International Journal of

Communications, Network and System Sciences

ISSN: 1913-3715

Volume 2, Number 3, June 2009

www.scirp.org/journal/ijcns/

Scientific Research

JOURNAL EDITORIAL BOARD

ISSN 1913-3715 (Print) ISSN 1913-3723 (Online)
Http://www.scirp.org/journal/ijcns/

Int. J. Communications, Network and System Sciences, **2009, 3, 169-247**
Published Online June 2009 in SciRes (http://www.SciRP.org/journal/ijcns/).

TABLE OF CONTENTS

Volume 2 Number 3 June 2009

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 monthly 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

PRODUCTION INFORMATION

For manuscripts that have been accepted for publication, please contact:
E-mail: ijcns@scirp.org

Int. J. Communications, Network and System Sciences, 2009, 3, 169-247
doi:10.4236/ijcns.2009.23019 Published Online June 2009 (http://www.SciRP.org/journal/ijcns/).

Device-to-Device Communication Underlaying Cellular Communications Systems

Pekka JÄNIS[1], Chia-Hao YU[2], Klaus DOPPLER[3], Cássio RIBEIRO[3], Carl WIJTING[3], Klaus HUGL[3], Olav TIRKKONEN[2], Visa KOIVUNEN[1]
[1]*Department of Signal Processing and Acoustics, Helsinki University of Technology, Espoo, Finland*
[2]*Department of Communications and Networking, Helsinki University of Technology, Espoo, Finland*
[3]*Nokia Research Center, Nokia Group, Helsinki, Finland*
Email: *pekka.janis@tkk.fi*
Received December 9, 2008; *revised March* 14, 2009; *accepted March* 18, 2009

ABSTRACT

In this article we propose to facilitate local peer-to-peer communication by a Device-to-Device (D2D) radio that operates as an underlay network to an IMT-Advanced cellular network. It is expected that local services may utilize mobile peer-to-peer communication instead of central server based communication for rich multimedia services. The main challenge of the underlay radio in a multi-cell environment is to limit the interference to the cellular network while achieving a reasonable link budget for the D2D radio. We propose a novel power control mechanism for D2D connections that share cellular uplink resources. The mechanism limits the maximum D2D transmit power utilizing cellular power control information of the devices in D2D communication. Thereby it enables underlaying D2D communication even in interference-limited networks with full load and without degrading the performance of the cellular network. Secondly, we study a single cell scenario consisting of a device communicating with the base station and two devices that communicate with each other. The results demonstrate that the D2D radio, sharing the same resources as the cellular network, can provide higher capacity (sum rate) compared to pure cellular communication where all the data is transmitted through the base station.

Keywords: Peer-to-Peer, Device-to-Device, Power Control, Cellular Systems, IMT-Advanced

1. Introduction

Major effort has been spent in recent years on the development of next-generation wireless communication systems that will bring higher data rates and system capacity to end users and network operators. Examples of such next-generation systems are 3GPP Long Term Evolution (LTE) and WiMAX (see http://www.3gpp.org/ and http://www.wimaxforum.org/).

Currently the evolution of such systems has been started under the scope of IMT-Advanced. In addition to traditional performance targets of high data rates and better coverage, the success of IMT-Advanced systems will depend on their ability to enable new services. It is expected that local services will contribute significantly to the growth of mobile communications. The widespread development of local services will be enabled by decreasing infrastructure costs and direct connectivity that supports peer-to-peer communication between local services and the end users. In fact already today mobile phones act as web server (see http://mymobilesite.net/) and offer direct connectivity, e.g. using Bluetooth technology.

In this article we propose to facilitate the local peer-to-peer communication by a Device-to-Device (D2D) radio that operates as an underlay network to an IMT-Advanced cellular network. This D2D radio is a potential key enabler for low cost, seamless and high capacity local connectivity. We assume the infrastructure network to be a cellular network based on Orthogonal Frequency Division Multiple Access (OFDMA) technology. The cellular network operates in licensed bands, and it is important to guarantee that D2D transmissions will not generate harmful interference to cellular users. Similar

problems are observed in the context of cognitive radios [1−3], where the cellular usage is the primary service.

