAC address* is assigned, otherwise, each NIC has its real physical address. When node discovers a neighbor, not only the virtual address of this neighbor should be known, but also the real address of the interface. As seen in Figure 3, there's two links (Link 1 & Link 2) between node N1 and N2. We use the following format to character Link 1:

$$(Id_{N1}, Id_{Radio1})-(Id_{N2}, Id_{Radio2})$$

Id_{Ni} denotes the id of node N_i . *virtual MAC address* could be used here. $Id_{Radio(i)}$ figures the real physical address of which radio is linked.

In this paper, link metric is proposed to character the quality of link. Thus, each network interface will provide a link list which includes all links using this interface. Table 1 describes each element of this data structure. Meanwhile, we call it *Mac Metric (M_Metric)*.

As described above, (Id_{node}, Id_{NIC}) could character the information of both node and interface. *Link Metric* is used to character the quality of this link. There're many types of metrics have been researched, such as ETX, etc. ETT [5] is proposed to fit multi-radio system, thus it is used to characterize link metric in this paper.

$$ETT = \frac{ETX * S}{B} \quad (2)$$

S is the size of packet and B is the bandwidth of this link.

M_Metric of each NIC is input to Selection Module. When a link is selected, the *link Status* filed in the M_Metric of it will be updated to 1 (1 indicates selected and 0 unselected). After selecting completes, the V-MAC exposes a single *virtual MAC address* in place of the selected multiple physical MAC addresses used by the wireless Network Interface Cards (NICs).

4.2. Parallel Transmission Scheduling

As described above, Selection Module and Scheduling Module will determine which interfaces to choose for parallel transmission and how to schedule them.

1) Localized Selecting Algorithm

When a transmission occurs on node u , it must know which radios to transmit these packets firstly. A localized selecting algorithm is provided here based on the quality of wireless links. M_Metric of each link is INPUT of this algorithm. While selecting ends, it gives a structure ($SUM-MAC$) which is the OUTPUT of this algorithm. $SUM-MAC$ should indicate the information of selected radios.

Algorithm 1: Selecting Algorithm for Node u

INPUT: $L(M_Metric)$ link of M_Metric for each NIC
 OUTPUT: $SUM_MAC(u)$, the associated MAC structure of node u
 PROCEDURE:

```

1:  function  $SUM\_MAC(u) = \text{RadioSelect}(L(M\_Metric))$ 
2:    for each neighbor of node  $u$ 
3:       $Best\_NIC = \text{Select\_Best}(L(M\_Metric))$ 
         $\Rightarrow$  Find the NIC with best link metric
4:    for each NIC of node  $u$ 
5:      if (  $\text{compare}(M\_Metric(cur), M\_Metric(Best))$ 
         $< \varepsilon$  )
         $\Rightarrow$  Value difference of link metric between NIC
         $cur$  and NIC  $Best$  is less than  $\varepsilon$ 
6:         $Add(SUM\_MAC(u), M\_Metric(cur))$ 
7:         $Update(M\_Metric(cur))$ 
         $\Rightarrow$  Select the current NIC and update its
        status
8:      end if
9:    end for
10:    $MAC\ Combine(SUM\_MAC(u))$ 
11: end for
12: end function

```

The basic idea for this selecting algorithm can be sketched below. NIC with the best link-metric for each link is selected first. Then the algorithm will select certain NICs which have the similar link quality with the best. And ε is a common-sense value. The selected NICs are added to SUM_MAC and the field of *Link Status* is set to 1. Finally a combining work is done for each link and SUM_MAC of node u is outputted.

Note that Algorithm 1 is a localized one with each node u running a copy and making its decisions independently. As we know, links of wireless environments are asymmetric due to interference of neighbor nodes, background noise and so on. E.g. as shown in Figure 3, the quality of transmission from node N1 to node N2 with link 1 may be different from transmission from node N2 to node N1 with link1. Thus, the selecting results between transmission (N1->N2) and transmission (N2->N1) are different. It is suitable for asymmetry of wireless environments.

2) Two Scheduling Algorithm

After certain radios are selected, schedule module assigns packets transmitted to these radios for transmitting. INPUT of scheduling is SUM_MAC and OUTPUT is SCH_D which is packet queue for transmitting. Two kinds of scheduling algorithms are implemented here.

Algorithm 2.1: Copy-based Scheduling Algorithm for Node u

INPUT: $SUM_MAC(u)$ of Node u , Transmission m
 OUTPUT: $SCH_D(u)$ for Transmission m
 PROCEDURE:

```

1:  function  $SCH\_D(u) = \text{Copy\_Schedule}(SUM\_MAC(u), m)$ 
2:     $Initialize(SCH\_D(u))$ 
3:     $Node\ i = \text{GetNeighbor}(m)$ 
         $\Rightarrow$  Get the neighbor node by transmission
4:    for each member of  $SUM\_MAC(u)$  and Neighbor ID is  $i$ 
5:       $Update(SCH\_D(u), m)$ 
6:    end for
7: end function

```

A $SCH_D(u)$ for each transmission m will be gotten by this algorithm. Packets are copied and sent on each selected NIC in SUM_MAC . At the receiver, Virtual MAC handles data transmissions and retransmissions. Since the packets are copy-sent, loss resilience is improved here. For example, there're two links are selected between two nodes. We assume that loss rate of them are p_1, p_2 separately. Packets are lost only if link 1 and link 2 all missed. The probability is $p = p_1 * p_2$.

Multi-transmission with one packet will affect the performance of network when it is with heavy load. Another algorithm called partition-based scheduling is present as follows. It aims to improve throughput of network.

Algorithm 2.2: Partition-based Scheduling Algorithm for Node u

INPUT: $SUM_MAC(u)$ of Node u , Transmission m
 OUTPUT: $SCH_D(u)$ for Transmission m
 PROCEDURE:

```

1:  function  $SCH\_D(u)$  =
    Partition_Schedule( $SUM\_MAC(u), m$ )
2:     $Initialize(SCH\_D(u))$ 
3:     $Node\ i = \text{GetNeighbor}(m)$ 
4:    for each member  $i$  of  $SUM\_MAC$ 
5:       $Compute\_prob(i)$ 
6:    end for
7:     $vectorp = \text{BuildVector}(prob\ of\ each\ NIC\ of$ 
     $V\_MAC)$ 
         $\Rightarrow$  Get the neighbor node by transmission
8:    for each packet  $p$  of  $m$ 
9:       $hit = \text{random}(vectorp)$ 
10:      $update(SCH\_D)$ 
11:    end for
12: end function

```

The basic idea of Algorithm 2.2 is sketched below. Node u , the source, first gets the neighbor node which the packets will be transmitted to. For each packet, it is transmitted by the selected NICs with a random probability. This probability is denoted as follows. We assume that there're t links are selected and the metric of them are $ETT_1, ETT_2 \dots ETT_t$. The probability of each link is:

$$prob_i = \frac{ETT_k}{\sum_{k \neq i} ETT_k \prod_{1 \leq s \leq num(selected), j \neq i} ETT_j} \quad (3)$$

This probability is obtained by these rules:

$$\begin{cases} \sum_{\text{for all selected NIC } i} prob_i = 1 \\ \frac{prob_i}{prob_j} = \frac{ETT_j}{ETT_i} \end{cases} \quad (4)$$

It is possible that there is only one best link, in this case, it will be a one-link selected. For each packet i , it will be randomly transmitted on NIC k with a probability of $prob_k$, which is computed by (2). For example, if two radios are selected, and link metric of them (ETT) are 0.2s and 0.4s separately. Then, $prob_1 = ETT_2 / (ETT_1 +$

$ETT_2) = 2/3$, and $prob_2 = ETT_1/(ETT_1+ETT_2) = 1/3$.

Some other link metric could also be used here. However, it must be short-term and propagated localized. In Wireless Mesh Network, the locations of nodes are fixed, and therefore the set of potential neighbors (adjacencies) of a node that are within its transmission range is also static. On the other hand, the quality of a wireless channel between adjacent nodes varies frequently because of various factors such as external interference, channel fading, and inclement weather. So our parallel-transmission is based on a neighbor pair. In other words, it is just one hop, and our link metric is short-term based.

5. Parallel Adaptive Routing Metric

As stated in [11] the metric ‘‘hop count’’ cannot work well in static wireless networks. Meanwhile, wireless link quality characterization is a hard problem because of the variability of wireless network. As a result, forecasting more veracious information of the wireless network performance is kernel part of the routing design.

However, as the main work of this paper focuses on parallel-transmission, designing a new link metric for wireless mesh network is not considered here. This part only proposes a parallel-adaptive routing metric which is improved from ETT.

An assumption is made that *link 1* and *link 2* between the node pair A and B are selected to join the parallel transmission.

As in Parallel-1 Algorithm, packet p will be sent both on *link 1* and *link 2*. Packet will be come first when link has the best ETT. Thus, for Parallel-1 Algorithm, the metric of sending time is calculated as follows:

$$send(T) = \min_{i \in selected} (ETT_i) \quad (5)$$

While in the previous description of Parallel-2, the packet list is divided into two parts, which will transmit separately. Suppose the expected transmission time is ETT_1 on *link 1* and ETT_2 on *link 2*, and there are n packets, each of size S_0 , waiting to be sent. As described above, the number of packets transmitted on link i will be:

$$\begin{aligned} num(Transmit_i) &= n * prob_i \\ &= n * \frac{\sum_{k \neq i} ETT_k}{\sum_{1 \leq i \leq num(selected)} \prod_{j \neq i} ETT_j} \end{aligned} \quad (6)$$

Then the sending time will be:

$$send(T) = n * \frac{\sum_{1 \leq k \leq num(selected)} ETT_k}{\sum_{1 \leq i \leq num(selected)} \prod_{j \neq i} ETT_j} \quad (7)$$

Now, let us denote the constant cost of packet scheduling is T_0 . The routing metric Cost Time (CT) is defined as follows:

$$CT = T_0 + send(T) \quad (8)$$

Equation (8) gives a routing metric which is adapting to our scheduling mechanism for the two parallel scheduling. It works well for routing protocol.

6. Simulation Results

The distributed algorithms presented in Section 4 were implemented in *ns-2* [17]. The simulation for our protocol proposed in this paper uses the topology creator to randomly create the scenarios. Table 2 gives the default parameter setting used in the simulation study.

Each node is equipped with 5 radios. Two kinds of traffic source are set, one is CBR, and the other is FTP. The source-destination pairs are spread randomly over the network.

Evaluation of the serial algorithm and our 2 parallel scheduling algorithms are investigated. Meanwhile three key performance metrics are evaluated: (1) *Throughput* – the data packets delivered to the destination generated by the CBR and FTP sources; (2) *Delay*–the delay jitter above the minimum one-way packet delivery time; (3) *Retransmission probability*–number of retransmission packets with a certain number of sending packets.

6.1. Throughput

As described above, 10 pairs of UDP and TCP flows are run in the scenario randomly. We compare the performance of none-parallel scheme (MUP) against our two parallel schemes.

1) Impact on UDP Throughput

We now consider the 10 pairs UDP transmissions in the scenario. Each throughput of the 10 pairs is shown in Figure 4. As shown in Figure 4, both the two parallel algorithms improve the throughput in UDP flows. The average throughput over 10 UDP flows for the none-parallel experiments is 2.86Mbps. Meanwhile, our parallel-1 is 3.31Mbps and parallel-2 is 5.44Mbps. The two parallel algorithms constitute improvements of 10% and 90% separately.

Table 2. Basic settings for simulation experiment.

Parameter	value
T_0	5% of transmission time
ϵ	10%
Number of nodes	50
Filed	2000meters * 2000 meters
Transmission Range	250 meters
Carrier Sensing Range	500 meters
Application	CBR FTP
Packet Size	500 bytes
Path Metric	Hop Count WCETT SMTT

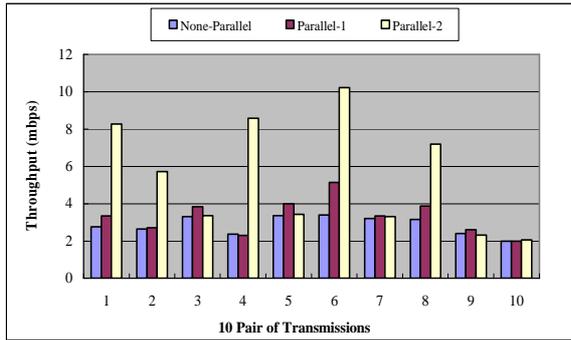


Figure 4. UDP throughput comparison.

Table 3. Average throughput for UDP transmissions.

Transmission	Average Throughput (Mbps)
None-parallel	2.86
Parallel-1	3.31
Parallel-2	5.45

2) Impact on TCP Throughput

We now consider the impact of parallel on TCP transmissions. 10 pairs of TCP transmissions are also tested in the simulation. The results of throughput are shown in Figure 5. Table 4 figures the average throughput of the three kinds of transmissions. None-parallel is 2.48 Mbps, Parallel-1 is 2.89 Mbps and Parallel-2 is 2.80 Mbps. The average throughput of both two parallel algorithms improves by more than 10% to the original scheduling. Compared to UDP flows, it has less impact on TCP flows. It is caused by that performance of TCP flows is influenced by RTT. When the node doesn't choose the best link to transmit packets, it will cause more delay. It increases the value of RTT. Thus, TCP performance decreases. However, the whole capacity enhances.

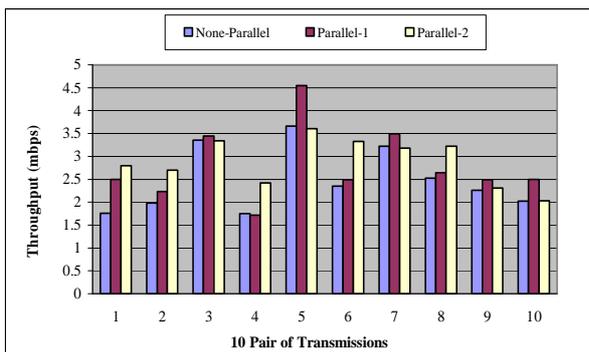


Figure 5. TCP throughput comparison.

Table 4. Average throughput for TCP transmissions.

Transmission	Average Throughput (Mbps)
None-parallel	2.48
Parallel-1	2.89
Parallel-2	2.80

6.2. Delay Analysis

Many applications such as VoIP and video streaming require a relatively low packet delivery delay not exceeding 100-150 ms [18,19]. Here, we investigate all the flows described above. In other words, both the TCP transmissions and UDP transmissions are calculated here. To simply analysis, retransmissions are neglected, and only the one-way packet delivery delay is captured.

Figure 6 shows the one way delay distribution of our simulations. Compared to none-parallel scheduling, parallel-1 delivers packets with a lot smaller delay because it is able to recover most corrupt frame retransmission to the neighbor, and meanwhile, it could improve the bandwidth of the network. Parallel-2 could also improve the throughput of the network. Since parallel-2 uses some devices which have more packet loss probability, it may impact the delay. However, it still has 5% more packets delivered than none-parallel scheme below 1ms. However, about 25% of the packets in the two parallel scheduling mechanisms are delivered with a significantly higher delay than none-parallel. Nonetheless, our parallel scheduling is able to deliver 95% of the packets within a delay of 45 ms, which is well below the delay bound of 150 ms that can be tolerated by telephony and video applications.

6.3. Retransmission Probability Analysis

To characterize the loss probability of each link by higher layer, we summarize the number of retransmission packets. The comparison of 3 scheduling is shown in Figure 7. This is based that we have limited the number of packets sending. We set the maxpkts_ to 1500. That means the application only procedure 1500 packets. And the sum of retransmission packets for all flows is calculated. We could see that the none-parallel and parallel-2 schedule almost have the same number of retransmission packets, because parallel-2 only choose these NICs which have the similar loss probability as the best one. However, as multi-send of packets by parallel-1, it has the less retransmission numbers than others.

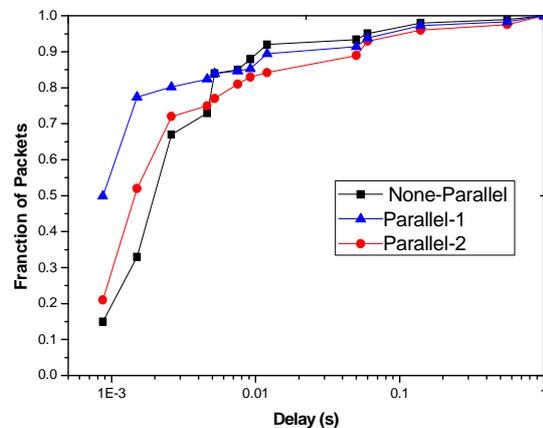


Figure 6. Delivery delay comparison.

7. Conclusions

In this paper we present a new usage of multi-radio diversity in wireless mesh network. A new parallel-transmission model is proposed which makes packets transmit on different radios simultaneously. Based on this model, a radio selecting algorithm and two distributed scheduling algorithms are presented. We also provide an adaptive routing metric based on our parallel-transmission, which can fully take advantage of the diversity of multi-radio.

Simulation results show that our new scheme could enhance the performance of network both in throughput and delivery delay. And also the whole retransmission probability decreases.

The main work of this paper is to provide a new transmission scheme. There're still many works to be done in the future. For example, some scheduling optimization based on multi-hop could be further investigated, which will require the analysis of behavior of flows.

8. Acknowledgements

The research is supported by USTC Innovation Foundation for Graduate Students 2007 under Grant No. KD2007048 and Natural Science Foundation of Anhui under Grant No. KJ2007A053.

We are grateful to Dr. Vivy Suhendra, who has made a lot of help in this paper. We are also grateful to anonymous reviews for their invaluable and constructive comments and suggestions.

9. References

- [1] R. Bruno, M. Conti, and E. Gregori, "Mesh networks: Commodity multihop ad hoc networks", *IEEE Communications Magazine*, Vol. 43, No. 3, pp. 123–131, March 2005.
- [2] I. F. Akyildiz, X. D. Wang, and W. L. Wang, "Wireless mesh networks: A survey," *Computer Networks and ISDN Systems*, Vol. 47, No. 4, pp. 445–487, March 15, 2005.
- [3] P. Bahl, A. Adya, J. Padhye, and A. Wolman, "Re-considering wireless systems with multiple radios," *ACM SIGCOMM Computer Communication Review*, Vol. 34, pp. 39–46, October 2004.
- [4] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. D. Zhou, "A multi-radio unification protocol for IEEE 802.11 wireless networks," *Proceedings of the First International Conference on Broadband Networks (BROADNETS'04)*, pp.344–354, October 25–29, 2004.
- [5] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom'04)*, pp. 114–128.
- [6] A. A. Pirzada, M. Portmann, and J. Indulsk, "Evaluation of multi-radio extensions to AODV for wireless mesh networks," *Proceedings of the International Workshop on Mobility Management and Wireless Access (MOBIWAC 2006)*, pp. 45–51.
- [7] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multi-radio multi-channel wireless mesh networks," *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking (Mobicom'05)*, pp.73–87, September 2005.
- [8] M. Alicherry, R. Bhatia, and L. Li, "Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks," *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking (Mobicom'05)*, pp. 58–72, September 2005.
- [9] A. Miu, H. Balakrishnan, and C. E. Koksal, "Improving loss resilience with multi-radio diversity in wireless network," *Proceedings of the 11th Annual International Conference on Mobile and Networking (Mobicom'05)*, September 2005.
- [10] S. Douglas, J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless network," *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (Mobicom'03)*, pp. 134–146, San Diego, California, September 2003.
- [11] R. Draves, J. Padhye, and B. Zill, "Comparison of routing metrics for static multi-hop wireless networks", *Proceedings of the Sigcomm'04*, 2004.
- [12] Y. H. Kim and J. B. Sunk, "Performance improvement for best-effort traffic of IEEE 802.11e QoS MAC in mobile ad hoc networks," *Proceedings of WiCOM 2006*.
- [13] P. Kyasanur and N. H. Vaidya, "Routing and interface assignment in multi-channel multi-interface wireless networks," *Proceedings of Wireless Communications and Networking Conference 2005*, Vol. 4, pp. 2051–2056, March 2005.
- [14] A. P. Subramanian, H. Gupta, and S. R. Das, "Minimum interface channel assignment in multi-radio wireless mesh networks," *Proceedings of the Fourth Annual IEEE Communications Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks (SECON 2007)*, June 2007.
- [15] Trops Network: Assignment of channels to links of nodes within a mesh network.
- [16] B. J. Ko, V. Misra, J. Padhye, and D. Rubenstein, "Distributed channel assignment in multi-radio 802.11 mesh networks," *Proceedings of IEEE WCNC 2007*, Hong Kong, China, March 2007.
- [17] NS2, <http://www.isi.edu/nsnam/ns/>.
- [18] A. Kopsel and A. Woliz, "Voice transmission in an IEEE 802.11 WLAN based access network," in *Proceeding of ACM WoWMoM (Rome, Italy)*, pp. 23–32, July 2001.
- [19] S. W. Lu, Thyagarajan, and V. Bharghavan, "Design and analysis of an algorithm for fair service in error-prone wireless channels," *Wireless Network*, Vol. 6, pp. 323–343, 2000.
- [20] Bel Air Networks, <http://www.belairnetworks.com/>.

Generality Challenges and Approaches in WSNs

Fadi TIRKAWI, Stefan FISCHER

Institute of Telematics, University of Luebeck, Luebeck, Germany

Email: {tirkawi, fischer}@itm.uni-luebeck.de

Received September 4, 2008; revised December 1, 2008; accepted December 31, 2008

Abstract

Ignoring the generality in the design of Wireless Sensor Networks (WSNs) applications limits their benefits. Furthermore, the possibilities of future extension and adaptation are also restricted. In this paper, several methods to enhance the generality in WSNs are explained. We have further evaluated the suitability of these methods in centralized and de-centralized management scenarios.

Keywords: Component, Middleware, Standardization, Generality, Wireless Sensor Network, Management, COTS

1. Introduction

The field of wireless sensor networks (WSN) is becoming a very popular area that starts having many applications in different fields. In addition to the benefits which one can get, this expansion also comes with many difficulties. In this paper, we target one of them, namely to provide standard and general paradigms in dealing with WSNs. Supporting platform- and hardware- independent applications for WSN is very important in order to ease the use of the big diversity of WSN applications. Moreover, in WSNs, finding generic management architectures that can be reused for managing different sensor node platforms is a challenge being posed and emphasized by the WSN community for long time. The management architecture should be capable of running over a broad range of WSN platforms and supporting a wide variety of WSN applications. The purpose of this paper is to present generality solution which can provide this feature.

WSN Management provides control and management of the system components such as the network layer, operating system parameters and application settings in real sensor nodes. Such tools that can provide generic remote interactions of the nodes are really missing.

Initially, this generality issue should be viewed in the context of the current traditional network management challenges, so that the learned lessons from generality of the traditional networks should be exploited. Therefore, we are going to discuss some similarities and distinctions of generality between the traditional and wireless sensor networks.

In traditional networks, the generality of the system

provides several important benefits such as compatibility, interchangeability, and simplicity. The compatibility enables the users to take the advantage of different components in similar architectures. The interchangeability among components with different architectures is enabled. The simplicity enables having a simple and similar way of interacting with the components and using them. Generality solutions provide other advantages such as resolving the heterogeneity and providing openness. Although generality provides many advantages, it adds additional overhead to the system. However, this overhead should not cause a significant loss of the efficiency.

WSNs are applications-specific, which is, obviously, in contrary to generality. Here, it is very difficult to find general solutions due to the following reasons. The nodes in WSNs have limited resources such as small physical size, small memory, weak computation, limited energy budget, and narrow bandwidth. Moreover, there is wide range of applications in which one can apply WSNs such as environmental tracking applications, medical and industrial applications, home automation applications, surveillance systems, etc. Furthermore, there are diverse hardware equipments used in designing sensor nodes. For example, there are many sensor technologies which can be built in sensor nodes such as: ECG, EMG, humidity, temperature, vibration sensors. This diversity requires a specific design and specific settings. Finally, over the last few years of WSNs involvement, a large number of software solutions have been introduced which have increased the software diversity and hence have complicated further finding general solutions.

This paper is organized as follows. Section 2

3. Generality Paradigms

To achieve generic solutions for the heterogeneous matrices of WSNs, we propose five schemes. These are based on a survey of existing solutions and on schemes we have proposed. These solutions can also be integrated with the management of WSNs to provide a generic management.

3.1. Middleware

The middleware aims at providing transparent common level abstractions of one or different levels of the local or remote nodes. In other words, middleware techniques are used to resolve two main challenges. The first is interfacing homogenous applications on different platforms, and the second is interfacing two heterogeneous applications running on homogenous platforms. In WSNs, Middleware can be used to reduce and address the limitations of the application specificity. It also supports the commonness of the systems, deployment, development and maintenance. Many middleware solutions for WSNs have been proposed. Salem, *et al.* [3] have covered in their classification large number of the WSN middleware proposals such as: Mate' [4], TinyDB [5], SINA [6], and many others.

In Mate', a middle abstraction layer is provided. This layer interprets the applications as byte code. Mate' has 24 one-byte-long instructions which can be injected into the network and then propagated to the nodes. Such an abstraction hides the original platform (hardware and operating system) to unify the interpretation of the Mate' instructions; hence, it provides the portability of the applications, which are built using Mate' instruction set, among different platforms that support Mate' virtual machine.

As it is not feasible to accomplish common middleware for all kinds of WSNs' nodes, WSNs should be classified into subcategories. As a proposal, WSNs applications can be divided into high rate WSNs (industrial application, medical applications, home surveillance,

etc.) and low rate WSNs (habitat monitoring, agriculture monitoring, etc.). For each of these main categories, common middleware specifications can be given.

In order to provide a general solution by using middleware, the sensor node designer should adopt one of the middleware techniques, so that these particular sensor node types support the generality of all nodes using the same middleware.

The middleware approach can be applicable for both centralized and de-centralized management solutions. In centralized management, where nodes are globally managed by one or multiple external entities such as special strong central nodes or a cluster head, the management framework on these strong nodes should follow the middleware specifications followed on the individual nodes. This can easily be achieved, as these central nodes have unlimited resources as compared to ordinary sensor nodes. Therefore, they can accommodate multiple middleware of diverse existing sensor nodes. Hence, such strong nodes can provide support for different heterogeneous sensor nodes at the same time. In decentralized management, where nodes are locally establishing the management among each other, vendors should adopt the middleware which is used by other existing individual nodes, so that these different nodes have a common interface. Here, a general management is difficult to achieve due to nodes' restrictions. Figure 2 shows a representation of the system layers for both centralized and de-centralized management.

3.2. Dynamic and Mobile Agents

Mobile agents, in this context, are small pieces of code which can be exchanged between the nodes in order to resolve the heterogeneity. These mobile agents can also be added while the compilation time as the management entities in [7]. Here, the sensor nodes' manufacturer provides an agent which comprises the node specifications. These specifications can be later used to enhance compatibility with the other nodes specifications.

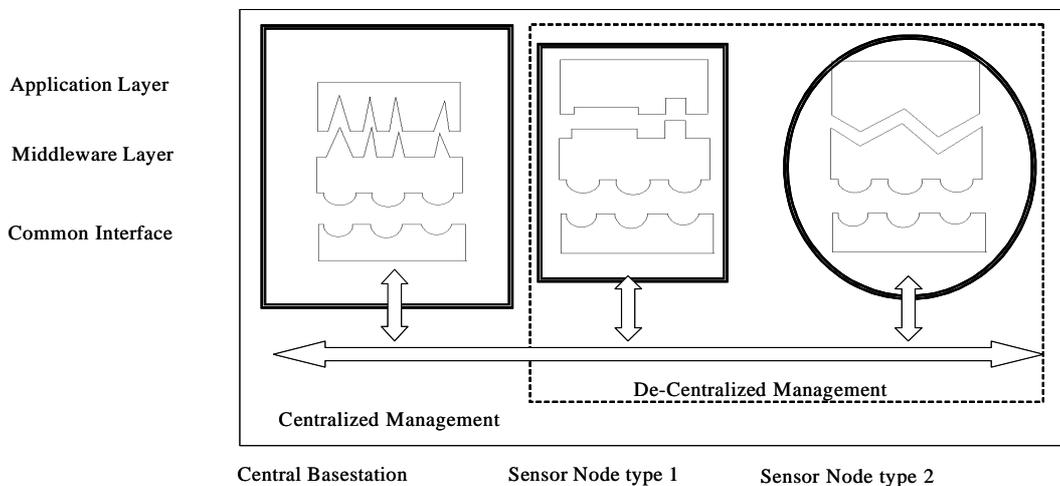


Figure 2. Representation of a middleware to have a general interface between nodes from multiple vendors.

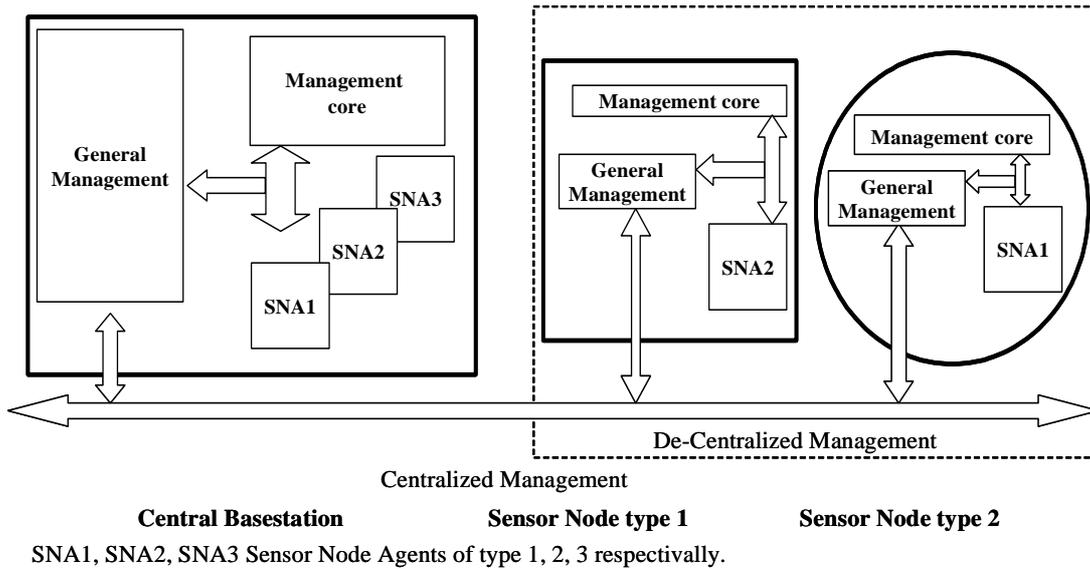


Figure 3. Representation of the mobile agent paradigm to enhance the generality of WSNs applications.

In Agilla [8], the users are able to inject the mobile agents, which are special code segments, into the nodes. These agents propagate into the nodes to perform the application-specific tasks. This fluidity of these agents has the potential to convert the nodes in WSN into a shared, general-purpose computing platform. Such platforms are capable of running several autonomous applications in parallel.

In decentralized management, the sensor nodes initially announce their specifications by exchanging their agents. Then, the received agents are configured with the management core in order to establish a general management compatible with other heterogeneous nodes. This method is very limited due to the restricted resources on the sensor nodes. Also, agents should be sent as binary code due to the complexity correlated with sending agents as sources.

In centralized management, this method is more efficient due to the unlimited resources available on central nodes as compared to the sensor nodes. As shown in Figure 3, the management core on the base station can launch the agents as dynamic libraries to resolve the specificity

of other nodes.

3.3. Semantic Methods

This method is based on agreement of the functional meanings of data fields and functionality of components. Here, the exchanged messages among heterogeneous application are semantically interpreted in order to produce a general messaging structure. This general messages format can be then used to provide general management. Figure 4 represents an example of parsing semantically two different messages from heterogeneous platforms (TinyOS and Contiki), in order to have common message structure.

Due to the complexity of applying formal language concepts to generate semantically common messages from different messages, this method can be mainly set on base stations in centralized management. The complexity and the effectiveness of this method depend on the degree of overlapping of the functionality and data structures among the heterogeneous WSNs.

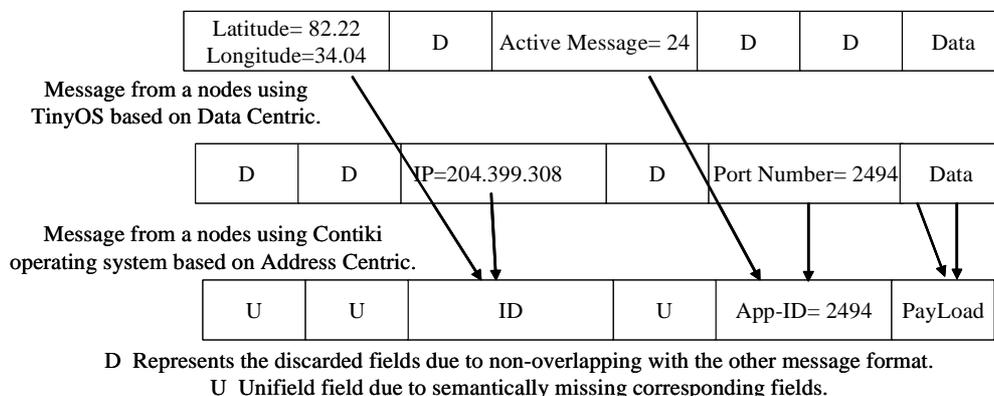


Figure 4. How to produce semantically a general packets used in management heterogeneous WSNs.

This method is followed in a tool called Message Interface Generator (MIG) [9] in TinyOS applications. It is specific for TinyOS applications. However, it provides generality among heterogeneous application using TinyOS. In this method, the user assigns semantic names to the messages' fields according to well-known naming conventions. On a base station, compatible messages can be semantically generated from the different messages format. Then, these generated messages can be used by the management core on the base station by the management framework which can be based on JAVA, C or any other programming language.

3.4. Standards

Standardization is a way to have compatibility, interchangeability or commonality among multiple systems based on different technical and operational fields. Using standards is another way to have generality among different systems using a particular standard.

In traditional networks, there have been many management standards (technical agreement) defined by different standardization organizations such as International Organization for Standardization (ISO), International Telecommunication Union (ITU), International Electromechanical Commission (IEC), Internet Engineering Task Force (IETF), etc. Out of these groups we find many standards that are proposed for networks management solutions. Simple Network Management Protocol (SNMP) is defined by IETF. Telecommunication Management Network (TMN), which uses Common Management Information Protocol (CMIP), is defined by ITU-T based on the management's specifications of OSI/ISO.

Normally, in all standards, one or more of the system abstraction matrices are fixed. These fixed matrices have to be followed by all vendors or manufacturers. They should fulfill and cover all needed functionality at a specific level of the system. Example for that is the SNMP. SNMP, which is proposed by IETF, has specifications which should be fulfilled by all vendors who are going to introduce a solution compatible with or general to this protocol.

In the WSN field, there are standards that have been adopted such as ISO-18000-7 [10], 6lowpan [11], WirelessHART [12], ZigBee [13] and Wibree [14]. All these standards are not developed explicitly for WSNs; rather, they are mainly proposed for supporting general low power and low rate networks, which is one category of current WSNs.

SP [2] (sensor net protocol) is, so far, the only standard-alike that has been proposed specifically for WSNs. SP is a significant step forward towards generalization of wireless sensor networks. SP represents a unifying abstraction layer that bridges the different network and application protocols to the different underlying data link and physical layers. SP is not at the network layer, which is the IP layer in OSI model,

instead it sits between the network and data-link layer (because in sensor nodes data-processing normally occurs at each hop, not just at the end points). SP can be used to identify an individual node, a set of nodes, or a communication structure such as a tree. SP provides generality because it enables the users to use any MAC layer without having to care about the overlaying network layers. Also, it enables users to use any network layer without having to care about the MAC specifications.

In WSN, using standards is the most efficient paradigm to obtain generality due to its simplicity and its efficiency. Standards cause lesser overhead in both centralized and decentralized management as compared to the other techniques that provide generality.

3.5. COTS (Commercial Off The Shelf)

COTS paradigm means using the components, hardware and software, available in the market in the design. Use of well-known legacy components based on existing standards enhance the generality, while introducing new components increases the specificity of the system. For instance, using the common AVR microcontrollers would ease dealing with sensor nodes rather than designing new application-specific microcontrollers for WSNs, since users have to design more specific development tools and conventions in the latter case. This way is still one of the dominating ways in designing general WSNs applications.

4. Conclusions

To sum up, generality is an important feature used to address the specificity of heterogeneous platforms. As WSNs are application-specific, solutions that support the generality provide a great help in management, deployment and development. Many methods can be applied to support the generality such as using middleware, semantic interpretation, mobile agents and using agreed standards. The selection of one of these methods is based on the degree of application specificity and on the management type whether it is centralized or de-centralized.

	<i>Centralized Management</i>	<i>Decentralized Management</i>
<i>Middleware</i>	Not scalable, easy to use, resources-efficient	Scalable, difficult to use, resources-inefficient
<i>Dynamic and Mobile Agent</i>	Not scalable, easy to use, resources-efficient	Scalable, difficult to use, resources-inefficient
<i>Semantic Methods</i>	Not scalable, easy to use, resources-efficient	Scalable, difficult to use, resources-inefficient
<i>Standards</i>	Not scalable, easy to use, resources-efficient	Scalable, ease to use, partially resources-efficient

In general, centralized management schemes are not suitable in large scale WSNs; however, decentralized schemes perform well in such scenarios, as the management is distributed over all the nodes. The implementation and the use of different generalization schemes in centralized management are simpler than in the decentralized management. Since the management in centralized schemes is performed at the central nodes having sufficient resources, the resources of the individual sensor nodes are less consumed. The table below summarizes the comparison of the generalization methods in both centralized and decentralized management.

5. References

- [1] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in the Proceedings of the 6th Annual ACM/IEEE MobiCom00.
- [2] J. Polastre, J. Hui, P. Levis, J. Zhao, D. Culler, S. Shenker, and I. Stoica, "A unifying link abstraction for wireless sensor networks," in SenSys 2005.
- [3] S. Hadim and N. Mohamed, "Middleware: Middleware challenges and approaches for wireless sensor networks," 2006 Published by the IEEE Computer Society, Vol. 7, No. 3, March 2006.
- [4] P. Levis and D. Culler, "Mat'e: A tiny virtual machine for sensor networks," in the Proceedings of Architectural Support for Programming Languages and Operating Systems ASPLOS 2008.
- [5] S. R. Madden, M. J. Franklin, and J. M. Hellerstein, "TinyDB: An acquisitional query processing system for sensor networks," ACM Transactions on Database Systems, Vol. 30, 2005.
- [6] C. Srisathapornphat, C. Jaikaeo, and C. Shen, "Sensor information networking architecture," in the Proceedings of ICPPW 2000.
- [7] F. Tirkawi and S. Fischer, "Remote interaction tool for wireless sensor networks," 3rd IEEE International Symposium on Wireless Pervasive Computing, May 2008.
- [8] D. C. Fok, G. Roman, and C. Lu, "Mobile agent middleware for sensor networks: An application case study," in the Proceedings of IPSN 05.
- [9] <http://www.tinyos.net/tinyos-1.x/doc/nesc/mig.html>.
- [10] Standards under development, <http://www.iso.org>.
- [11] "6lowpan," IPv6 over Low Power WPAN, <http://www.ietf.org/>.
- [12] "WirelessHART," <http://www.hartcomm2.org>.
- [13] "Zigbee," <http://www.zigbee.org>.
- [14] "Wibree," <http://www.wibree.com>.

Five Basic Types of Insider DoS Attacks of Code Dissemination in Wireless Sensor Networks

Yu ZHANG^{1,3}, Xing She ZHOU¹, Yi Ming JI², Yee Wei LAW³, Marimuthu PALANISWAMI³

¹*School of Computer Science, Northwestern Polytechnical University, Xi'an, Shaanxi, China*

²*College of Engineering and Computer Science, Australian National University, Canberra, ACT, Australia*

³*Department of Electrical and Electronic Engineering, the University of Melbourne, Parkville, VIC, Australia*

Email: ¹{zhangyu, zhous}@nwpu.edu.cn, ²yiming@rsise.anu.edu.au, ³{y.law, swami}@ee.unimelb.edu.au

Received October 27, 2008; revised December 8, 2008; accepted December 28, 2008

Abstract

Code dissemination is one of the important services of wireless sensor networks (WSNs). Securing the process of code dissemination is essential in some certain WSNs applications, state-of-the-art secure code dissemination protocols for WSNs aim for the efficient source authentication and integrity verification of code image, however, due to the resource constrains of WSNs and the epidemic behavior of the code dissemination system, existing secure code dissemination protocols are vulnerable to Denial of Service (DoS) attacks when sensor nodes can be compromised (insider DoS attacks). In this paper, we identify five different basic types of DoS attacks exploiting the epidemic propagation strategies used by Deluge. They are (1) Higher-version Advertisement attack, (2) False Request attack, (3) Larger-numbered Page attack, (4) Lower-version Adv attack, and (5) Same-version Adv attack. Simulation shows these susceptibilities caused by above insider DoS attacks. Some simple models are also proposed which promote understanding the problem of insider DoS attacks and attempt to quantify the severity of these attacks in the course of code dissemination in WSNs.