In order to control the interference from D2D connections to the cellular network, we propose that the Base Station (BS) is able to control the maximum transmit power and the resources to be used for each D2D connection. Note that such a scenario is different from pure ad-hoc networks, without coordination from an infrastructure network, e.g. [4,5]. Further, we present a novel power control mechanism for D2D connections that share cellular uplink resources. The mechanism limits the maximum D2D transmit power, utilizing cellular power control information of the devices in D2D communication. The performance of D2D and cellular communications is evaluated by means of system simulations that include interference from multiple cells.

Secondly, we study a single cell scenario consisting of a device communicating with the BS and two devices that communicate with each other. We consider three modes of operation: D2D communication can share either uplink (UL) or downlink (DL) resources with the cellular network or use exclusive resources. If direct communication between the terminals is not beneficial, the two devices communicate through the BS of the cellular network. In semi-analytical studies we show that the D2D radio, sharing the same resources as the cellular network, can provide higher capacity (sum rate) than pure cellular communication through the BS.

This article is organized as follows: In Section 2 we present the motivation for mobile D2D communications with an example application and give a brief overview of the state of the art in D2D communication. In Section 3 we present the power control mechanism for the coexistence of D2D and cellular transmissions. In Section 4 we describe the simulation methodology and present the simulation results. In Section 5 the semi-analytical analysis is described and results are presented. In Section 6 we present results on indoor D2D connections sharing the DL resources with a metropolitan area network. In Section 7 we summarize our results and the conclusions are given.

2. Mobile Device-to-Device Communication

Next generation mobile communication systems such as 3GPP LTE and WiMAX are optimized for wide area and metropolitan area operation. In recent years local area networks based on WLAN have been increasingly popular, as they enable access to the internet and to local services with low cost APs and cheap and fast access to wireless spectrum in the license exempt bands. However only a licensed band can guarantee a controlled interference environment and local service providers might prefer to pay a small amount of money to get access to licensed spectrum when the license exempt bands get crowded. Cellular operators may offer such cheap access to spectrum with controlled interference enabled by D2D communication as underlay to the cellular network.

This concept is illustrated in Figure 1, where UE denotes User Equipment. The BS allows UE2 and UE3 to communicate directly to each other while keeping some control over the D2D link to limit the interference to the cellular receiver. As an example, consider the case where a media server is put up at a rock concert from which visitors can download promotional material using the D2D connection. At the same time, the cellular network can handle phone calls and internet data traffic without the additional load that would be caused by traffic from the media server. The D2D operation itself can be transparent to the user. She simply enters a URL, the network would detect traffic to the media server and hand it over to a D2D connection. The same application could also be enabled by a media server with built in WLAN AP or Bluetooth. However in that case the user has to define the WLAN AP or perform Bluetooth pairing which can be tedious especially if a secure connection is required.

Compared to other local connectivity solutions based on for example Bluetooth or WLAN the D2D communication supported by a cellular network offers additional compelling advantages. First the network can advertise local services available within the current cell. Thus for automated service discovery, the devices do not have to constantly scan for available WLAN AP or Bluetooth devices. This is especially advantageous when considering that the constant scanning of Bluetooth devices or WLAN APs is often switched off to reduce the power consumption. Secondly, the cellular network can distribute encryption keys to both D2D devices so that a secure connection can be established without manual pairing of devices or entering encryption keys.

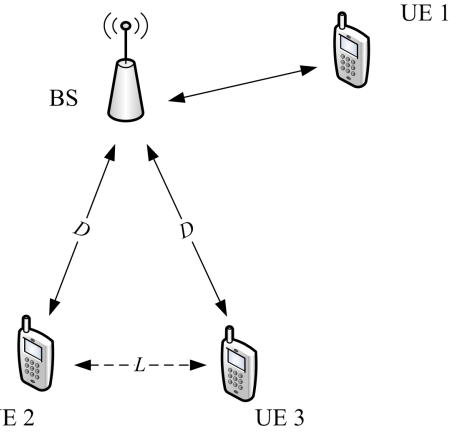


Figure 1. Illustration of D2D communication as an underlay network to an infrastructure network. UE1 is a cellular user whereas UE2 and UE3 are D2D users. *D* denotes the distance between D2D nodes and BS, and *L* denotes the D2D link distance.

Several wireless standards have addressed the need for D2D operation in the same band as the BS, also called Access Point (AP) or central controller. Examples of such standards are Hiperlan 2 [6], TETRA [7], and WLAN.