Keywords: Sensor Networks, Code Dissemination, Deluge, Security, DoS Attacks

1. Introduction

Wireless sensor networks (WSNs) now can provide many services with a large number of resource-constrained nodes. One important service is *code dissemination* which can disseminate new code images into all sensor nodes that need them over the wireless link. In order to guarantee flexibility, efficiency and reliability of code propagation, a number of code dissemination protocols (MOAP [1], Deluge [2], MNP [3] and Infuse [4], Sprinkler [5], Aqueduct [6], and Freshet [7], etc.) have been developed. However, none of them consider the communication security of WSNs.

Recently, some research works (Sluice [8], SecureDeluge [9] and Deng-tree [10]) have attempted to provide efficient authentication of code dissemination. These approaches, unfortunately, are vulnerable to Denial of Service (DoS) attacks because they do not take the authentication of the control packets (in Deluge, they are named as Advertisement (Adv) and Request (Req)) into consideration.

The contribution of this paper is that we identify five different basic kinds of DoS attacks made by malicious

nodes exploiting control packets. First is Higher-version Adv attack, second one is False Req attack, the third is Larger-numbered Page attack, the fourth is Lower-version Adv attack and finally is Same-version Adv attack. We also present the degree of damage made by each attack through quantitative analysis.

The paper is organized as follows. Section 2 reviews the related work on security and DoS attacks in WSNs. Section 3 gives an overview of Deluge and describes some vulnerability of the epidemic propagation strategies. Section 4 introduces five basic types of insider DoS attacks against Deluge and proposes the system models. Section 5 evaluates the performance of Deluge under different forms of the DoS attacks and discusses the simulation results. Section 6 concludes the paper.

2. Related Work

A variety of protocols have been proposed to support code dissemination in wireless sensor networks. MOAP [1] developed by Stathopoulos *et al.* extended XNP [11] and employed a publish/subscribe scheme to propagate

software update over a multi-hop network. Deluge [2], which is distributed with TinyOS [12], shared many ideas with MOAP, including the use of unicast NACKs and broadcast of the code. It spreads the code using spatial multiplexing. MNP [3], which was implemented in the Michigan State University, introduced a sender-selection algorithm which limits the total number of senders in one neighborhood to mitigate the hidden terminal effect.

Recent researches have developed some protocols to provide secure reprogramming services by extending Deluge with authentication and integrity mechanisms. Sluice [8], SecureDeluge [9] and Deng-tree [10] leveraged the similar solutions which based on digital signature and cryptographic hash function to guarantee the security of code images. they are distinguished through structure, granularity and strength of hashing [13].

Furthermore, some other protocols which are focused on security of communication in wireless sensor networks have been proposed. They can be classified into two types: *asymmetric* and *symmetric* mechanism. μ TELSA [14] is the representative of the former one. It provided broadcast authentication via symmetric primitives only, and introduced asymmetry with delayed key disclosure and one-way function key chains. The typical protocols of symmetric mechanism are q-composite key pre-distribution and random pairwise key schemes [15] proposed by Chan *et al.* They used pairwise keys to establish a secure communication infrastructure of wireless sensor networks and attempt to mitigate the threat of compromised nodes. Moreover, routing security is another important issue which is needed to pay attention to. Karlof and Wagner analyzed the security of all the major sensor network routing protocols and described crippling attacks against all of them and presented countermeasures [16].

However, though many secure protocols have been proposed, few of them could mitigate DoS attacks in sensor networks. Strictly speaking, although we usually use the term to refer to an adversary's attempt to disrupt, subvert, or destroy a network, a denial of service attack is any event that diminishes or eliminates a network's capacity to perform its expected function [17,22]. One typical DoS attack is that a captured node broadcast malicious messages to other nodes, resulting in a large amount of extra transmission, storage or computation overhead. Because wireless sensor networks are more resource-constrained compared with traditional networks, they are much easier to be destroyed by DoS attacks.

3. Problem of the Epidemic Propagation of Deluge

In this section we will first give a detail description of Deluge and then we point out some security vulnerabilities

Supported by the National Natural Science Foundation of China under Grant No.60573161, the Australian Research Council Research Network on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), and the DEST International Science and Linkage Grant.

of the epidemic propagation strategies in Deluge.

3.1. Overview of Deluge

Deluge is an epidemic protocol used for code dissemination which can guarantee large data objects to be disseminated quickly and reliably over a multi-hop wireless sensor network. It employs advertise-request-code handshaking protocol [18] to set up a reliable bi-directional link before transferring data and reduce the transmission of redundant data throughout the network. In Deluge, the binary image is divided into fixed-size pages, each page can be transferred within 48 same-size packets. This data representation supports software update based on page differences and makes use of spatial multiplexing to allow parallel transfers of code image. Deluge also borrows some ideas from Trickle [19] which uses suppression mechanism and dynamic adjustment of advertisement rate to achieve *density-aware capability* and energy efficiency.

Trickle uses a "polite gossip policy", where sensor nodes periodically broadcast a code summary to local neighbors but stay quiet if they have recently heard a summary identical to theirs. It divides time into a series of rounds and in each round nodes can decide whether or not to broadcast its own Adv. Deluge uses $\tau_{m,i}$ to denote the duration of round i . And $\tau_{m,i}$ is bounded by τ_l and τ_h . In each round, a node could broadcast its Adv at r_i which is picked up randomly in the range $[\tau_{m,i}/2, \tau_{m,i}]$. An Adv has a code summary φ which contains two integers $\{v, \gamma\}$, where v is the version number and γ is the largest numbered page available for transfer. If there is a new code image injected into WSN by base station, a node S, which has received the image, broadcasts an Adv with a summary $\varphi\{v_s, \gamma_s\}$. When a node R hears this Adv, it first compares its own image version v_R with v_s . If $v_s > v_R$, node R will update its version to v_s . Then if $\gamma_s > \gamma_R$, node R will send a Req to node S to request smallest numbered page it needed. S broadcasts the requested page after it received the Req. Through this process new images are propagated to all nodes page by page.

Density-aware capability of Deluge makes effect on Adv and Req propagation, where redundant advertisement and request messages are suppressed to minimize contention. In each round a node will not broadcast its own Adv unless the number of Adv with same version it heard is less than a predefined threshold k which is adjusted according to the density of the network. Moreover, if a node has heard Req packets for the page it needs before transmitting Req, it will stop requesting its own Req. Similarly, if a node hears request for the pages with smaller than that of the page it is currently transmitting, the node suppresses transmitting of the subsequent code packets.

If the network is consistent, Deluge will decrease its advertisement rate, i.e. it set $\tau_{m,i}$ to $2\tau_{m,i}$ in each round, but will be not larger than τ_h . Otherwise, it set $\tau_{m,i}$ to the minimal value τ_l . Deluge changes dynamically the rate of

advertisements to allow quick dissemination when needed and save resources when new code updates propagation is not needed.

3.2. Security Vulnerabilities of the Epidemic Propagation Strategies in Deluge

Epidemic propagation strategies allow rapid dissemination of information through purely local interactions in large scale, dynamic and not all the time coherent environments. In an epidemic protocol such as Deluge [2], a new code initiated from a source is rebroadcasted by neighboring nodes and extends outward, hop by hop, until the entire network is reached. The epidemic behavior provides high resilience to random process and network failures in the free attack scenario.

However, the power, communication, computation and storage capabilities of each sensor node are extremely limited and wireless sensor networks could be deployed in hostile and unattended environments for long periods of time. Each sensor node is insecure, which means it is trivially easy to retrieve program code, static data, and even dynamic program memory from nodes [20]. Moreover, the most of current code dissemination protocol like Deluge have not been designed with security in mind. Consequently, code dissemination may be possible to face threats from compromised nodes. If a node is captured, the attacker can gain control of that node and even gain complete control of an entire deployed network due to the epidemic nature of protocols like Deluge. For example, an adversary may use suppression and dynamic adjustments of the broadcast rate mechanisms of Deluge to prevent the propagation of code updates, waste network resources, introduce unnecessary latency or disrupt the normal operation of code dissemination.

Although many secure code dissemination protocols have been proposed recently, most of them aim for providing authentication and integrity of code updates in sensor networks to ensure malicious code are not disseminated or installed. However, none of them provide the effect schemes to prevent insider attackers from exploiting epidemic and suppression mechanisms of Deluge to launch DoS attacks. Although our future goal is to provide Insider-DoS-Resistant code dissemination scheme, we first should discover and analyze the susceptibilities of Deluge caused by insider DoS attacks.

4. Five Basic Types of Insider DoS Attacks and System Models

In this section we first describe five different basic types of insider DoS attacks against efficient code dissemination mechanisms used by Deluge. Then we proposed simple models for these attacks made by malicious nodes using control packets.

4.1. Insider DoS Attacks on Deluge

Attack 1: Higher-version Adv

This is an insider attack which aims to increase energy consumption and prevent normal nodes from receiving new code images. We assume the current version of image in it is v , at this moment a malicious node broadcast an Adv with higher version v' ($v' > v$), the network will be inconsistent and all the neighbor nodes will update their version to v' . Meanwhile, they will adjust $\tau_{m,i}$ to the minimal value τ_l according to the Deluge's rules. Therefore the adversary could exploit the dynamic adjustments of the advertisement rate mechanisms to enforce legitimate nodes to transmit more frequently and eventually result in energy waste of network. Furthermore, if at the same time the base station has injected a new image with version v'' lower than v' , the nodes, whose image version have updated to v' , could not get the new image with version v'' . In this case, the code dissemination will be failed.

Attack 2: False Request

This is another insider attack which targets to introduce unnecessary communication overhead or disrupt the normal code dissemination. According to the Deluge's rules, if a node S sends a Req to node R to request a specific page, R will broadcast a large number of code packets to S as response. Besides this case, if the page requested has a smaller page number than other neighbor nodes', Deluge will give this Req higher priority. Therefore the adversary could leverage these weaknesses of the rules to send bogus Req to trigger unnecessary transmission of code packets or send Req modified which included a smaller page number in order to suppress other Req from normal nodes.

Attack 3: Larger-numbered Page

This is an insider attack which is to target sensor node's energy consumption and looks similar to Attack1 mentioned above. We assume a node R has successfully received all packets in page0 to page γ' . According to the Deluge's rules, when this node R hears an Adv sent by node S which contains higher page γ ($\gamma > \gamma'$), R will send a Req packet to S to request new page $\gamma'+1$. This Larger-numbered Page of Adv from S could cause inconsistencies among neighboring nodes. And these nodes including R will adjust $\tau_{m,i}$ to the minimal value τ_l . Therefore a malicious node can take advantage of this principle to enforce nodes to broadcast Adv more frequently and eventually waste sensor nodes' energy through inducing them disseminating meaningless Req and producing a lot of Adv messages.

Attack 4: Lower-version Adv

This is another insider attack which is similar to Higher-version attack except that in this attack adversarial nodes could broadcast Adv with lower version than its neighboring nodes'. Therefore this attack induces a larger number of extra communications overhead

Table 1. Five kinds of DoS attacks exploiting the epidemic propagation strategies used by Deluge.

Attack Class	Resource Consumption	Prevention of code propagation or introduction of unnecessary latency
Adv-based	Attack 1, Attack 3, Attack 4	Attack1, Attack 5
Req-based	Attack 2	Attack 2

which lead to resource consumption because many normal nodes which have already the new object profile [2] would send their Adv and code packets to malicious node.

Attack 5: Same-version Adv

This is an attack which aims to introduce unnecessary latency. According to the Deluge’s rules, a node will broadcast its Adv with summary φ only if less than a threshold k advertisements with summary $\varphi' \{v'=v, \gamma'=\gamma\}$ have been received. Therefore, a compromised node could exploit suppression mechanisms of Deluge to reduce the Adv transmission of legitimate nodes through sending multiple bogus advertisements with φ' . As a result, the code updates would not be disseminated efficiently and rapidly.

These five basic kinds of DoS attacks mentioned above can be divided into two classes in terms of attack methods.

Furthermore, the insider DoS attacks against Deluge can be more complex and effective through combining the different basic kinds of DoS attacks. For example, a malicious node can broadcast Higher-version Adv while send Req to neighbor nodes. This hybrid attack could cause much greater damage to the propagation of code updates in WSNs.

4.2. System Models

In this subsection, we set up some simple system models for four different kinds of attacks described above.

$$I_i = \begin{cases} 0 & 0 < t - \frac{\tau_l}{2} h_i \leq \frac{\tau_l}{2} \\ 1 & \frac{\tau_l}{2} < t - \frac{\tau_l}{2} h_i \leq 2\tau_l \\ n & \tau_l 2^{n-2} + \sum_{j=1}^{n-1} \tau_l 2^{j-1} < t - \frac{\tau_l}{2} h_i \leq \min(\tau_l 2^{n-1} + \sum_{j=1}^n \tau_l 2^{j-1}, T), n = \left\lceil \log_2 \left(\frac{\tau_h}{\tau_l} \right) \right\rceil \\ \left\lceil \log_2 \left(\frac{\tau_h}{\tau_l} \right) \right\rceil + \left\lceil \frac{t - h_i - T}{\tau_h} \right\rceil & t - \frac{\tau_l}{2} h_i > T, T = \tau_l (2^{n+1} - 1) + \frac{\tau_h}{2} \end{cases}$$

where $Cost_{tpkt}$ is the communication overhead a node broadcasts an Adv. b is the average amount of neighbors for each node (it is decided by network density and communication range of a sensor node). S_n is the number of sensor nodes in the network. I_i is the number of rounds within Deluge during the $T_{bAdv, h}$. h_i is the number of hops for $node_i$ away from the first malicious node. T is a

Because the impact of the Same-version Adv attack is obvious and easy to understand, we shall not present this model here. Section 5 will give more performance results obtained by simulation.

Attack 1: Higher-version Adv

Firstly we analyze the propagation time of a bogus Adv from one comprised node to all sensor nodes in a WSN through epidemic mechanisms like Deluge. This duration $T_{bAdv, h}$ for a node h hops away from the first malicious node of the bogus Adv is $T_{bAdv, h} = h \cdot T_{Adv}$ where T_{Adv} is the time used by the nodes in advertising their bogus Adv. To calculate T_{Adv} , we need to find the expected number of transmission required for a successful transmission of a packet. Let $P_{one-hop}$ be the probability of a successful transmission of a packet over a single hop. Assuming that the retransmission of a packet is independent, the probability that the number of transmissions of a packet N_{tpkt} equals k is

$$P(N_{tpkt} = k) = (1 - P_{one-hop})^{k-1} P_{one-hop}$$

The expected number of transmissions for a given packet is

$$E[N_{tpkt}] = \sum_{k=1}^{\infty} k(1 - P_{one-hop})^{k-1} P_{one-hop}$$

T_{Adv} could be approximated as follows:

$$T_{Adv} = E[N_{tpkt}] \left(\frac{\tau_l}{2} + T_{MAC} + T_{tpkt} + T_{ppkt} \right)$$

where T_{MAC} is MAC delay for a sing packet. T_{tpkt} is the transmission time for a single packet. T_{ppkt} is the processing time required by a node after receiving the packet. Then the total communication overhead caused by a bogus Adv during $T_{bAdv, h}$ is

$$Cost_{high-version} = Cost_{tpkt} S_n \frac{C_b^{k-1}}{C_b^k} \sum_{i=1}^{S_n} I_i$$

up bound of the time, when this threshold is exceeded, $\tau_{m,i}$ will be set to 60 seconds [2].

Attack 2: False Request

In Deluge we know that one Req packet sent by a malicious node to a neighbor node which has received packets requested will cause 48 code packets as response.

Meanwhile all neighbor nodes of this malicious node will adjust $\tau_{m,i}$ to the minimal value $\tau_{l,i}$. Therefore, the latency introduced by this kind of Attack2, T_{Latency} , consists of the following components:

$$T_{\text{Latency}} = T_{\text{Req}} + T_{\text{GiveUp}} + T_{\text{Code}}$$

where T_{Req} is the time used for requesting the code packets. T_{GiveUp} is the time which is due to the condition when a node exceeds its limit of λ requests, it must transmit to MAINTAIN and wait for another advertisement before making additional requests [2]. T_{Code} is the time required to send 48 code packets.

T_{Req} could be calculated as follows based on system model above:

$$T_{\text{Req}} = E[N_{\text{ipkt}}]E[N_{\text{reqs}}](E[t_r] + T_{\text{MAC}} + T_{\text{ipkt}} + T_{\text{ppkt}})$$

where $E[N_{\text{reqs}}]$ is the expected number of requests a node makes to complete a given page and $E[t_r]$ is the expected time between two requests.

T_{Code} could be calculated as follows:

$$T_{\text{Code}} = 48 \times E[N_{\text{ipkt}}](T_{\text{MAC}} + T_{\text{ipkt}} + T_{\text{ppkt}})$$

The communication overhead caused by a false Req during T_{Latency} is

$$\text{Cost}_{\text{false-req}} = 48 \times E[N_{\text{ipkt}}] \text{Cost}_{\text{ipkt}}$$

Attack 3: Larger-numbered Page

In this attack, the malicious node continuously broadcasts Adv ($v=v'$, $\gamma > \gamma'$) containing high page number. Therefore neighboring nodes always keep MAINTAIN state with high rate of advertisements. The communication overhead caused by this kind of attack is

$$\text{Cost}_{\text{high-pageid}} = \text{Cost}_{\text{ipkt}}(E[N_{\text{reqs}}] + S_{\text{one-hop}} \frac{C_b^{k-1}}{C_b^k} \frac{t}{\tau_l})$$

where t is attack time caused by a malicious node. $S_{\text{one-hop}}$ is the number of contending nodes of neighboring nodes of the malicious node in one-hop range.

Attack 4: Lower-version Adv

In that attack a malicious node pretend having an object with old version. As a result, this node will require the transfer of all pages in the code image. The communication overhead caused by Lower-version Adv attack is given by

$$\begin{aligned} \text{Cost}_{\text{lower-version}} = & \text{Cost}_{\text{ipkt}} E[N_{\text{ipkt}}] (S_{\text{one-hop}} \frac{C_b^{k-1}}{C_b^k} \frac{t}{\tau_l} + E[N_{\text{reqs}}] \\ & + E[N_{\text{GiveUp}}] + 48 \times N_{\text{page}}) \end{aligned}$$

where t is attack time caused by a malicious node. N_{page} is the number of pages in code updates.

5. Evaluation

We have provided system models of basic insider DoS attacks against Deluge in Section 4. In this subsection,

we compare normal Deluge's performance to that of attacking in different cases through using TOSSIM [21] which is a bit-level node simulator designed specifically for the TinyOS platform.

We use two performance metrics in our evaluation: *Communication overhead* and *update completion time*.

The communication overhead is measured as the total number of packets transmitted by all the sensor nodes during a code dissemination, which is related to energy consumption. The update completion time is the time required to finish disseminating a code image to all the sensor nodes in the network.

In the simulation, we set the parameters of Deluge as following. For the maintenance service, we set $\tau_r = 2$ seconds, $\tau_h = 60$ seconds, and $k = 1$. For requests, we set $\tau_r = 0.5$, $\lambda = 2$, and $\omega = 8$. And for each set of results, we perform the simulation 10 times using the same topology and then take an average over them. The results of these simulations are presented in the following subsections

Attack 1: Higher-version Adv

In this kind of DoS attack, we repeat the simulation with a 10×10 grid topology network with adjacent nodes spaced 15 feet apart (see Figure 1). The black node denotes a malicious node. And the grey node represents the first node which can receive packets from the base station.

Figure 2 shows the communication overhead of different conditions which are measured as the total number of Adv broadcasted by all sensor nodes in each test case. For the case of free attack, Deluge takes about 10.4 seconds for all nodes in the network to update their image version to v' . And in this duration the total number of Adv under free attack is transferred only 30% less than the condition under High-version Adv attack.

Figure 3 shows the number of each packet type sent by all sensor nodes during disseminating a code image in the two different conditions respectively.

Among all the simulations, the average number of Adv packets sent under Higher-version Adv attack is 7.5 times more than that of the free attack condition, while the average number of code packets sent under Attack1 is reduced by approximately 90%.

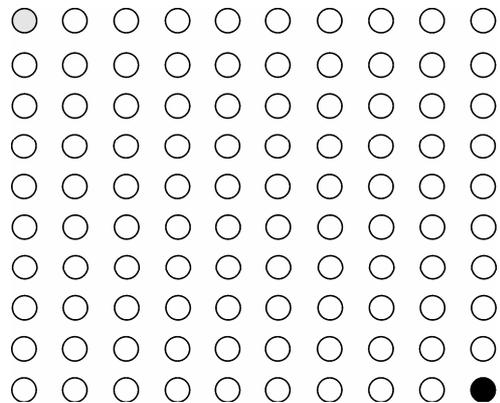


Figure 1. Topology of WSN.

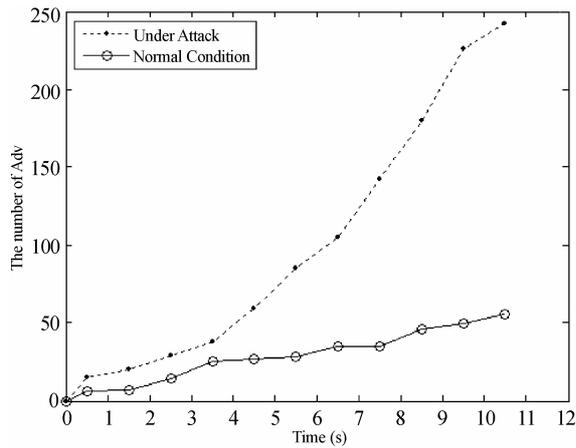


Figure 2. The number of Adv of free attack and Attack1.

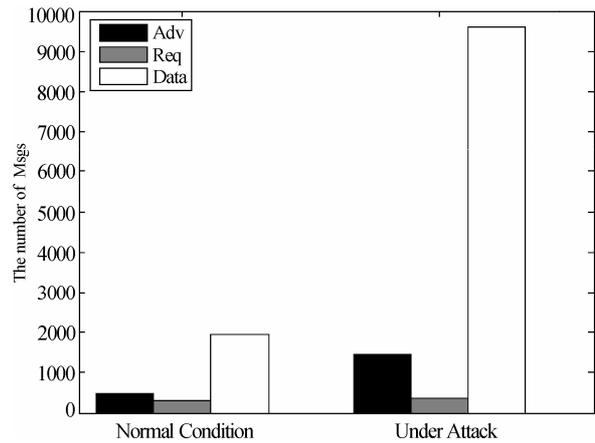


Figure 4. Communication overhead of free attack and Attack2.

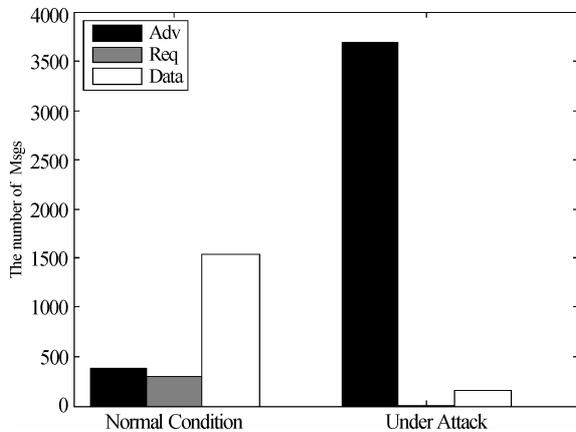


Figure 3. Communication overhead of free attack and Attack1.

As we explained earlier, the main reason for this performance difference is that the malicious node enforces legitimate nodes to transmit more Adv packets by using the dynamic adjustments of the advertisement rate mechanisms. When the bogus Adv contains a higher image version than the image injected by base station, infected nodes are prevented from requesting the new image from base station. Therefore the number of Req and code packets decrease dramatically.

Attack 2: False Request

In this simulation with a 10×10 grid topology network with neighboring nodes spaced 15 feet apart. We let a malicious node always propagate Req for the first page of a new image.

Figure 4 shows the number of each packet type sent by all nodes during disseminating a code update in the two cases respectively.

From Figure 4 we can see that under free attack condition, the control packets (Adv and Req) occupy about 18.18% of total messages, while code packets take the left 81.82%. This results fits fairly well that of Deluge [2]. In contrast, under the False Request attack condition, the number of code packets increases 3.9 times more than that of the normal condition) due to bogus Req sent by the adversary.

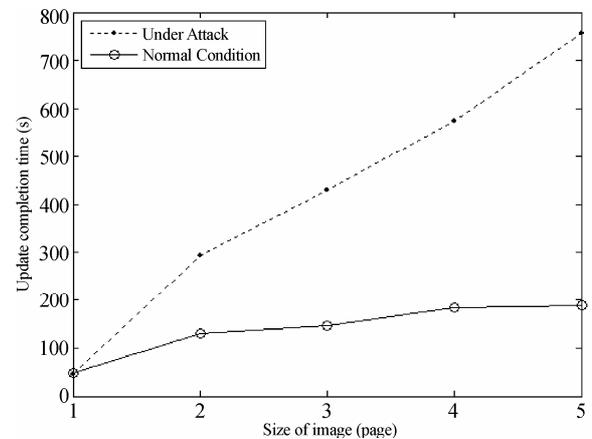


Figure 5. Update completion time of free attack and Attack2.

Figure 5 shows the update completion time of the whole network for different size of images in two conditions. Under the False Request attack, when five pages of the code image have been disseminated, it takes 4 times longer than that of the free attack. From Figure 5 we can also see that when there is only one page in an image, this attack could not block reprogramming because the malicious node always propagates the Req for first page.

The additional latency introduced by Attack2 is due to suppressing other Req from normal nodes when the adversary sends Req modified which included a smaller page number.

These simulation results demonstrate that the False Request attack introduces a large amount of unnecessary transmission of code packets which would cause energy consumption of resource-constrained sensor nodes.

Attack 3: Larger-numbered Page

We simulate this attack in a 10×10 grid topology network with nodes spaced 15 feet apart and set the attack time to be 60 seconds. During this period, the malicious node continuously broadcasts Adv ($v=v', \gamma>\gamma'$) to its neighboring nodes.

Figure 6 shows the number of each packet type sent by all sensor nodes during disseminating a code image in the two different conditions respectively.

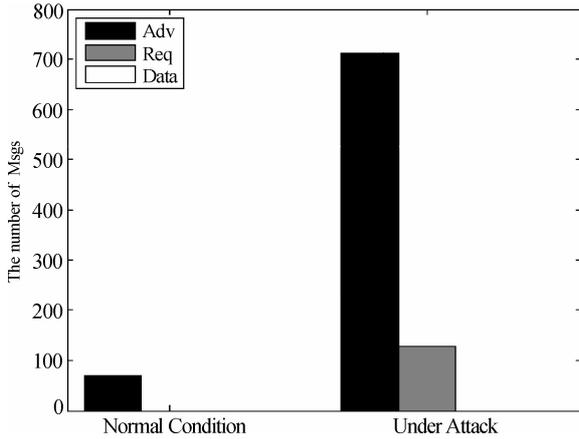


Figure 6. Communication overhead of free attack and Attack3.

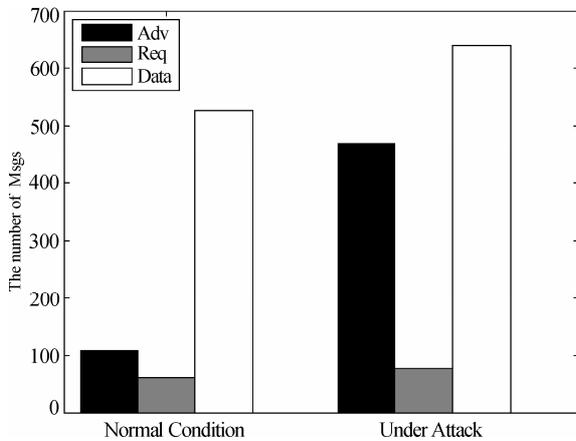


Figure 7. Communication overhead of free attack and Attack4.

In Figure 6, the average number of Adv packets sent under Large-numbered Page attack is 7.8 times more than that of the free attack condition, while the average number of Req packets sent under Attack3 increase approximately to 150 packets. Because $\tau_{m,i}$ is minimized, the rate of advertisements increases, the total number of Adv is much greater compared with that of free attack condition.

These simulations confirm that Large-numbered Page attack eventually waste sensor nodes' energy through inducing a number of meaningless Req and Adv packets.

Attack 4: Lower-version Adv

The simulations were performed in a 10×10 grid topology network with nodes spaced 15 feet apart. Figure 7 shows the number of each packet type sent by all sensor nodes during disseminating a code image in the two different conditions respectively.

In Figure 7, the average number of code packets under Lower-version Adv attack is about 17% more than that of the case under free attack. Meanwhile the average number of Adv packets under Attack4 increase approximately to 500 packets which are about 2 times more than that of the free attack.

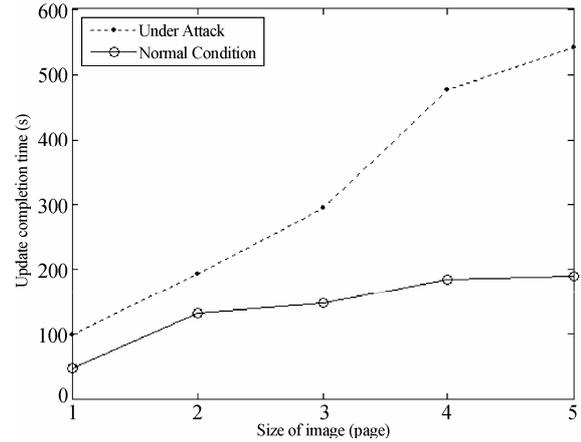


Figure 8. Update completion time of free attack and Attack4.

Figure 8 shows the update completion time of the whole network for different size of images in two conditions.

From Figure 8 we can see that when the network is under attack the larger code image is, the longer the delay would be compared with normal condition. Under the Lower-version Adv attack, when five pages of the code image have been disseminated, it takes 2.6 times longer than that of the free attack.

From the simulation results we know the main reason for this performance is that this attack can cause the delay of code dissemination while extra resource consumption is introduced.

Attack 5: Same-version Adv

In this simulation with a 10×10 grid topology network with neighboring nodes spaced 15 feet apart. Figure 9 shows the update completion time of the whole network for different size of images in two conditions.

Under the Same-version Adv attack, when five pages of the code image have been disseminated, it takes about 42% longer than that of the free attack.

The additional latency introduced by Attack2 is due to a compromised node could exploit suppression mechanisms of Deluge to reduce the Adv transmission of legitimate nodes.

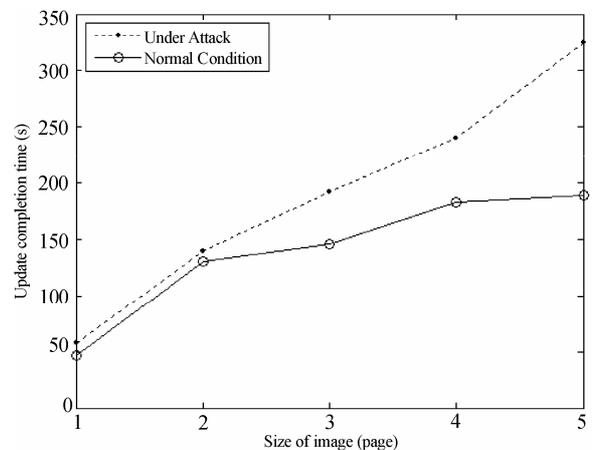


Figure 9. Update completion time of free attack and Attack5.

6. Conclusions and Future Work

In this paper we identify five different basic types of insider DoS attacks exploiting the epidemic propagation strategies used by Deluge. They are Higher-version Advertisement attack, False Request attack, Larger-numbered Page attack, Lower-version Adv attack, and Same-version Adv attack. We also proposed the simple system models for these DoS attacks to try to find out the impact of those attacks on Deluge. Despite the fact that Deluge is an efficient protocol for code propagation in WSNs, it is susceptible to different kind of attacks. To understand more deeply about them, we simulate these five basic types of insider DoS attacks by using TOSSIM and report the detailed statistical results.

There are still many issues that need further investigation to make reprogramming highly available although some recent works have attempted to provide DoS-Resistant code dissemination in WSNs [23,24]. Y. Zhang, et al. proposed a public-key scheme called "combined public key" to secure Advertisement and Request packets. The ignorable problem with the approach is the resource requirement [24]. Seluge is the latest work on secure code dissemination and is a solution that seamlessly integrates the security mechanism and original deluge. Unfortunately, although the current version of Seluge adopts the cluster key approach to provide authentication of ADV and SNACK packets [23], it cannot uniquely identify senders. As a result, a compromised node can still pretend to be its neighbors using their cluster keys to launch DoS attacks. In our future work, we will investigate techniques to detect insider DoS attacks exploiting the Deluge epidemic and suppression mechanisms. Finally, we plan to provide Insider-DoS-Resistant code dissemination scheme.

7. Acknowledgment

We would like to thank the anonymous reviewers for their valuable comments.

8. References

- [1] T. Stathopoulos, J. Heidemann, and D. Estrin, "A remote code update mechanism for wireless sensor networks," Technical Report, UCLA, Los Angeles, CA, USA, 2003.
- [2] J. W. Hui and D. Culler, "The dynamic behavior of a data dissemination protocol for network programming at scale," in ACM International Conference on Embedded Networked Sensor Systems, pp. 81–94, November 2004.
- [3] S. S. Kulkarni and L. Wang, "MNP: Multihop network reprogramming service for sensor networks," in International Conference on Distributed Computing Systems (ICDCS'05), June 2005.
- [4] S. S. Kulkarni and M. Arumugam, "INFUSE: A TDMA based data dissemination protocol for sensor networks," Technical Report, Michigan State University, East Lansing, MI, USA, 2004.
- [5] V. Naik, *et al.*, "Sprinkler: A reliable and energy efficient data dissemination service for wireless embedded devices," 26th IEEE Real-Time System Symposium, December 2005.
- [6] L. A. Phillips, "Aqueduct: Robust and efficient code propagation in heterogeneous wireless sensor networks," Master's thesis, University of Colorado at Boulder, 2005.
- [7] M. D. Krasniewski, R. K. Panta, S. Bagchi, C. L. Yang, and W. J. Chappell, "Energy-efficient on-demand reprogramming of large-scale sensor networks," ACM Transactions on Sensor Networks, 4(1): pp. 1–38, 2008.
- [8] P. E. Lanigan, R. Gandhi, and P. Narasimhan, "Sluice: Secure dissemination of code updates in sensor networks," in IEEE International Conference on Distributed Computing Systems, Lisbon, Portugal, July 2006.
- [9] P. K. Dutta, J. W. Hui, D. C. Chu, and D. E. Culler, "Securing the Deluge network programming system," in the Fifth International Conference on Information Processing in Sensor Networks (IPSN'06), 2006.
- [10] J. Deng, R. Han, and S. Mishra, "Secure code distribution in dynamically programmable wireless sensor networks," in Proceedings of the Fifth International Conference on Information Processing in Sensor Networks (IPSN'06), April 2006.
- [11] Crossbow Tech Inc., Mote In-Network Programming User Reference, <http://www.tinyos.net/tinyos-1.x/doc/Xnp.pdf>, 2003.
- [12] TinyOS: An open-source OS for the networked sensor regime, <http://www.tinyos.net/>.
- [13] P. E. Lanigan, P. Narasimhan, R. Gandhi, "Tradeoffs in configuring secure data dissemination in sensor network: An empirical outlook: [CMU-CyLab-07-006]," CyLab, Carnegie Mellon University, PA, 2007.
- [14] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wem, and D. E. Culler, "SPINS: Security protocols for sensor networks," Wireless Networks, 8(5): pp. 521–534, 2002.
- [15] H. Chan, A. Perrig, and D. Song, "Random key pre-distribution schemes for sensor networks," in IEEE Symposium on Research in Security and Privacy, 2003.
- [16] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," Sensor Network Protocols and Applications (SNPA 03), May 2003.
- [17] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," IEEE Computer, pp. 48–56, October 2002.
- [18] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy efficient communication protocol for wireless microsensor networks," in the 33rd Hawaii International Conference on System Sciences, 2000.
- [19] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks," NSDI 2004, pp. 15–28, 2004.
- [20] J. Deng, R. Han, and S. Mishra, "Practical study of transitory master key establishment for wireless sensor

- networks,” in 1st IEEE/CreateNet Conference on Security and Privacy in Communication Networks (SecureComm 2005), Athens, Greece, pp. 289–299, September 2005.
- [21] P. Levis, N. Lee, M. Welsh, and D. Culler, “TOSSIM: Accurate and scalable simulation of entire tinyos applications,” in Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003), ACM Press, November 2003.
- [22] D. R. Raymond and S. F. Midkiff, “Denial-of-service in wireless sensor networks: Attacks and defenses,” IEEE Pervasive Computing, 2008.
- [23] S. Hyun, P. Ning, A. Liu, and W. L. Du, “Seluge: Secure and DoS-resistant code dissemination in wireless sensor networks,” In Proceedings of the Seventh International Conference on Information Processing in Sensor Networks (IPSN’08), April 2008.
- [24] Y. Zhang, X. S. Zhou, Y. M. Ji, Z. Y. Fang, and L. F. Wang, “Secure and DoS-resistant network reprogramming in sensor networks based on CPK,” 4th IEEE International Conference on Wireless Communications, Networking and Mobile Computing, 2008.

On the Scalable Fairness and Efficient Active Queue Management of RED

Hui WANG¹, Xiao-Hui LIN*¹, Kai-Yu ZHOU², Nin XIE^{1,4}, Hui LI³

¹*Department of Communication Engineering, Shenzhen University, Shenzhen, China*

²*China Telecom Beijing Research Institute, Beijing, China*

³*Shenzhen Graduate School, Peking University, Shenzhen, China*

⁴*National Mobile Communications Research Laboratory, Southeast University, Nanjing, China*

Email: *xhlin@szu.edu.cn

Received October 22, 2008; revised December 2, 2008; accepted December 31, 2008

Abstract

Internet routers generally see packets from a fast flow more often than a slow flow. This suggests that network fairness may be improved without per-flow information. In this paper, we propose a scheme using Most Recently Used List (MRUL)—a list storing statistics of limited active flows that sorted in most recently seen first mode—to improve the fairness of RED. Based on the list, our proposed scheme jointly considers the identification and punish of the fast and unresponsive fast flows, and the protection of slow flows. Its performance improvements are demonstrated with extensive simulations. Different from the previous proposals, the complexity of our proposed scheme is proportional to the size of the MRUL list but not coupled with the queue buffer size or the number of active flows, so it is scalable and suitable for various routers. In addition, another issue we address in this paper is queue management in RED. Specifically, we replace the linear packet dropping function in RED by a judicially designed nonlinear quadratic function, while original RED remains unchanged. We call this new scheme Nonlinear RED, or NLRED. The underlying idea is that, with the proposed nonlinear packet dropping function, packet dropping becomes gentler than RED at light traffic load but more aggressive at heavy load. As a result, at light traffic load, NLRED encourages the router to operate in a range of average queue sizes rather than a fixed one. When the load is heavy and the average queue size approaches the pre-determined maximum threshold (i.e. the queue size may soon get out of control), NLRED allows more aggressive packet dropping to back off from it. Simulations demonstrate that NLRED achieves a higher and more stable throughput than RED and REM. Since NLRED is fully compatible with RED, we can easily upgrade/replace the existing RED implementations by NLRED.

Keywords: Random Early Detection, TCP, Unresponsive Flows, Fairness, Queue Management

1. Introduction

With the increasing popularity of stream media applications, the fairness of networks has attracted much research attention [1–11]. With these research efforts, a number of schemes [4–12] were proposed to improve the fairness in networks with modifications to the queue management schemes implemented in Internet routers.

Known as Active Queue Management (AQM), Random Early Detection (RED) [13] is recommended by IETF for queue management in routers. However, past work shows that unfairness of RED may occur with RED

under two conditions. Firstly, when two or more TCP flow with different RTT competing for the bottleneck bandwidth, RED tends to let the flow with shorter RTT use more bandwidth [9]. Secondly, when responsive TCP flows shares a RED router with unresponsive UDP flows, unresponsive UDP flows may has unreasonable high throughput than TCP flows [8].