In all these standards D2D communication is assumed to occur on separate resources. For example, in standards employing Orthogonal Frequency Division Multiplexing (OFDM) as the physical layer, like Hiperlan 2, User Equipments (UE) involved in D2D communications are not allowed to share the same OFDM symbol with UEs communicating to the infrastructure network. This restriction limits the interference. However it leads to inefficient utilization of resources, especially for large system bandwidths. The resource utilization is even more inefficient in TETRA, where several frequency channels are country-wide reserved solely for D2D communication; reducing the resources available for cellular communication when no D2D communication is present.

In WLAN the UE senses the medium and transmits if the resources are free. A drawback of such a scheme is that the AP does not have a direct possibility of controlling the D2D links and providing assistance, which could prove highly beneficial for the network [8].

3. D2D Power Control when Sharing UL Resources

Since the D2D communication takes place as an underlay communication to the cellular OFDMA network, the interference from D2D communication to the cellular network has to be coordinated and the BS should be aware of ongoing D2D connections. The UEs in D2D connections are still associated to the BS and can receive for example cellular calls. Thus, we propose that the D2D link initialization and the allocation of OFDMA Resource Block (RB) to the D2D links is managed by the BS. Therefore, there is an immediate opportunity for the BS to reduce the interference between the cellular and D2D links. Such a scheme for sharing UL resources is proposed in this section. The power level of D2D transmitters is chosen based on the cellular UL power control information to limit the interference to the cellular BS.

The easiest way to restrict the D2D interference would be to mandate a predefined maximum power level to the D2D transmitters, and this level could be chosen such that the expected degradation in the cellular links stays at a tolerable level. However, such an approach would have to be designed for the worst case scenario and would lead to inefficient use of resources. As the D2D transmitter may be arbitrarily close to the cellular receiver, the power level thus determined is likely to be inadequate for establishment of reliable D2D links, other than for extremely short range communication. On the other hand, the power level of the D2D transmitter could be substantially higher if the network would have some means to determine how close the D2D transmitter is to the cellular receiver.

In fact the cellular BS has just the required information for controlling the interference from D2D transmitters to the cellular BS in case the D2D links share UL resources with the cellular network. The UL power control in a cellular network aims at reducing the dynamic range of signals received from multiple devices, i.e. to reduce the near-far effect. The BS may use the cellular UL power control framework in setting the D2D transmit power. To be more specific, let us consider the SINR of the UL cellular transmission in an isolated cell with ideal transceivers and flat fading channel. In this case the expression for the cellular UL SINR may be written as

$$\xi = \frac{P_1 c_1}{P_2 c_2 + \sigma_w^2} \quad \textbf{(1)}$$

where P_1 and P_2 denote the transmit powers of the cellular and D2D UEs, c_1 and c_2 the corresponding link gains to the base station, and σ_w^2 the additive white Gaussian noise power. The base station has full control over the powers P_1 and P_2 in Equation (1), given the limitations on the transmit power range of the terminal and on the dynamic range of the power control specified for the radio interface.

Equation (1) implies that in case of ideal UL power control without the presence of a D2D transmitter (P_2=0) and a target SNR of P/σ_w^2, the cellular power control target is P_1c_1=P.

In order to keep the interference to UL transmissions under control, the BS can signal the D2D transmitter to apply a power level such that P_2c_2=P/B, where B is a backoff parameter. For large values of the backoff parameter B, D2D transmissions cause very low interference to UL transmissions. However, a large B implies a reduced range for the D2D link itself. We can avoid this limitation on the range of the D2D link by incorporating a power boosting factor α to the transmit power of the UL transmitter that compensates for the remaining interference from D2D transmissions. In this case, Equation (1) is modified to

$$\xi' = \frac{\alpha P_1 c_1}{P_2 c_2 + \sigma_w^2} \quad \textbf{(2)}$$

The power boosting value α is defined such that the received SINR from UL transmission in Equation (2) is equal to the target SNR, i.e. $\xi' = P/\sigma_w^2$. Hence, substituting $P_1 c_1 = P$ and $P_2 c_2 = P/B$ into Equation (2), we obtain

$$\alpha = \frac{P}{B\sigma_w^2} + 1 \quad \textbf{(3)}$$

Naturally, in case of no UL transmissions, the D2D transmitter does not need to apply a power backoff, i.e.