Although per-flow queue (i.e. Fair Queuing [4,12]) is the most direct solution to the unfair problems, with the large number of flows possibly sharing a link, it is not scalable for an Internet router. Notice that a router sees packets from a fast flow more often than a slow flow; we propose in this paper the Scalable Fair Random Early

Detection (SFRED) to improve the fairness of RED. A Most Recent Used List (MRUL) storing up to N most active connections' traffic statistics is maintained by SFRED. Based on the list, SFRED has jointly considered the punishment of fast flows, unresponsive fast flows, and the protection of short life slow flows (e.g. WEB applications). Simulations demonstrate that SFRED has significantly improved the fairness of RED. The complexity of SFRED is proportional to the size of the MRUL but not coupled with the queue buffer size or the number of active flows, so it is scalable and suitable for various routers.

Another issue we address in the paper is efficient queue management in RED. Among various AQM schemes, RED is probably the most extensively studied. RED is shown to effectively tackle both the global synchronization problem and the problem of bias against bursty sources. Due to its popularity, RED (or its variants) has been implemented by many router vendors in their products (e.g. Cisco implemented WRED). On the other hand, there is still a hot on-going debate on the performance of RED. Some researchers claimed that RED appears to provide no clear advantage over drop-tail mechanism. But more researchers acknowledged that RED shows some advantages over drop-tail routers but it is not perfect, mainly due to one or more of the following problems.

- RED performance is highly sensitive to its parameter settings. In RED, at least 4 parameters, namely, maximum threshold (max_{th}), minimum threshold (min_{th}), maximum packet dropping probability (max_p), and weighting factor (ω_q), have to be properly set.
- RED performance is sensitive to the number of competing sources/flows.
- RED performance is sensitive to the packet size.
- With RED, wild queue oscillation is observed when the traffic load changes.

As a result, RED has been extended and enhanced in many different ways. It can be found that a common underlying technique adopted in most studies is to steer a router to operate around a fixed target queue size (which can either be an average queue size or an instantaneous queue size). There are some concerns on the suitability of this approach, since the schemes thus designed are usually more complicated than the original RED. This renders them unsuitable for backbone routers where efficient implementation is of primary concern. In some schemes, additional parameters are also introduced. This adds extra complexity to the task of parameter setting. Unlike the existing RED enhancement schemes, we propose to simply replace the linear packet dropping function in RED by a judiciously designed nonlinear quadratic function. The rest of the original RED remains unchanged. We call this new scheme Nonlinear RED, or NLRED. The underlying idea is that, with the proposed nonlinear packet dropping function, packet dropping is gentler than RED at light traffic load but more aggressive at heavy load. Therefore, at light traffic load NLRED

encourages the router to operate in a *range* of average queue sizes rather than a fixed one. When the load is heavy and the average queue size approaches the maximum threshold max_{th} —an indicator that the queue size may soon get out of control, NLRED allows more aggressive packet dropping to quickly back off from it. Simulations demonstrate that NLRED achieves a higher and more stable throughput than RED and REM, an efficient variant of RED. Since NLRED is fully compatible with RED, we can easily upgrade/replace the existing RED implementations by NLRED.

The rest of this paper is organized as following. In Section 2, we introduce the background and the related work of RED, together with the proposed SFRED algorithm. In Section 3, we give the simulation results of SFRED. In Section 4, we present Nonlinear RED. This is followed by extensive simulations in Section 5. Finally, we give the concluding remarks in Section 6.

2. Scalable Fair RED

RED [13] provides high throughput while keeping short queue length (i.e. queuing delay) at the routers. However, [8,9] have shown that RED has fairness problems. When two or more TCP flow with different RTT competing for the bottleneck bandwidth, RED tends to let the flow with shorter RTT use more bandwidth [9]. Similarly, when responsive TCP flows shares a RED router with unresponsive UDP flows, unresponsive UDP flows may have unreasonable high throughput than TCP flows [8].

LRU-RED [4] was developed based on a LRU list to identify high bandwidth flows. This scheme derived from the fact that a router should see a packet from a fast flow more often than a slow flow, so that the number of states to be kept for maintaining the fairness can be bounded. However, we find the design of LRU-RED in [5] is too rough to fully utilize the potential of LRU table. For example, it cannot identify an unresponsive flow so that the second unfairness condition mentioned in Section 1 cannot be solved. Motivated by this, in this paper, we will propose a new scalable fair AQM scheme, the SFRED. SFRED is developed based on a list similar to the LRU but the list (MRUL) keeps more information. SFRED has combined many novel ideas in previous work in its design, such as the punishment of unresponsive flows [8,10] and the protection of slow flows [2].

The fairness of Scalable Fair RED (SFRED) is enforced in three steps, namely, *identifying and limiting fast flows, identifying and punishing unresponsive fast flows, and protecting slow flows*. In this paper, we consider that the SFRED working in “packet mode.” That is, all the computations of throughput and bandwidth allocation are in packets. However, the proposed mechanisms can be easily extended to work with all the throughputs and bandwidth allocations computed in bits (i.e. “bit mode”).

2.1. MRUL

From the viewpoint of fairness, the flows in networks can be a *fast* or a *slow* flow. A fast flow transmits faster than the fair rate and may interfere with the transmission of other flows; a slow flow utilizes no more than the fair bandwidth.

A fast flow has packets arrives at the router more often than a slow flow. In other words, given the set of fast flows in a router, with packets in the queue sorted with their arrival time, we should able to find at least one packet from each of the fast flow before reaching the head of the queue (searching the queue from the end, i.e. most recently received packet, to the head i.e. lest recently received packet). This suggests that it is possible to build a scalable fair queue management algorithm with limited complexity. Notice that a fair queue management scheme concerns only the aggregated traffic characteristics and the characteristics of the fast flows. The aggregated traffic characteristics are needed for determining the fair rate. The characteristics of fast flow are needed for determining the per flow punishment. Motivated by this, we develop the Scalable Fair RED (SFRED) based on the Most Recent Used List (MRUL).

In SFRED, a router maintains a linked list (MRUL) for up to N most recently seen active flows (flows that has packet routed through the router recently). This linked list keeps simple traffic statistics for each of the active flows in list, such as the number of packet received (H_{ri} , $i=1,2,\dots,N$) and dropped (H_{di} , $i=1,2,\dots,N$). Besides, SFRED also keeps the number of packet received (H_r), dropped (H_d) for the aggregated traffic, and the total number of activate flows (n).

The MRUL is maintained as follows. Upon receiving a packet, SFRED searches the list for a node matching the address of the arrived packet. If it is not found, SFRED creates a new item for the flow as the list header, and the new item is initialized with $H_r=1$ and $H_d=0$. Otherwise, if a node matching the address is located, SFRED increases the number of packet received (H_r) by one and moves this item to the list header. SFRED then processes and checks if the packet should be dropped, and changes the number of packet dropped (H_d) accordingly. When there are already N nodes in the list (i.e. the list is full), SFRED deletes the tail node before creates the new one.

Based on the data stored in MRUL, SFRED performs the identification of slow, fast, and unresponsive fast flows, as well as determines the particular punishment to the fast and unresponsive fast flows.

2.2. Identify Fast Flow

To be fair, a queue management scheme should first be able to identify the fast flow. In SFRED, the fair rate T_f is computed as:

$$T_f = \frac{H_r - H_d}{n}$$

With the MRUL, the throughput of an active flow can

be shown with the number of packet received (H_{ri}) as

$$T_i = H_{ri}$$

So a flow is a fast flow, if

$$T_i > (1 + \alpha)T_f \quad (1)$$

where α is a constant set to 0.1 in this paper.

Once a fast flow is identified, packets from this flow are processed by normal RED with the loss probability increase by T_f/T_i times, as

$$max_{pd} = (T_f max_p) / T_i$$

where max_p is the original maximum dropping probability of RED, and max_{pd} is the maximum dropping probability for the fast flow.

Generally, such increases in loss probability is sufficient for a responsive TCP flow, since the TCP analytical models [14] indicates that TCP throughput is inversely proportional to the packet loss rate. Considering the bursty nature of TCP transmission, this is also necessary as applying a strict bandwidth limitation (e.g. dropping all packets received from a flow except the first five in each 0.5s interval) results in lower than fair rate throughput for a TCP flow.

2.3. Punish Unresponsive Fast Flows

However, simply increasing the dropping probability has been proved to be not effective for the unresponsive flows [8], such as consistent bit rate user datagram protocol flows (CBR-UDP). An unresponsive flow does not dynamically change its throughput with network state (e.g. the packet loss rate). In other words, it does not adopt the similar congestion control mechanisms as TCP. Thus, to maintain the fairness of networks, when there are unresponsive fast flows, the queue management scheme should take a more actively part in punishing them.

In SFRED, the identification of unresponsive fast flows is based on analyzing the drop history of each fast flow that performs similar with the method adopted to identify fast flows in [10]. Notice that an unresponsive fast flow does not changes its transmission rate with the packet loss rate. When it shares a SFRED queue with some responsive flows, comparing with the responsive flows, it has a higher packet loss rate as well as a higher transmission rate. On the contrary, a TCP flow cannot maintain high throughput under a high loss rate. Thus an unresponsive fast flow is identified by comparing the per-flow/average loss rate and the per-flow/average throughput. From sub-section 2.1, the packet loss rate for a flow is

$$L_i = H_{di} / H_{ri}$$

The average packet loss rate is $L = H_d / H_r$.

Thus, when (1) and $L_i > (1 + \beta)L$,

where β is a constant set to 0.1 in this paper, a unresponsive fast flow is identified. The identified flow is then applied with a deterministic packet loss

probability of 1. In other words, if a flow has a packet income rate higher than fair bandwidth and at the same time its packet drop rate is higher than the average loss rate, the flow is identified as an unresponsive fast flow and all the subsequent packets from the flow are dropped.

2.4. Protect Slow Flows

The requirement of protecting slow TCP flows, i.e. protecting TCP flows from a packet loss at the slow-start phase, arises from the fact that most short life TCP flows are low slow flows. With limited data to be transmitted, a single short life flow generally cannot reach its full transmission rate before the connection is terminated. However, the performance of short life connection is important for the overall Internet performance.

In SFRED, a new flow seen by the router is protected from experiencing packet loss by modifying the threshold of RED. That is, when a packet is received and the MRUL shows that the flow is with $H_{ri} < \eta$ and zero H_{di} , where η is a constant, the flow is treated as a slow flow. To determine whether the packet should be dropped, SFRED calls RED with both the minimum threshold and maximum thresholds increased by k packets. This equals allocating k packets buffer in the queue to protect slow flow. Because this buffer is still a part of the queue, when congestion becomes severe, slow flows will experience packet loss so that their throughputs are controlled by SFRED. Because the congestion window of a TCP connection reaches a level that able to generate duplicate ACK after send out about 10 packets, the constant η is set to 10.

3. Simulation Validation of SFRED

We evaluate the performance of SFRED by simulating the network in Figure 1 with NS2 [15] simulator. The senders (noted S_i , $i=1, \dots, n$) are linked to router G0 with 10Mbps links, with variable propagation delay τ_i ms. A common receiver R is linked to router G1 with a 10Mbps link with delay τ_r ms. From Figure 1, the link between routers G0 and G1 is the bottleneck of the network, with a bandwidth of 1.5Mbps and a delay τ_c ms. Note that all the delays, namely, τ_i , τ_r , and τ_c , are variable. For the RED parameters [13], unless otherwise stated, we use minimum threshold $minth=5$, maximum threshold $maxth=15$, weighting factor $w_q=0.002$, and maximum dropping probability $max_p=0.1$, with a fixed buffer size of 50 packets.

The traffic simulated in the network includes long life FTP, CBR-UDP, and short life Web-like flows. In all the simulations, packet size of 1KB is used for TCP flows (FTP and Web-like). The receiver's advertised window is set to two times of the bandwidth delay products, so

that it does not limit the throughput of a flow. For a CBR-UDP flow, it transmits a UDP packet with size X bytes every t_d s, where X is fixed for a flow but may differ between flows. The path from S_i to R (forward path) always carries the data packets, while the reverse path from R to S_i carries the ACK packets.

The Web-like flows are implemented to acquire enough simulation results with short life TCP connections, and avoid collapsing the simulator with too many resource allocation requests for new connections. A Web-like flow is designed to get a small amount of data (D_w packets) each time. Upon finishing the current transmission, it resets the connection state (so that its transmission restarts with slow start) and transmits another D_w packets.

3.1. Fairness among TCP Flows

Figures 2 and 3 show the bandwidth usage of two FTP flows under RED and SFRED queues, with $\tau_1=10$ ms and $\tau_2=100$ ms, $\tau_c=10$ ms, and $\tau_r=5$ ms. Each marker on the curves represents a 10s average of the TCP throughput traced at router G0. With much less margin between the two curves, Figures 2 and 3 show that SFRED achieved fairer bandwidth allocation than RED. Figures 4–6 show the simulation results with 30 Web-like flows and 2 Ftp flows, over 2000s of simulation duration. The simulation parameters are $\tau_1=\tau_2=\dots=\tau_{30}=\tau_c=10$ ms, $\tau_{31}=10$ ms, $\tau_{32}=100$ ms, $\tau_r=3$ ms, and $D_w=10$ packets. Figure 4 shows the bandwidth usage of the two Ftp flows under the RED queue and Figure 5 shows similar results under the SFRED queue. The dash dotted curve presents the theoretical fair rate—the bottleneck bandwidth over the total number of flows. Again we see SFRED enables much efficient fair bandwidth allocation than RED.

Figure 6 shows the cumulative distribution function of the data transfer delay of the 30 Web-like flows. From this figure, the transfer delay of Web-like flows under SFRED is more stable than RED, in that most of the transmissions finish in 2s. The improvement to transfer delay performance is further presented statistically with the variance and mean in Table 1. It shows that SFRED performs better in the both metrics.

Table 1. Statistic of data transfer delay, 30 web and 2 FTP flows.

	RED	SFRED
mean(s)	2.057	1.825
Var	18.1812	3.977

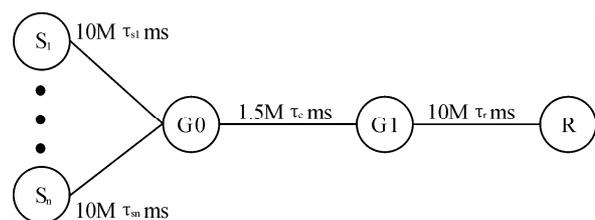


Figure 1. The network simulated.

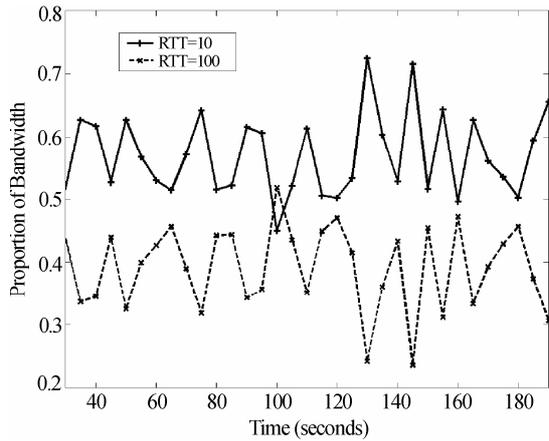


Figure 2. Bandwidth usage of two FTP flows, with $\tau_1=10\text{ms}$ and $\tau_2=100\text{ms}$, under RED queues.

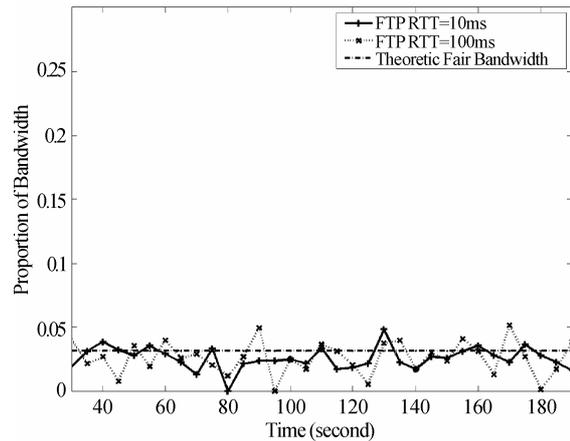


Figure 5. Bandwidth usage of two FTP flows, with $\tau_1=10\text{ms}$ and $\tau_2=100\text{ms}$, under SFRED queues, with 30 web-like flows.

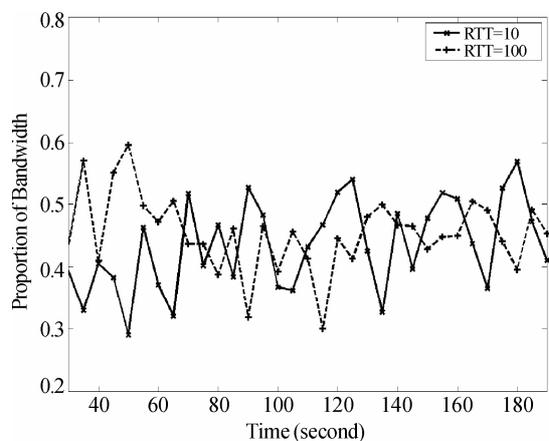


Figure 3. Bandwidth usage of two FTP flows, with $\tau_1=10\text{ms}$ and $\tau_2=100\text{ms}$, under SFRED queues.

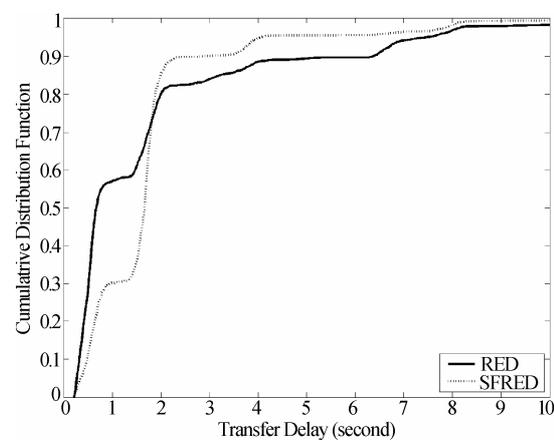


Figure 6. Cumulative distribution function of the transmission delay, 30 web and 2 FTP flows.

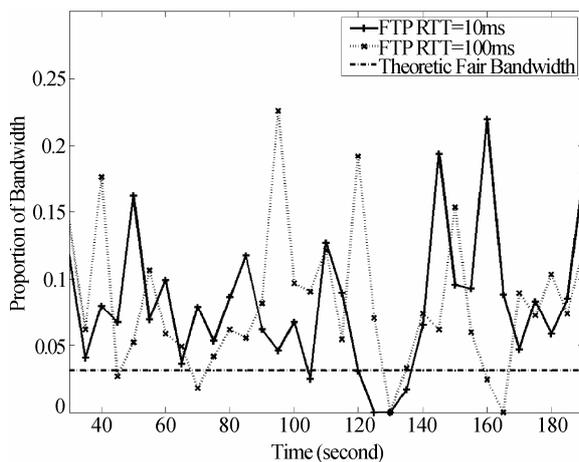


Figure 4. Bandwidth usage of two FTP flows, with $\tau_1=10\text{ms}$ and $\tau_2=100\text{ms}$, under RED queues, with 30 web-like flows.

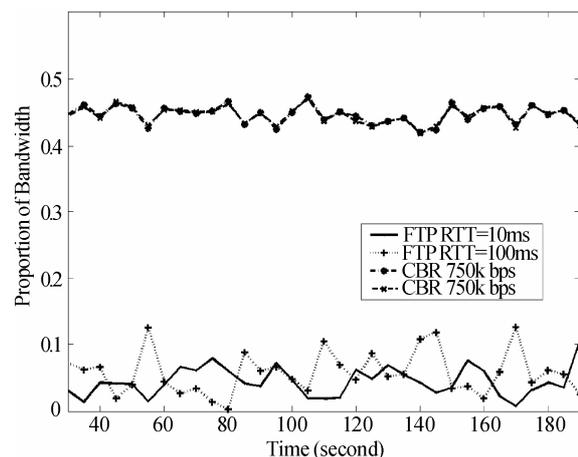


Figure 7. Instantaneous bandwidth usage of two FTP flows and two CBR-UDP flows, under RED queues.