$B = 1$. Conversely, no power boosting is needed in case of no D2D transmissions. In fact, in cases when only a subset of the RBs is used by D2D traffic, the power boosting is only applied to those UL transmissions that share the RBs with a D2D pair. As a result, the received UL power is non-uniform over the system bandwidth, which tends to increase the inter-RB interference caused by power amplifier nonlinearities and limited receiver dynamic range. Moreover, UL transmissions are not perfectly orthogonal due to the effects of non-ideal synchronization and wireless propagation environment. Therefore we limit the boosting values to 10dB, which we assume to be still manageable.

4. Numerical Results on Coexistence of D2D Communication and Cellular Network

In this section we study the coexistence of D2D communication links with an interference-limited cellular local area network. The D2D pair is sharing either UL or DL resources with the cellular links. The aim is to find out the achievable D2D link quality when giving priority to the cellular links. The study is carried out by static system simulations and empirical SINR distributions for both the cellular and D2D links are evaluated.

4.1. Scenario and Channel Model

The scenario and network layout resembles a local area indoor scenario, illustrated in Figure 2. Nine BS serve a whole floor of 100m times 100m. The scenario incorporates small room, corridor like longer rooms and a large open area in the center. Similar elements can be typically found in shopping malls or office areas.

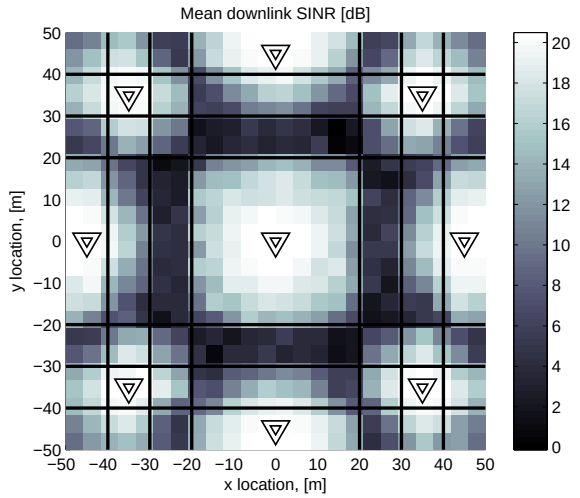


Figure 2. The simulated scenario for coexistence studies. The triangles represent the locations of base stations and the black horizontal and vertical lines represent walls. The color indicates the mean DL SINR without D2D interference as a function of location.

We have used the channel and propagation models defined in WINNER [9] scenario A1 (indoor/office) for our studies. Links in the same room have a distance dependent probability for Line-Of-Sight (LOS) conditions.

In the channel model, for the LOS and Non-Line-Of-Sight (NLOS) propagation conditions, the path-loss exponent is 1.87 and 3.68, respectively, and the shadow fading standard deviation is 3dB and 4dB, respectively. In addition, each wall introduces an additional attenuation of 5dB. Frequency selective fading is also modeled, with a resolution of 2MHz.

The cellular UEs are uniformly distributed over the area and the locations of the D2D pairs are independent from the cellular UEs. The D2D pairs are generated with the restriction that the D2D link must reside within a single room.

4.2. System Model

We assume that the network operates on a 100MHz band using Time Division Duplexing (TDD). The base stations have acquired frame-synchronism and use the same split between UL and DL resources, such that there is no interference from neighboring cell DL transmission to UL transmissions or vice versa. The modulation scheme allows Frequency Division Multiplexing (FDM) transmissions from the BS to several UEs simultaneously, as well as Frequency Division Multiple Access (FDMA) for several UEs to the BS. Specifically, the 100MHz band is split into five orthogonal RBs of 20MHz.

4.2.1. Scheduler for Cellular Transmissions

Each BS randomly selects five UEs to be scheduled, from those UEs that are not in D2D mode. For each UE it tries to allocate the RB with best SNR or the RB with second best SNR. If these are not available, the first free RB is allocated. In case there are less than five UEs associated to the BS, the remaining free RBs are allocated to a UE with allocation in the adjacent RB. Hence, the network is fully loaded at every time instant. Since the channel is assumed reciprocal, the same frequency resources that are used for UL are also used for DL.

4.2.2. UL Power Control

For each BS, the total transmitted power is 25dBm, which is evenly distributed over all sub-bands, i.e. 18dBm for each 20MHz band. For the UE, the uplink power control aims a target SNR of P/σ_w^2, limited by the maximum transmit power of 18dBm for the UE. With these settings, in the studied scenario about 10% of the UEs utilize maximum output power. When the UL power boosting defined in Section 3 is used, the portion of UEs reaching maximum transmit power increases to 40%.

4.2.3. SINR Calculations

The SINR of each 2MHz sub-band for a transmission originating from node *n* and received by node *m* is calculated as

$$\xi_{n,m} = \frac{P_n c_{n,m}}{\sigma_w^2 + \sigma_{\mathrm{Tx}}^2 + \sigma_{\mathrm{Rx}}^2 + I_{\mathrm{NL}} + \sum_{k \neq n} P_k c_{k,n}}$$

where I_{NL} is the power of all out-of-band emissions at the receiver, and σ_w^2, σ_{Tx}^2, and σ_{Rx}^2 are the thermal noise power, the transmitter in-band distortion, and the receiver Analog-to-Digital Converter (ADC) noise floor, respectively, integrated over the 2MHz sub-bands. The thermal noise power is derived from the noise figure defined in LTE specifications for UE and BS [10]. With the chosen transmit power levels our network scenario is clearly interference-limited.

4.3. Network Simulation Description

The simulation arrangement is as follows. A large number of random independent snapshots of network operation, called drops, are modeled. For each drop, a set of cellular and D2D UEs are independently placed in the scenario. The path-loss and shadow fading values of each link are then determined and each cellular and D2D UE is associated with the BS to which it has the strongest link. Each cell has 5 cellular UEs and 5 D2D pairs, and the D2D transmissions are multiplexed to 5 sub-bands. This way we can ensure that each cellular transmission is interfered by exactly one intracell D2D link and vice versa. After all transmissions are scheduled, the transmit powers of the D2D transmitters are determined as in Section 3 and the SINR is computed as in Subsection 4.2.3.

4.4. Results and Discussion

In this section we present the simulation results on the SINR distributions for the cellular UL, cellular DL, and the D2D links. A wide range of transmit power levels without power control for the D2D link was simulated along with an UL power control based D2D case. The settings for the power control case have been chosen such that for 95% of the cellular links the SINR degradation is less than 3dB. This was achieved with a power backoff of $B = 5$ dB, for a UL power control target SNR of 13dB. From Equation (3), this implies an UL power boost $\alpha = 8.64$ dB.

From Figure 3(a) and 3(b), we observe that the maximum allowed D2D transmit power should be limited to -10dBm in DL and -24dBm in the UL phase of the frame, assuming that a 3dB degradation of the cellular SINR at the 5-th percentile of the SINR CDF would be still tolerated. Assuming as well that a D2D link with $\mathrm{SINR} \geq 0$ dB is usable, we observe from 3(c) and 3(d) that the fraction of usable D2D links is ≈45% in DL phase and ≈33% in UL phase.

However, if the UL-based D2D power control scheme is applied, the percentage of usable links rises to 73% in UL phase while still maintaining the same cellular performance as for -24dBm D2D transmit power, as observed in Figure 3(b) and 3(d).