3.2. With CBR-UDP Flows

Figures 7 and 8 show the bandwidth of 4 flows. Two of them are CBR-UDP flows, with $\tau_1=10\text{ms}$, $X_1=1\text{KB}$, $t_{d1}=10\text{ms}$, $t_2=100\text{ms}$, $X_2=500\text{B}$, and $t_{d2}=5\text{ms}$. The other

two flows are Ftp flows, with $\tau_3=10\text{ms}$ and $\tau_4=100\text{ms}$. $\tau_c=10\text{ms}$ and $t_r=5\text{ms}$. From the figures, it is clear that the implementation of SFRED protected TCP flows by punishing the unresponsive UDP flows, although rigorous fairness is still not attained.

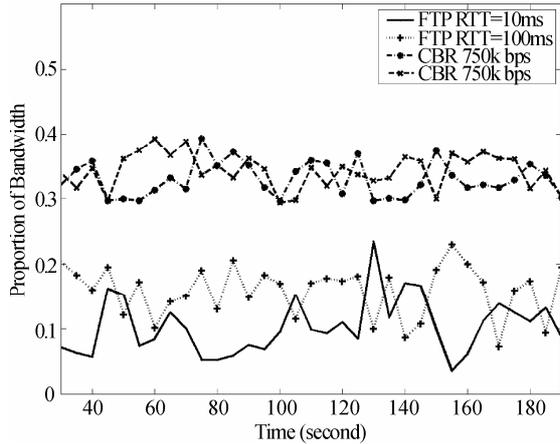


Figure 8. Instantaneous bandwidth usage of two FTP flows and two CBR-UDP flows, under SFRED queues.

4. Nonlinear Random Early Detection

4.1. NLRED Algorithm

RED was mainly designed to overcome the two problems associated with drop-tail routers, namely, global synchronization and bias against bursty sources. Unlike the drop-tail mechanism, RED measures congestion by the average queue size and drops packets randomly before the router queue overflows. When a packet arrives at a router, the average queue size, denoted avg , is updated using the following exponentially weighted moving average (EWMA) function,

$$avg = (1 - \omega_q)avg' + \omega_q q$$

where avg' is the calculated average queue size when the last packet arrived, q is the instantaneous queue size, and ω_q is the pre-determined weighting factor with a value between 0 and 1.

As avg varies from a minimum threshold min_{th} to a maximum threshold max_{th} , the packet dropping probability p_d increases linearly from 0 to a maximum packet dropping probability max_p , or

$$p_d = \begin{cases} 0 & avg \leq min_{th} \\ \frac{avg - min_{th}}{max_{th} - min_{th}} max_p & min_{th} \leq avg \leq max_{th} \\ 1 & max_{th} \leq avg \end{cases} \quad (2)$$

The throughput performance of RED is not stable. For example, when the traffic load is very light and RED parameters are aggressively set or when the traffic load is very heavy and the parameters are tenderly set, the throughput is low. It has been shown that no single set of parameters for RED could get a stable performance under different traffic loads. We believe such instability is due, at least in part, to the linear packet dropping function adopted by RED, which tends to be too

aggressive at light load, and not aggressive enough when the average queue size approaches the maximum threshold max_{th} . We also believe that the performance improvement of some previous work is at least partly due to the employment of nonlinear dropping function, either intentionally or unintentionally. (More reasons to be provided later.) However, we notice that these improvements may not be suitable for core routers, as their corresponding nonlinear dropping functions greatly complicate the basic mechanism of RED. In this paper, we propose to replace the linear packet dropping probability function by a judiciously designed quadratic function. The resulting scheme is called non-linear RED or NLRED. The pseudocode of NLRED is summarized in Figure 9.

When avg exceeds the minimum threshold, NLRED uses the nonlinear quadratic function shown in (3) to drop packets, where max'_p represents the maximum packet dropping probability of NLRED. Figure 10 compares the packet dropping functions for RED and NLRED. (The choice of a quadratic function is further explained in the next subsection.)

$$p'_d = \begin{cases} 0 & avg \leq min_{th} \\ \left(\frac{avg - min_{th}}{max_{th} - min_{th}}\right)^2 max'_p & min_{th} < avg \leq max_{th} \\ 1 & max_{th} < avg \end{cases} \quad (3)$$

Comparing (3) to the dropping function of original RED in (2), if the same value of max_p is used, NLRED will be gentler than RED for all traffic load. This is because the packet dropping probability of NLRED will always be smaller than that of RED. In order to make the two schemes have a comparable total packet dropping probabilities, we set $max'_p = 1.5max_p$, such that the areas covered by both dropping functions from min_{th} to max_{th} are the same, or

$$\int_{min_{th}}^{max_{th}} p_d d(avg) = \int_{min_{th}}^{max_{th}} p'_d d(avg)$$

4.2. Why Use a Quadratic Function?

Given that N TCP flows equally share a link with

NLRED

- for each packet arrival:
 - calculate the average queue size avg
 - if $avg \leq min_{th}$
 - no packet drop
 - else if $min_{th} \leq avg \leq max_{th}$
 - calculate the packet drop probability using (2)
 - drop the packet with the calculated probability
 - else
 - drop the packet

Figure 9. Pseudocode of NLRED.

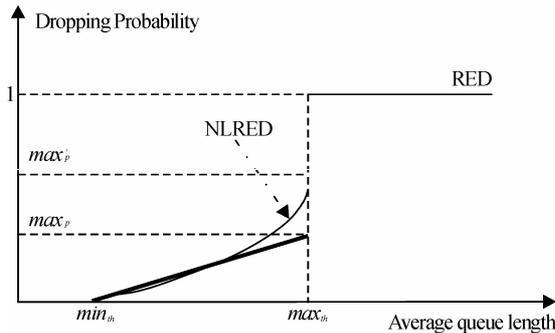


Figure 10. Dropping functions for NLRED and RED.

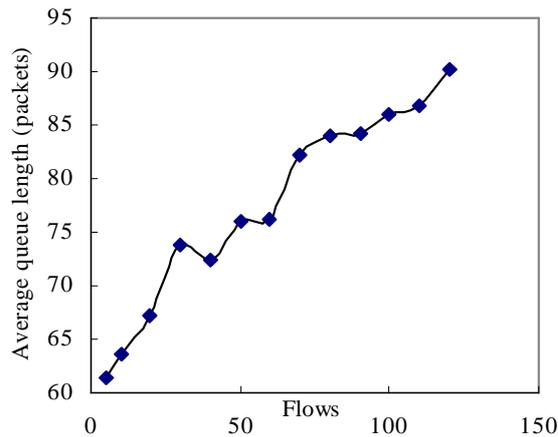


Figure 11. Average queue size vs. number of flows, with drop-tail router.

bandwidth L , and experience a random packet loss/drop probability p . It was shown that p and N has the following relationship.

$$p < \left(\frac{N \text{ MSS} * \alpha}{L \text{ RTT}} \right)^2$$

where α is a constant. This equation indicates that to effectively manage the flows (so as to fully utilize the available network bandwidth) the packet dropping probability should vary quadratically with the number of flows. However, finding the number of active flows N needs 1) per flow information, 2) extra storage space for storing extra state information, and 3) extra router processing overhead. Besides, the resulting flow number is nothing more than an estimation [13,16].

In (3), we have proposed to vary the packet dropping probability based on a quadratic function of average queue size. In [17], it is shown that the average queue size at a router is roughly directly proportional to the number of active TCP flows passing through it. This is further verified by the simulations results shown in Figure 11. The average queue size versus the number of flows is obtained by simulating the network in Figure 12 with drop-tail router mechanisms. (Other simulations using RED with different traffic load also show similar results.)

In fact, choosing a quadratic function is also intuitively

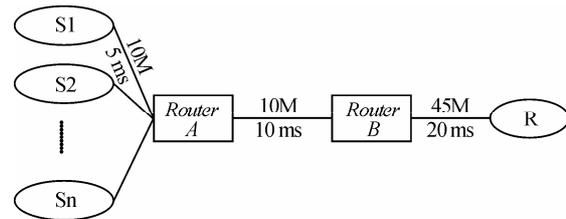


Figure 12. The network simulated.

appealing. From Figure 10, when the average queue size is slightly larger than min_{th} , the packet dropping probability is smaller than the corresponding RED. As such, the average queue size will not be forced to work around min_{th} as strongly as that in RED. Or, one can interpret this as follows. Under current traffic load, the signal for congestion is not strong enough to justify any severe measures to cut back queue size; so a gentler than RED packet dropping probability is desirable. While doing this, we naturally encourage the routers to operate over a range of queue sizes closer to min_{th} (instead of at a fixed target queue size). When avg approaches max_{th} , the congestion becomes more pronounced. The routers can thus take decisive actions to drop packets at a rate higher than RED. When avg is bigger than max_{th} , the routers drop any packets received. Although GRED shows superior performance than RED with an additional linear dropping function when avg is between max_{th} and $2max_{th}$, the design of NLRED does not adopt similar approach. Besides simplifying the algorithm, determined drop is more reasonable for NLRED than another slow changed dropping function (such as used in GRED), because higher than RED dropping probability has already been proven to be too gentle.

5. Simulation of Nonlinear RED

NLRED is implemented using ns-2 simulator [18]. We conduct the simulations based on the network in Figure 12, which consists of N senders and one sink, connected together via two routers A and B. The link between the two routers is the bottleneck. By varying N , we produce different levels of traffic load and thus different levels of congestion on the bottleneck link. The active queue management schemes under investigation are implemented at router A, which has a queue buffer size of 120 packets. Unless otherwise stated, we assume that all packets generated by the senders are 1000 bytes long. Extensive simulations based on this network using different TCP implementations (Tahoe, Reno, and New Reno), RTTs, and AQM schemes (with different parameter sets), are conducted, whereas only a representative subset of the results based on TCP Reno is reported below. Besides, we choose to compare NLRED with GRED [19] instead of RED, due to the superior performance of GRED over RED. We also compare NLRED with REM [18] as it is a representative scheme that steers a router to operate

around a fixed target queue size with excellent reported performance.

Experiment 1

Figures 13 to 16 show the results of a set of simulations with the number of long-lived TCP flows increasing from 5 to 120 and max_p varying from 0.02 to 0.5. The receiver's advertised window of each connection is set to be bigger than the bandwidth delay product. Each point of the simulation results is obtained from a single 200 seconds simulation while the statistics are collected in the second half of the simulation time (i.e. the second 100-second interval).

As explained earlier, in order to compare GRED and NLRED, the maximum packet dropping probability of NLRED is set as $max'_p = 1.5max_p$. As such, the simulation results/curves obtained using NLRED will be labelled by its equivalent max_p instead of max'_p . As an example, the line labelled with $max_p = 0.1$ in Figure 16 means the actual maximum packet dropping probability is $max'_p = 0.15$. Both GRED and NLRED use the same set of parameters, $\omega_q = 0.002$, $min_{th} = 10$, and $max_{th} = 30$.

Figures 13 and 14 show the bottleneck link throughput against the number of flows. Each curve in the figures represents the simulation results with a given max_p . Comparing the two figures, we can see that NLRED is less sensitive to the choice of max_p under different traffic loads (i.e. number of flows). Although the throughput of NLRED still changes with the load, for some max_p selections (e.g. $max_p = 0.05$ to 0.1, or $max'_p = 0.075$ to 0.15), NLRED is very successful in maintaining a high throughput regardless of the loading. This is mainly due to NLRED's nonlinear quadratic packet dropping function, which allows more packet bursts to pass when the average queue size is small, and drops more packets when the average queue size becomes large.

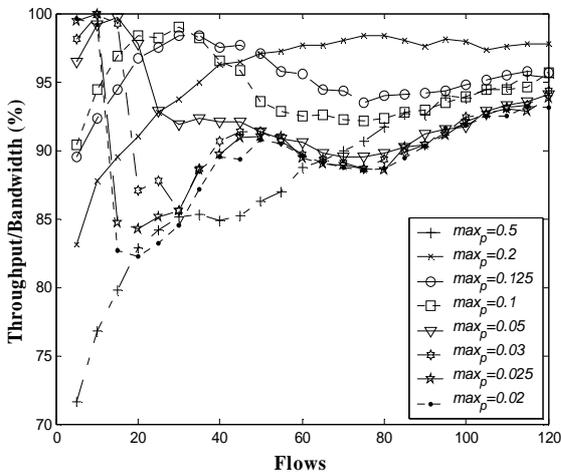


Figure 13. Throughput vs. number of flows using GRED.

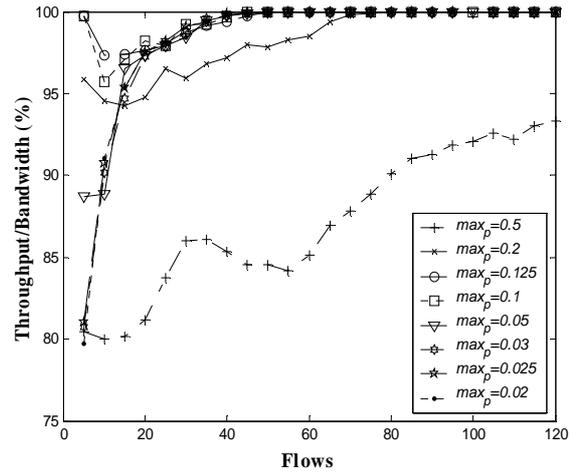


Figure 14. Throughput vs. number of flows using NLRED.

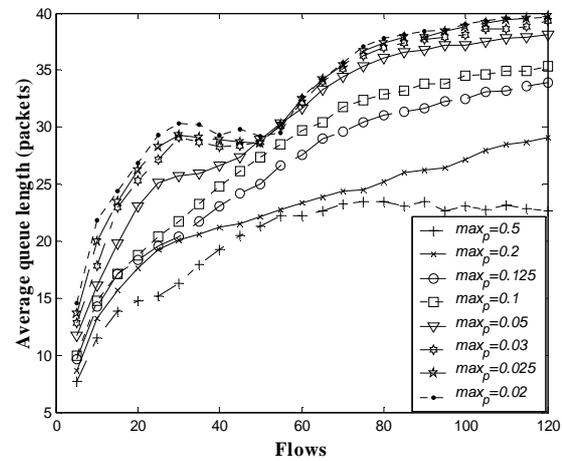


Figure 15. Average queue size vs. number of flows using GRED.

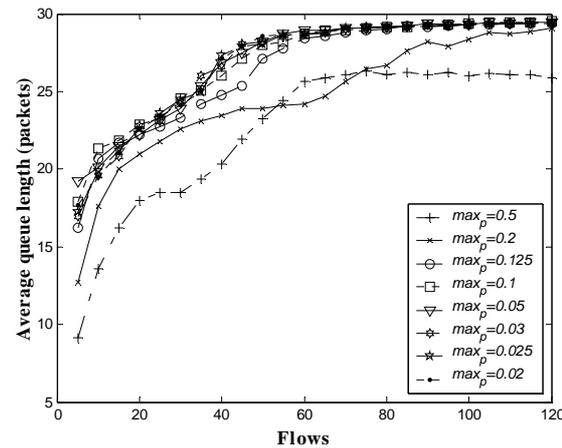


Figure 16. Average queue size vs. number of flows using NLRED.

Figures 15 and 16 show the change of the average queue size with the number of flows. Unlike GRED, we can see that NLRED allows the average queue size to grow at a faster rate when the number of flows is small. As the number of flows increases, NLRED tends to control the average queue size better (i.e. the queue size converges to a stable value faster) than GRED.

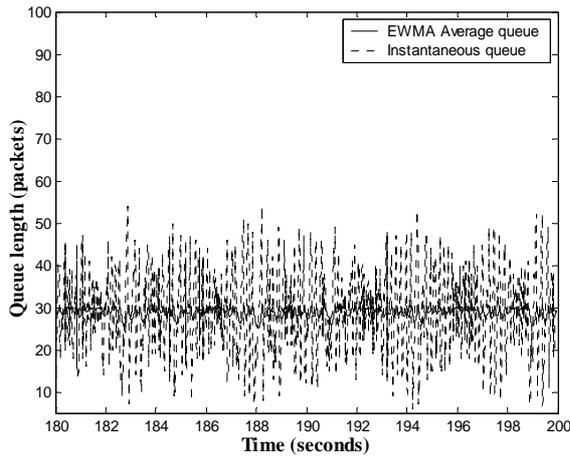


Figure 17. Change in queue occupancy when NLRED is used with $N=100$ flows.

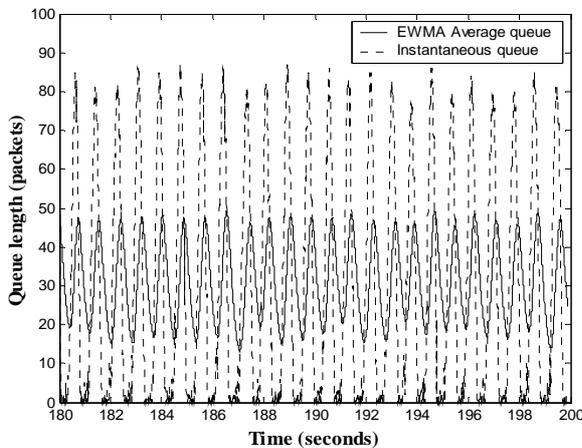


Figure 18. Change in queue occupancy when GRED is used with $N = 100$ flows, $max_p = 0.02$, $\omega_q = 0.002$, $min_{th} = 10$, $max_{th} = 30$.

To have a closer examination on the ability to control queue size, we show in Figures 17 and 18 the instantaneous and average queue sizes against time, with the number of flows $N=100$ and $max_p=0.02$. We can see that the oscillations in both instantaneous and average queue sizes are much more noticeable when GRED is used. With NLRED, the oscillations are effectively suppressed, again due to its nonlinear packet dropping function.

Experiment 2

We compare the performance of GRED, REM [20] and NLRED under different traffic loads. We set max_p of all the three AQM schemes to 0.1, $\omega_q = 0.002$, $min_{th} = 10$, and $max_{th} = 30$. The default parameters of REM in ns-2 are used, they are $\gamma = 0.001$, $\alpha = 0.1$, $\phi = 1.001$, and $b = 20$.

From Figure 19, we can see that NLRED has the highest overall throughput, whereas GRED is the lowest.

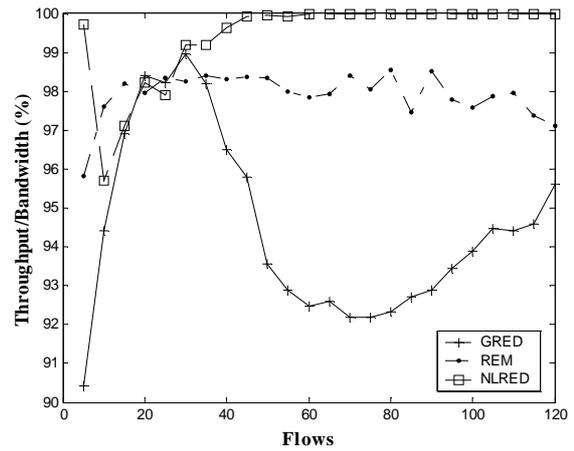


Figure 19. Throughput vs. flow number: GRED, REM and NLRED.

It is interesting to see a short concave phase when the traffic is changed from 10 flows to 40 flows. It is shown that the performance of NLRED is not very stable during this range, partly because of the sharp non-contiguous increase of dropping probability from max_p to 1 when avq grows over max_{th} . However, as soon as the number of flows is larger than 40, the throughput for NLRED quickly converges to the link bandwidth. Besides, during the concave range, the throughput of NLRED is still always higher than GRED. Figure 20 shows the corresponding average queue size of using GRED, REM, and NLRED. By steering the queue around a target length, REM suffers the low throughput when traffic load is extremely light (less than 5 flows) and extremely high. When $N > 60$, the throughput of REM is unstable and drops as N increases.

Since Misra, *et al.* [14] indicated that packet size affects the performance of AQM schemes, in this experiment (again based on Figure 12), we test and compare the packet size sensitivity of GRED, REM, and NLRED. REM is configured to work in byte mode because packet mode shows extremely poor performance. (We believe the error

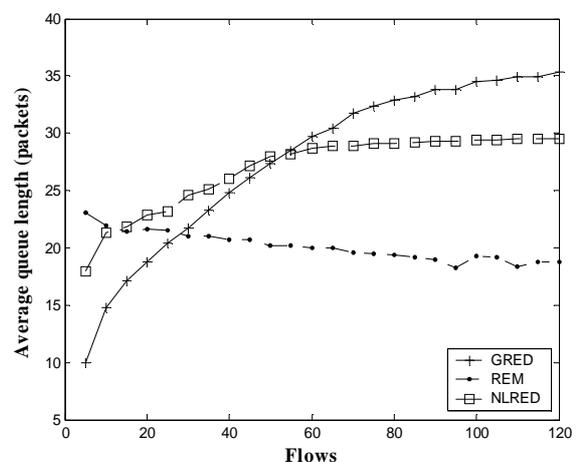


Figure 20. Average queue size vs. flow number: GRED, REM and NLRED.