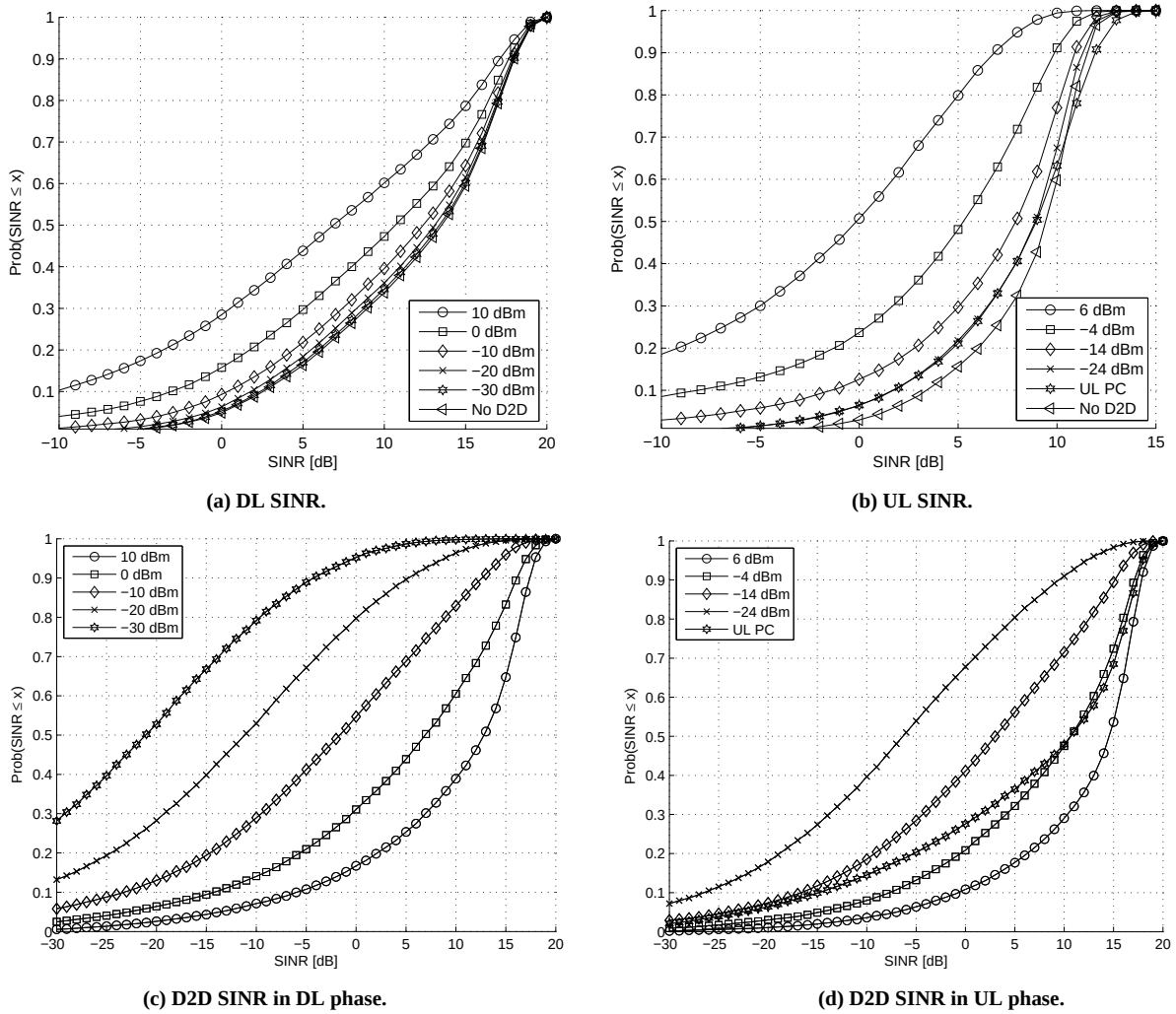
As it can be appreciated from Figure 3 and from the discussion above, when the D2D transmit power is set to a fixed level such that the degradation to the cellular performance remains tolerable, D2D performance is slightly better when it uses DL resources than when it uses UL resources. This may sound counter-intuitive since, due to the fact that the BS's constant transmit power is significantly higher than the mean UL transmit power, the interference to the D2D links is higher in the DL than in UL phase. On the other hand, due to the same reason the cellular DL transmissions can tolerate much higher D2D transmit powers than the cellular UL transmissions. In the UL phase, the D2D transmit power must be set such that even in the event of a D2D transmitter being close to the BS the cellular performance does not degrade too much. Since UL power control guarantees that all UEs experience similar SINR in their UL transmissions regardless of their position in the network, this becomes a very strict requirement.

The proposed UL-based power control scheme effectively removes this restriction, resulting in significant improvement on the D2D performance. A similar power control scheme is not applicable in DL since the BS might schedule resources shared with D2D links to multiple cellular UEs. Each of these candidate cellular UEs would have to set a power control target to the D2D transmitters, implying significant overhead.

5. Resource Allocation Analysis

In the preceding sections we considered the coexistence of a D2D underlay and cellular network. Specifically, we demonstrated that it is possible to allow D2D communication to share the cellular resources and at the same time guarantee that the performance of the cellular communication links is not sacrificed, thus taking an approach where the cellular communication has priority. In this section we take a different viewpoint where neither the cellular nor the D2D communication have priority over the other. This gives insight on the maximum benefits in terms of overall performance that D2D underlay communication can provide.