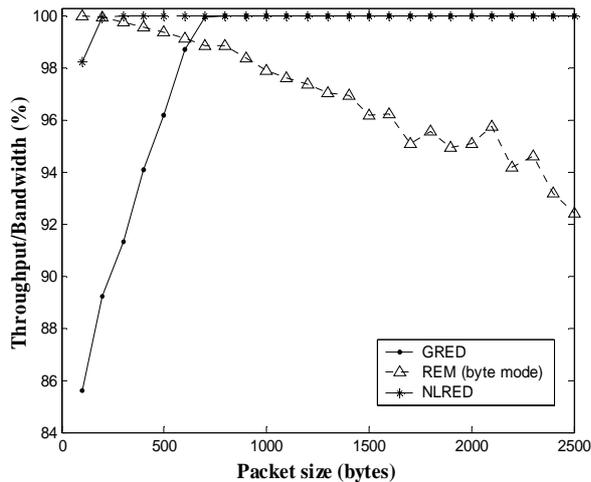


Figure 21. Throughput vs. packet size.

in arrival rate estimation is the reason for such poor performance with packet mode REM.) However, the queue length of byte mode REM cannot be directly compared with the results of other schemes, because it uses bytes as unit whereas others use packets. To solve the problem, we normalize the queue length of byte mode REM to use packets as unit. The conversion assumes all the packets are with the same size as the referenced packet size. We simulate 50 long-lived FTPs. For each AQM algorithm, we conduct a set of simulations with the packet size ranging from 100 bytes to 2500 bytes. Figure 21 shows the throughput against packet size. We can see that NLRED is least sensitive to the packet size and therefore is better than both GRED and REM.

6. Conclusions

We have proposed a mechanism improving the fairness of Internet routers, which called SFRED. The mechanism was developed with a MRUL in which states of up to N most recently used flows are stored. SFRED then identifies and punishes the fast and unresponsive fast flows. To improve short TCP transaction performance, SFRED also protects slow flows by allocating a small amount of buffer. Simulations show that the SFRED proposed has significantly improved the fairness of RED, with only limited resource usage. Different from the previous proposals the complexity of SFRED is proportional to the size of the list but not coupled with the queue buffer size or the number of active flows, so it is scalable and suitable for various routers. Moreover, in this paper, we also proposed a new active queue management scheme called Nonlinear RED (NLRED). NLRED is the same as the original RED except that the linear packet dropping probability function is replaced by a nonlinear quadratic function. While inheriting the simplicity of RED, NLRED was shown to outperform RED as well as REM and some of its variants. In particular, NLRED is less sensitive to parameter settings, has a more

predictable average queue size, and can achieve a higher throughput. We credit the above performance gain to the idea of encouraging the router to operate over a range of queue sizes according to traffic load instead of at a fixed one. This is realized in NLRED by using a gentle packet dropping probability at the onset of the congestion, and a much more aggressive dropping probability when the congestion becomes more pronounced.

7. Acknowledgement

This work is jointly supported by National Science Foundation of China under the project number 60773203, 60602066 and 60872010, and grant from Guangdong Natural Science Foundation under project number 5010494. The work has also got support from Foundation of Shenzhen City under project number QK200601, open research fund of National Mobile Communications Research Laboratory, Southeast University under project number W200815, and National High Technology Research and Development Program of China under project number 2007AA01Z218.

8. References

- [1] F. Ren, C. Lin, and X. Huang, "TCC: A two-category classifier for AQM routers supporting TCP flows," *IEEE Communications Letters*, Vol. 9, pp. 471-473, 2005.
- [2] R. T. Morris, "Scalable TCP congestion control," Ph. D thesis, Harvard University, 1999.
- [3] C. V. Hollot, V. Misra, D. Towsley, *et al.*, "Analysis and design of controllers for AQM routers supporting TCP flows," *IEEE Transactions on Automatic Control*, Vol. 47, pp. 945-959, 2002.
- [4] M. Nabeshima and K. Yata, "Performance improvement of active queue management with per-flow scheduling," *IEEE Proceedings-Communications*, Vol. 152, pp. 797-803, 2005.
- [5] Smitha and A. L. N. Reddy, "LRU-RED: An active queue management scheme to contain high bandwidth flows at congested routers," in *Proceedings of IEEE Global Telecommunications Conference 2001 (GLOBECOM '01)*, San Antonio, TX, USA, Vol. 1, pp. 2311-2315, November 25-29, 2001.
- [6] C. Brandauer, G. Iannaccone, C. Diot, *et al.*, "Comparison of tail drop and active queue management performance for bulk-data and web-like Internet traffic," in *Proceedings of Sixth IEEE Symposium on Computers and Communications 2001, (ISCC '01)*, Hammamet, Tunisia, pp. 122-129, July 3-5, 2001.
- [7] T. J. Ott, T. V. Lakshman, and L. H. Wong, "SRED: Stabilized RED," in *Proceedings of Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '99)*, New York, USA, Vol. 3, pp. 1346-1355, March 21-25, 1999.
- [8] F. M. Anjum and L. Tassiulas, "Fair bandwidth sharing among adaptive and non-adaptive flows in the Internet," in *Proceedings of Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM*

- '99), New York, NY, USA, Vol. 3, pp. 1412–1420, March 21–25, 1999.
- [9] D. Lin and R. Morris, “Dynamics of random early detection,” in Proceedings of ACM SIGCOMM '97, Cannes, France, pp. 127–137, September 14–18, 1997.
- [10] R. Mahajan, S. Floyd, and D. Wetherall, “Controlling high-bandwidth flows at the congested router,” in Proceedings of Ninth International Conference on Network Protocols, pp. 192–201, 2001.
- [11] M. Christiansen, K. Jeffay, D. Ott, *et al.*, “Tuning RED for web traffic,” in Proceedings of ACM SIGCOMM 2000, Stockholm, Sweden, pp. 139–150, August 28–September 1, 2000.
- [12] B. Suter, T. V. Lakshman, D. Stiliadis, *et al.*, “Design considerations for supporting TCP with per-flow queueing,” in Proceedings of Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '98), San Francisco, CA, USA, pp. 299–306, March 31–April 2, 1998.
- [13] L. Zhang, S. Shenker, and D. D. Clark, “Observations on the dynamics of a congestion control algorithm: the effects of two-way traffic,” in Proceedings of ACM SIGCOMM '91, the Conference on Communications Architecture & Protocols, Vol. 1, pp. 133–147, Zurich, Switzerland, September 03–06, 1991.
- [14] S. Floyd, J. Mahdavi, M. Mathis, *et al.*, “An extension to the selective acknowledgement (SACK) option for TCP,” in RFC 2883, 2000.
- [15] K. Xu and N. Ansari, “Stability and fairness of rate estimation based AIAD congestion control in TCP,” IEEE Communications Letters, Vol. 9, pp. 378–380, 2005.
- [16] T. J. Ott, T. V. Lakshman, and L. H. Wong, “SRED: Stabilized RED,” in Proceedings of Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies, (INFOCOM'99), New York, NY, USA, Vol. 3, pp. 1346–1355, March 21–25, 1999.
- [17] B. Braden, D. Clark, J. Crowcroft, *et al.*, “Recommendations on queue management and congestion avoidance in the Internet,” in RFC2309, April 1998.
- [18] V. Misra, W. B. Gong, and D. Towsley, “Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED,” ACM SIGCOMM Computer Communication Review, Vol. 30, pp. 151–160, 2000.
- [19] B. Zheng and M. Atiquzzaman, “DSRED: An active queue management scheme for next generation networks,” in Proceedings of 25th Annual IEEE Conference on Local Computer Networks (LCN'00), Tampa, FL, USA, Vol. 1, pp. 242–251, November 8–10, 2000.
- [20] S. Athuraliya, S. H. Low, V. H. Li, *et al.*, “REM: Active queue management,” IEEE Network, Vol. 15, pp. 48–53, 2001.

A Genetic Based Fuzzy Q-Learning Flow Controller for High-Speed Networks

Xin LI, Yuanwei JING, Nan JIANG, Siying ZHANG

College of Information Science and Engineering, Northeastern University, Shenyang, China

Email: lixin820106@126.com

Received August 15, 2008; revised December 3, 2008; accepted December 30, 2008

Abstract

For the congestion problems in high-speed networks, a genetic based fuzzy Q-learning flow controller is proposed. Because of the uncertainties and highly time-varying, it is not easy to accurately obtain the complete information for high-speed networks. In this case, the Q-learning, which is independent of mathematic model, and prior-knowledge, has good performance. The fuzzy inference is introduced in order to facilitate generalization in large state space, and the genetic operators are used to obtain the consequent parts of fuzzy rules. Simulation results show that the proposed controller can learn to take the best action to regulate source flow with the features of high throughput and low packet loss ratio, and can avoid the occurrence of congestion effectively.

Keywords: High-Speed Network, Flow Control, Fuzzy Q-learning, Genetic Operator

1. Introduction

The growing interest on congestion problems in high-speed networks arise from the control of sending rates of traffic sources. Congestion problems result from a mismatch of offered load and available link bandwidth between network nodes. Such problems can cause high packet loss ratio (PLR) and long delays, and can even break down the entire network system because of the congestion collapse. Therefore, high-speed networks must have an applicable flow control scheme not only to guarantee the quality of service (QoS) for the existing links but also to achieve high system utilization.

The flow control of high-speed networks is difficult owing to the uncertainties and highly time-varying of different traffic patterns. The flow control mainly checks the availability of bandwidth and buffer space necessary to guarantee the requested QoS. A major problem here is the lack of information related to the characteristics of source flow. Devising a mathematical model for source flow is the fundamental issue. However, it has been revealed to be a very difficult task, especially for broadband sources. In order to overcome the above-mentioned difficulties, the flow control scheme with learning capability has been employed in flow control of high-speed network [1,2]. But the priori-knowledge of system to train the parameters in the controller is hard to achieve for high-speed networks.

In this case, the reinforcement learning (RL) shows its particular superiority, which just needs very simple information such as estimable and critical information, "right" or "wrong" [3]. RL is independent of mathematic model and priori-knowledge of system. It obtains the knowledge through trial-and-error and interaction with environment to improve its behavior policy. So it has the ability of self-learning. Because of the advantages above, RL has been played a very important role in the flow control in high-speed networks [4-7]. The Q-learning algorithm of RL is easy for application and has a firm foundation in the theory. In [8], a Metropolis criterion based Q-learning controller is proposed to solve the problem of flow control in high-speed networks.

In Q-learning based control, the learning agent should visit each state in a reasonable time. But in high-speed networks, the state space is large, so the usual approach of storing the Q-values in a look-up table is impractical. In [8], a state space partitioning method is introduced to reduce the number of state variables, but it can not solve this problem ultimately. In this paper, we adopt fuzzy Q-learning (FQL), which is an adaptation of Q-learning for fuzzy inference system (FIS), to facilitate generalization the state space. In FQL, both the actions and Q-values are inferred from fuzzy rules, and it can map a state-action pair to a Q-value in a continuous state space. Furthermore, we employ the changes of q values as the fitness values, and use the genetic operators

to obtain the consequent parts of fuzzy rules.

In this paper, a genetic based fuzzy Q-learning flow controller (GFQC) for high-speed networks is proposed. The proposed controller can behave optimally without the explicit knowledge of the network environment, only relying on the interaction with the unknown environment and provide the best action for a given state. By means of learning process, the proposed controller adjusts the source sending rate to the optimal value to reduce the average length of queue in the buffer. Simulation results show that the proposed controller can avoid the occurrence of congestion effectively with the features of high throughput, low PLR, low end-to-end delay, and high utilization.

2. Theoretical Framework

2.1. Architecture of the Proposed Flow Controller

The architecture of the proposed GFQC is shown in Figure 1. In high-speed networks, GFQC in bottleneck node acts as a flow control agent with flow control ability. The inputs of GFQC are state variables \mathbf{S} in high-speed networks composed of the current queue length q_L , the current change rate of queue length \dot{q}_L , and the current change rate of source sending rate \dot{u} . The output of GFQC is the feedback signal a to the traffic sources, which is the ratio of the sending rate. It determines the sending rate u of traffic sources. The learning agent and the network environment interact continually in the learning process. At the beginning of each time step of learning, the controller senses the states for the network and gets the reward signal. Then it selects an action to make decision on which ratio the sources should use to determine the source sending rate. The determined sending rate can reduce the PLR and increase the link utilization. After the sources take the determined rate to send the traffic, the network changes its state and gives a new reward to the controller. Then the next step of learning begins.

2.2. Fuzzy Q-Learning Flow Controller

Q-learning learns utility values (Q-values) of state and action pairs. During the learning process, learning agent uses its experience to improve its estimate by blending new information into its prior experience.

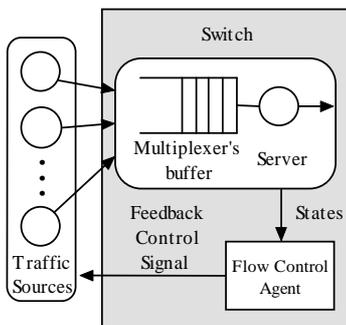


Figure 1. Architecture of the proposed GFQC.

In general form, Q-learning algorithm is defined by a tuple $\langle \mathbf{S}, \mathbf{A}, r, p \rangle$, where \mathbf{S} is the set of discrete state space of high-speed networks; \mathbf{A} is the discrete action space, which is the feedback signal to traffic sources; $r: \mathbf{S} \times \mathbf{A} \rightarrow \mathbf{R}$ is the reward of the agent; $p: \mathbf{S} \times \mathbf{A} \rightarrow \Delta(\mathbf{S})$ is the transition probability map, where $\Delta(\mathbf{S}) \in [0,1]$ is the set of probability distributions over state space \mathbf{S} .

Q-learning provides us with a simple updating procedure, in which the learning agent starts with arbitrary initial values of $Q(s, a)$ for all $s \in \mathbf{S}$, $a \in \mathbf{A}$, and updates the Q-values as

$$Q_{t+1}(s_t, a_t) = (1 - \alpha)Q_t(s_t, a_t) + \alpha \left[r_t + \beta \max_a Q_t(s_{t+1}, a) \right] \quad (1)$$

where α is the learning rate and $\beta \in [0,1)$ is the discount rate [9].

It is vital to choose an appropriate r in Q-learning [10]. In this paper, based on the requirement and experience of the buffer, r is defined as

$$r = \begin{cases} 0 & q_L \geq 1.1q_{LT} \text{ or } q_L \leq 0.9q_{LT} \\ \frac{1.1q_{LT} - q_L}{0.1q_{LT}} & q_{LT} < q_L < 1.1q_{LT} \\ \frac{q_L - 0.9q_{LT}}{0.1q_{LT}} & 0.9q_{LT} < q_L < q_{LT} \\ 1 & q_L = q_{LT} \end{cases} \quad (2)$$

where q_{LT} is the set value of queue length in the buffer. Refer to (2), if the value of q_L is less than $0.9q_{LT}$ or more than $1.1q_{LT}$, $r = 0$, the control result should be considered bad. If the value of q_L is equal to q_L , $r = 1$, it can be thought that the control result is good. Otherwise, r is in the range (0,1), the larger r is, the better control affects.

In Q-learning based control, the usual approach of storing the Q-values in a look-up table is impractical in the case of a large state space in high-speed networks. Furthermore, it is unlikely to visit each state in a reasonable time. Fuzzy Q-learning is an adaptation of Q-learning for fuzzy inference system, where both the actions and Q-values are inferred from fuzzy rules [11].

In high-speed networks, FIS relies on three parameters $\mathbf{S}(q_L, \dot{q}_L, \dot{u})$ to generate a selected action a . For an input state $\mathbf{s} = \{q_L, \dot{q}_L, \dot{u}\}$, we find the activate value of each rule $R^i: \omega_i(\mathbf{s})$. Each rule has m possible discrete control actions $\mathbf{A} = \{a_1, a_2, \dots, a_m\}$, and a parameter called q value associated with each control action. The state associates to each action in R^i , a quality with respect to the task. In FQL, one builds an FIS with competing actions for each rule $i \in N$ designated as

$$R^i: \text{ If } q_L \text{ is } L_1^i \text{ and } \dot{q}_L \text{ is } L_2^i \text{ and } \dot{u} \text{ is } L_3^i \\ \text{ then } a \text{ is } a_j^i \text{ with } q_j^i \quad (3)$$

where q_j^i is the j th q value in a rule i and

L_s^i =linguistic term (fuzzy label) of input variable s_s in rule R^i , its membership function is denoted by $\mu_{L_s^i}$. The q values in (3) are calculated according to total accumulated rewards and rules' activate values.

The functional blocks of FIS are a fuzzifier, a defuzzifier, and an inference engine containing a fuzzy rule base [12]. The fuzzifier performs the function of fuzzification that translates the value of each input linguistic variable into fuzzy linguistic terms. These fuzzy linguistic terms are defined in a term set $F(\mathbf{S})$ and are characterized by a set of membership function $\mu(\mathbf{S})$. The defuzzifier describes an output linguistic variable of selected action a by a term set $F(a)$, characterized by a set of membership functions $\mu(a)$, and adopts a defuzzification strategy to convert the linguistic terms of $F(a)$ into a nonfuzzy value representing selected action a .

The term set should be determined at an approximate level of granularity to describe the values of linguistic variables. The term set for q_L is defined as $F(q_L) = \{Low(L), Medium(M), High(H)\}$, which is used to describe the degree of queue length as "Low", "Medium", or "High". The term set for \dot{q}_L is defined as $F(\dot{q}_L) = \{Decrease(D), Increase(I)\}$, which describes the change rate of queue length as "Decrease" or "Increase". The term set for \dot{u} is defined as $F(\dot{u}) = \{Negative(N), Positive(P)\}$, which describes the change rate of source sending rate as "Negative" or "Positive". On the other hand, in order to provide a precise graded feedback signal in various states, the term for feedback signal is defined as $F(a) = \{Higher(HE), High(H), Normal(N), Low(L), Lower(LE)\}$. The membership functions (MFs) are shown in Figure 2.

In each rule R^i , the learning agent (controller) can choose one action a_j^i from the action set $\mathbf{A} = \{a_1, a_2, \dots, a_m\}$. The inferred global continuous action a_t at state \mathbf{s} is calculated as

$$a_t = \frac{\sum_{i=1}^N \omega_i(\mathbf{s}_t) a_j^i}{\sum_{i=1}^N \omega_i(\mathbf{s}_t)} \quad (4)$$

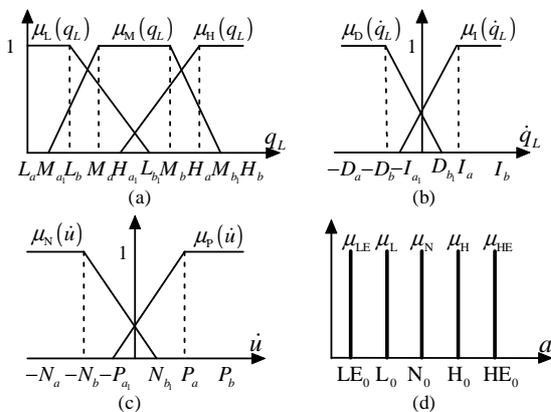


Figure 2. MFs of term set (a) $F(q_L)$, (b) $F(\dot{q}_L)$, (c) $F(\dot{u})$, and (d) $F(a)$.

where a_j^i is the action selected in rule R^i using a Metropolis criterion based exploration/exploitation policy in [8].

Following fuzzy inference, the Q-value for the inferred action a_t is calculated as

$$Q(\mathbf{s}_t, a_t) = \frac{\sum_{i=1}^N \omega_i(\mathbf{s}_t) q_j^i}{\sum_{i=1}^N \omega_i(\mathbf{s}_t)} \quad (5)$$

Under action $a(\mathbf{s}_t)$, the system undergoes transition $\mathbf{s}_t \xrightarrow{r} \mathbf{s}_{t+1}$ where r is the reward received by the controller. This information is used to calculate temporal difference (TD) approximation error as

$$\Delta Q = r + \beta \cdot \max_a Q(\mathbf{s}_{t+1}, a) - Q(\mathbf{s}_t, a_t) \quad (6)$$

The change of q value can be found by

$$\Delta q_j^i = \Delta Q \cdot \frac{\omega_i(\mathbf{s}_t)}{\sum_{i=1}^N \omega_i(\mathbf{s}_t)} \quad (7)$$

We can rewrite the learning rule (1) of q parameter values as

$$q_j^i \leftarrow q_j^i + \alpha \cdot \Delta q_j^i \quad (8)$$

2.3. The Genetic Operator Based Flow Controller

In this section we develop the fuzzy Q-learning controller by genetic operators. The consequent parts of fuzzy rules need to compete for survival within a niche. In this case, each rule in FIS maintains a q value, but it is no longer an estimation of accumulated rewards. The max operator in standard fuzzy Q-learning is not used since the rules that have maximum q value no longer represent rules with the best rewards. Because it is not suitable to use the q values as the fitness values in the learning, we employ their changes Δq as the fitness values. In this paper the fuzzy rule in (3) can be rewritten as follows:

$$R^i : \text{If } q_L \text{ is } L_1^i \text{ and } \dot{q}_L \text{ is } L_2^i \text{ and } \dot{u} \text{ is } L_3^i \text{ then } a \text{ is } a_j^i \text{ with } q_j^i \text{ and } \Delta q_j^i \quad (9)$$

The fitness value for a rule is an inverse measure of Δq . By using the fitness value calculation in [13], a predicted rule accuracy κ at time step t is defined as

$$\kappa_t = \begin{cases} \eta e^{-(\Delta q_t - \Delta q_0)} & \Delta q_t > \Delta q_0 \\ \eta & \text{otherwise} \end{cases} \quad (10)$$

The accuracy falls off exponentially for $\Delta q_t > \Delta q_0$. Δq_0 is an initial value. The predicted accuracy in (10) can be used to adjust rule's fitness value f_t using the standard Widrow-Hoff delta rule

$$f_t = f_t + \chi(\kappa_t - f_t) \quad (11)$$

where χ is an adjust rate of fitness values.

The niche genetic operators can prevent the population from the premature convergence or the genetic drift resulting from the selection operator. The niche genetic operators maintain population diversity and promote the formation of sub-population in the neighbourhood of local optimal solutions. In fuzzy Q-learning, the fitness sharing is implicitly implemented by assigning fitness values to the activated rules based on their contributions. The fuzzy rule antecedent constitutes an evolving niche or sub-population where the fuzzy rules with the same antecedent share similar environment states. The rule consequences or actions need to compete for survival within a niche, while the rules from different niches co-operate to generate the output.

In the definition of a fuzzy rule in (9), a fuzzy rule can be defined as a sub-population and the rule actions are encoded as individuals in sub-population. If there are N rules in fuzzy Q-learning, there will be N sub-population. As shown in Figure 3, in each learning step, the reward from the environment is apportioned to the rules that are activated in the previous step. The rule's fitness values are accordingly updated in the form of (11). There is a winner action in each sub-population and the winner actions from all sub-population are formed the consequent parts of fuzzy rules. The selection for the winners in sub-population is implemented by the niche genetic operator. The niche genetic operator uses two operators to select the actions:

Reproduce operator: individuals in each sub-population are selected as winners in terms of their fitness values. The roulette wheel selection is used.

Mutation operator: the mutation is taken for each sub-population with a mutation probability. The operator chooses an individual from sub-population randomly to replace a winner in the sub-population.

In the learning process, the network environment provides current states and rewards to the learning agent. The learning agent produces actions to perform in the network. The learning agent includes a performance component, a reinforcement component, and a discovery component.

The performance component reads states from network environment, calculates activation degrees of fuzzy rules, and generates an action. The action is then

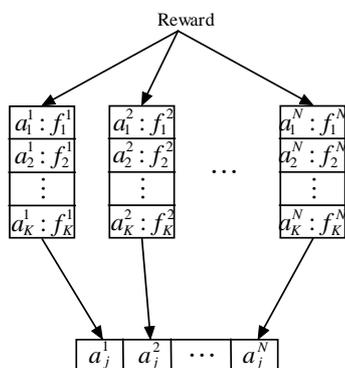


Figure 3. Learning mechanism of genetic operator.

executed by the traffic sources. The network moves into next state and receives evaluating reward from the network environment for its action.

The discovery component plays an action selection role. Two genetic operators are used to implement the selection. Finally, a set of rule actions is selected for the performance component.

The reinforcement component serves to assign the reward to the individual rules that are activated by current state.

3. Simulation and Comparison

The simulation model of high-speed network, as shown in Figure 4, is composed of two switches, Sw1 with a control agent and Sw2 with no controller are cascaded. The constant output link L is 80Mbps. The sending rates of the sources are regulated by the flow controllers individually.

In the simulation, we assume that all packets are with a fixed length of 1000bytes, and adopt a finite buffer length of 20packets in the node. On the other hand, the offered loading of the simulation varies between 0.6 and 1.2 corresponding to the systems' dynamics; therefore, higher loading results in heavier traffic and vice versa. For the link of 80Mbps, the theoretical throughput is 62.5K packets.

From the knowledge of evaluating system performance, the parameters of the membership functions for input linguistic variables in FIS are selected as follows. For $\mu_L(q_L)$, $\mu_M(q_L)$, and $\mu_H(q_L)$, $L_a = 0$, $L_b = 6$, $L_{b_1} = 10$, $M_{a_1} = 2$, $M_a = 8$, $M_b = 12$, $M_{b_1} = 20$, $H_{a_1} = 9$, $H_a = 14$, $H_b = 20$, and $H_{b_1} = 20$; for $\mu_D(\dot{q}_L)$ and $\mu_I(\dot{q}_L)$, $D_a = 4$, $D_b = D_{b_1} = 2$, $I_{a_1} = I_a = 2$, and $I_b = 4$; for $\mu_N(\dot{u})$ and $\mu_P(\dot{u})$, $N_a = 0.8$, $N_b = 0.4$, $N_{b_1} = 0.2$, $P_{a_1} = 0.2$, $P_a = 0.4$, and $P_b = 0.8$. Also, the parameters of the membership functions for output linguistic variables are given by $LE_0 = 0.2$, $L_0 = 0.4$, $N_0 = 0.6$, $H_0 = 0.8$, and $HE_0 = 1$.

The fuzzy rule base is an action knowledge base, characterized by a set of linguistic statements in the form of "if-then" rules that describe the fuzzy logic relationship between the input variables and selected action. After the leaning process, the inference rules in fuzzy rule base under various system states are shown in Table I. According to fuzzy set theory, the fuzzy rule base forms a fuzzy set with dimensions $3 \times 2 \times 2 = 12$. For example, rule

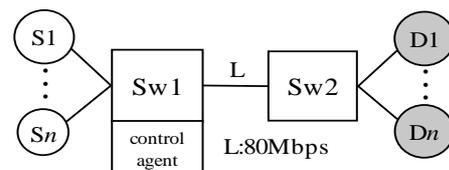


Figure 4. The simulation model of network with two switches.

1 can be linguistically started as “if the queue length is low, the queue length change rate is decreased, and the sending rate change rate is negative, then the feedback signal is Higher.”

In the simulation, four schemes of flow control agent, AIMD, standard reinforcement learning-based neural flow controller (RLNC), Metropolis criterion based Q-learning flow controller (MQLC), and the proposed GFQC are implemented individually in high-speed network. The first scheme AIMD increases its sending rate by a fixed increment (0.11) if the queue length is less than the predefined threshold; otherwise the sending rate is decreased by a multiple of 0.8 of the previous sending rate to avoid congestion [14]. Finally, for the other schemes, the sending rate is controlled by the feedback control signal a_t periodically. The controlled sending rate is defined by the equation

$$u_t = a_t FL \tag{12}$$

where $a_t \in [0.2, 1.0]$ is the feedback signal by the flow controller, F is a relative value in the ratio of source offered load to the available output bit rate, L denotes the outgoing rate of link, and $u_t \in [0.2 \cdot FL, FL]$ is the controlled sending rate at sample time t .

In simulation four measures, throughput, PLR, buffer utilization, and packets’ mean delay, are used as the performance indices. The throughput is the amount of received packets at specified nodes (switches) without retransmission. The status of the input multiplexer’s buffer in node reflects the degree of congestion resulting in possible packet losses. For simplicity, packets’ mean delay only takes into consideration the processing time at node plus the time needed to transmit packets.

The performance comparison of throughput, PLR, buffer utilization, and mean delay controlled by four different kinds of agents individually are shown in Figure 5-8. The throughput for AIMD method decrease seriously at loading of 0.9. Conversely, the GFQC proposed remain a higher throughput even though the offered loading is over 1.0, and can decrease the PLR enormously with high throughput and low mean delay. The GFQC has a better performance over RLNC and MQLC in PLR, buffer utilization, and mean delay. It demonstrates once again that GFQC possesses the ability to predict the network behavior in advance.

Table 1. Rule table of FIS.

Rule	q_L	\dot{q}_L	\dot{u}	a	Rule	q_L	\dot{q}_L	\dot{u}	a
1	L	D	N	HE	7	M	I	N	N
2	L	D	P	H	8	M	I	P	LE
3	L	I	N	N	9	H	D	N	L
4	L	I	P	N	10	H	D	P	LE
5	M	D	N	H	11	H	I	N	L
6	M	D	P	L	12	H	I	P	LE

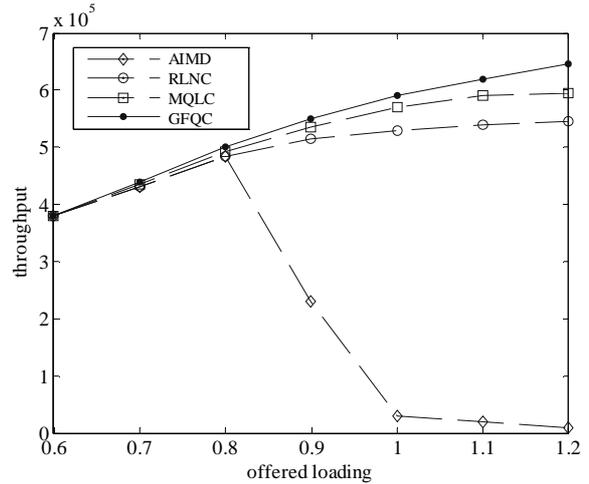


Figure 5. Throughput versus various offered loading.

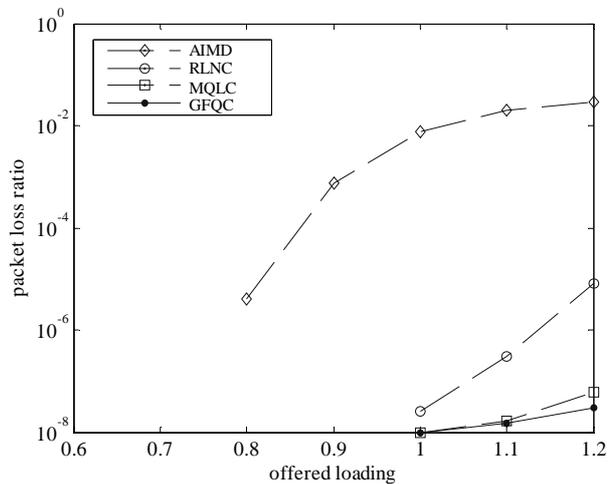


Figure 6. PLR versus various offered loading.

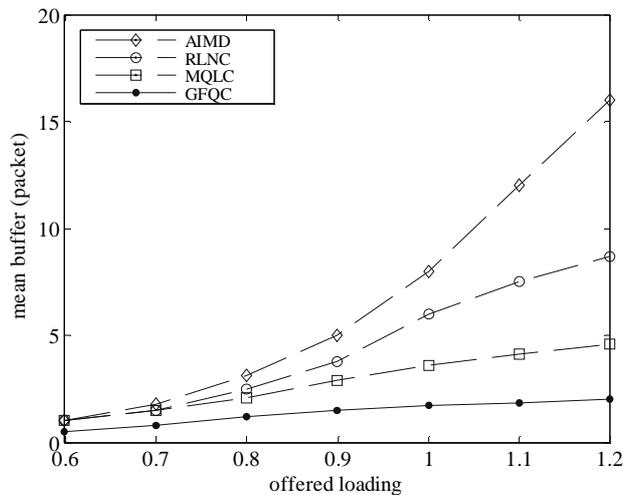


Figure 7. Mean buffer versus various offered loading.

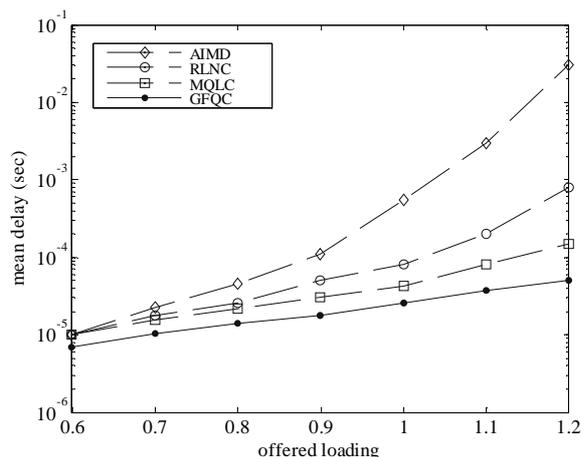


Figure 8. Mean delay versus various offered loading.

4. Conclusions

In the flow control of high-speed networks, the reactive scheme AIMD could not accurately respond to a time-varying environment due to the lack of prediction capability. The fuzzy Q-learning flow controller has good performance when the state space of high-speed network is large and continuous. The genetic operator is introduced to obtain the consequent parts of fuzzy rules. Through a proper training process, the proposed GFQC can respond to the networks' dynamics and learn empirically without prior information on the environmental dynamics. The sending rate of traffic sources can be determined by the well-trained flow control agent. Simulation results have shown that the proposed controller can increase the utilization of the buffer and decrease the PLR simultaneously. Therefore, the GFQC proposed not only guarantees low PLR for the existing links, but also achieves high system utilization.

5. References

- [1] R. G. Cheng, C. J. Chang, and L. F. Lin, "A QoS-provisioning neural fuzzy connection admission controller for multimedia high-speed networks," *IEEE/ACM Transactions on Networking*, Vol. 7, No. 1, pp. 111–121, 1999.
- [2] M. Lestas, A. Pitsillides, P. Ioannou, and G. Hadjipollas, "Adaptive congestion protocol: A congestion control protocol with learning capability," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, Vol. 51, No. 13, pp. 3773–3798, September 2007.
- [3] R. S. Sutton and A. G. Barto, "Reinforcement learning an introduction," Cambridge, MA: MIT Press, 1998.
- [4] A. Chatovich, S. Okug, and G. Dunder, "Hierarchical neuro-fuzzy call admission controller for ATM networks," *Computer Communications*, Vol. 24, No. 11, pp. 1031–1044, June 2001.
- [5] M. C. Hsiao, S. W. Tan, K. S. Hwang, and C. S. Wu, "A reinforcement learning approach to congestion control of high-speed multimedia networks," *Cybernetics and Systems*, Vol. 36, No. 2, pp. 181–202, January 2005.
- [6] K. S. Hwang, S. W. Tan, M. C. Hsiao, and C. S. Wu, "Cooperative multiagent congestion control for high-speed networks," *IEEE Transactions on System, Man, and Cybernetics-Part B: Cybernetics*, Vol. 35, No. 2, pp. 255–268, April 2005.
- [7] X. Li, X. J. Shen, Y. W. Jing, and S. Y. Zhang, "Simulated annealing-reinforcement learning algorithm for ABR traffic control of ATM networks," in *Proceedings of the 46th IEEE Conference on Decision and Control*, New Orleans, LA, USA, pp. 5716–5721, December 2007.
- [8] X. Li, Y. W. Jing, G. M. Dimirovski, and S. Y. Zhang, "Metropolis criterion based Q-learning flow control for high-speed networks," in *17th International Federation of Automatic Control (IFAC) World Congress*, Seoul, Korea, pp. 11995–12000, July 2008.
- [9] C. J. C. H. Watkins, and P. Dayan, "Q-learning," *Machine Learning*, Vol. 8, No. 3, pp. 279–292, May 1992.
- [10] M. L. Littman, "Value-function reinforcement learning in Markov games," *Journal of Cognitive System Research*, Vol. 2, No.1, pp. 55–66, 2001.
- [11] D. B. Gu, and E. F. Yang, "A policy gradient reinforcement learning algorithm with fuzzy function approximation," in *Proceedings of the 2004 IEEE International Conference on Robotics and Biomimetics*, Shenyang, China, pp. 934–940, August 2004.
- [12] Y. Zhou, M. J. Er, and Y. Wen, "A hybrid approach for automatic generation of fuzzy inference systems without supervised learning," in *Proceedings of the 2007 American Control Conference*, New York City, USA, pp. 3371–3376, July 2007.
- [13] S. W. Wilson, "Classifier fitness based on accuracy," *Evolutionary Computation*, Vol. 3, No. 2, pp. 145–179, 1994.
- [14] P. Gevros, J. Crowcoft, P. Kirstein, and S. Bhatti, "Congestion control mechanisms and the best effort service model," *IEEE Network*, Vol. 15, No. 3, pp. 16–26, May–June 2001.



International Journal of **Communications, Network and System Sciences (IJCNS)**

ISSN 1913-3715 (Print) ISSN 1913-3723 (Online)

<http://www.scirp.org/journal/ijcns/>

IJCNS is an international refereed journal dedicated to the latest advancement of communications and network technologies. The goal of this journal is to keep a record of the state-of-the-art research and promote the research work in these fast moving areas.

Editor-in-Chief

Prof. Tom Hou

Department of Electrical and Computer Engineering, Virginia Tech., USA

Subject Coverage

This journal invites original research and review papers that address the following issues in wireless communications and networks. Topics of interest include, but are not limited to:

MIMO and OFDM technologies

UWB technologies

Wave propagation and antenna design

Signal processing and channel modeling

Coding, detection and modulation

3G and 4G technologies

Sensor networks

Ad Hoc and mesh networks

Network protocol, QoS and congestion control

Efficient MAC and resource management protocols

Simulation and optimization tools

Network security

We are also interested in:

- Short reports—Discussion corner of the journal :

2-5 page papers where an author can either present an idea with theoretical background but has not yet completed the research needed for a complete paper or preliminary data.

- Book reviews—Comments and critiques.

Notes for Intending Authors

Submitted papers should not have been previously published nor be currently under consideration for publication elsewhere. Paper submission will be handled electronically through the website. All papers are refereed through a peer review process. For more details about the submissions, please access the website.

Website and E-Mail

<http://www.scirp.org/journal/ijcns>

ijcns@scirp.org

TABLE OF CONTENTS

Volume 2

February 2009

Game Theoretic Analysis of Joint Rate and Power Allocation in Cognitive Radio Networks D. LI, X. H. DAI, H. ZHANG.....	1
Synchronous Dynamic Adjusting: An Anti-Collision Algorithm for an RF-UCard System J. C. CAO, L. SHU, Z. D. LU.....	8
Performance Analysis of Sub-Rating for Handoff Calls in HCN X. L. WU, M. HE, F. WANG, J. ZHENG, E. REGENTOVA, G. S. HAO.....	21
A Priority Queuing Model for HCF Controlled Channel Access (HCCA) in Wireless LANs R. GHAZIZADEH, P. Z. FAN, Y. PAN.....	30
Gateways Placement in Backbone Wireless Mesh Networks M. TANG.....	44
Parallel-Transmission: A New Usage of Multi-Radio Diversity in Wireless Mesh Network Y. HU, S. B. YANG, Q. ZHANG, P. ZHANG.....	51
Generality Challenges and Approaches in WSNs F. TIRKAWI, S. FISCHER.....	58
Five Basic Types of Insider DoS Attacks of Code Dissemination in Wireless Sensor Networks Y. ZHANG, X. S. ZHOU, Y. M. JI, Y. W. LAW, M. PALANISWAMI.....	64
On the Scalable Fairness and Efficient Active Queue Management of RED H. WANG, X. H. LIN, K. Y. ZHOU, N. XIE, H. LI.....	73
A Genetic Based Fuzzy Q-Learning Flow Controller for High-Speed Networks X. LI, Y. W. JING, N. JIANG, S. Y. ZHANG.....	84