We assume the Channel State Information (CSI) of all the involved links is available at BS so that the resource allocation decision of D2D users can be controlled centrally by the BS. The scenario where at most one cellular



(a) DL SINR.

(b) UL SINR.

(c) D2D SINR in DL phase.

(d) D2D SINR in UL phase.

Figure 3. Empirical SINR CDFs in the local area scenario with various D2D transmitter power levels. The UL and DL SINR without D2D is shown for reference. The D2D SINR distributions are shown conditioned on the link distance being less than 8 meters. Each D2D link is within a single room.

user and one D2D communication pair will share the same radio resource is considered. Despite its simplicity, this scenario captures the minimum requirement for cellular communication and D2D communication to share the same resource.

In general, D2D communication causes no interference to the cellular users if they occupy separate resources. However, resource usage efficiency can be higher if the same resource is shared at the same time. We may achieve higher overall system performance if D2D and cellular communications co-exist in the same radio resource. We will discuss four different resource allocation modes including both separate and non-separate sharing schemes. They are illustrated in Figure 4 and detailed below:

- DL resource sharing (DLre): D2D communication happens in DL resources so that all the DL resources of the cellular user are interfered.
- UL resource sharing (ULre): Similar to DLre, D2D communication happens in UL resources, and all the UL resources of the cellular user are interfered.
- Separate resource sharing (SEPre): D2D communication takes half of the available resources from the cellular user, either from DL or UL resource. There is no interference between cellular and D2D communication.
- Cellular mode sharing (CellMod): The D2D users communicate with each other through the BS that acts like a relay node. They take half of the available resources either from the DL or the UL resources of the cellular user. Note that this mode is conceptually the same as traditional cellular system and is used as a reference.

In the following, we consider a normalized isolated circular cell (with radius equal to 1) as illustrated in

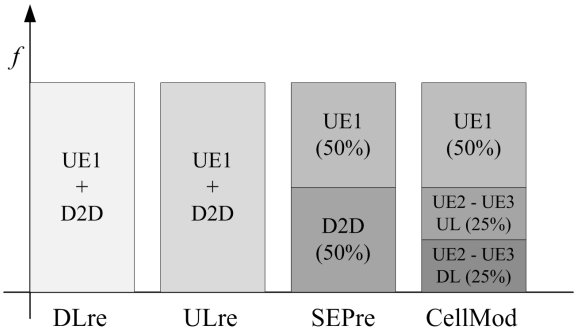


Figure 4. Illustration of resource allocation of considered resource sharing methods. In DLre and ULre modes, the cellular user and the D2D users operate in the same resource. In SEPre and CellMod modes, the cellular user and the D2D users occupy different resources.

Figure 1, and discuss the best resource allocation scheme out of the four possible modes. We assume one cellular user (UE1) and two D2D users (UE2 and UE3) sharing the available radio resources. For simplicity, we consider only distance-dependent pathloss, but no fading. Specifically, we consider the single-slope pathloss model [11] with pathloss exponent 4:

$$P(d) = \frac{P(d_0)}{d^4} \tag{4}$$

where $P(d)$ denotes the received power at the distance d from the transmitter and $P(d_0)$ is the received power at reference distance d_0. To adapt the normalized cell considered in our environment, we simply replace $P(d_0)$ with the transmit power. This channel model enables a one-to-one mapping between the distance of a channel link and the received signal strength. In addition, since the considered channel model provides the mean channel condition in a fading channel, the trend presented in this simplified model is consistent with the case where a more complex model is applied. We assume the distance between the D2D users and the BS to be D and the distance between the two D2D users to be L. Assuming no power control, the transmit power is fixed to unity for the cellular transmission in UL and DL, and for the D2D user transmissions. Note that the per-RB transmit powers used in Section 4 are according to this scheme-the BS and the UE have the same (maximum) power spectral density. Under this channel model and the geometric constraint of the D2D users, the resource allocation decision depends on the cellular user (UE1) position only, under a given set of D and L.

The interference caused by D2D users may come from any D2D users depending on which one is transmitting at the moment. Here, we assume the worst interference condition where the interference from D2D communication is caused by the user creating stronger interference. The AWGN noise power is assumed to be the same as the signal power received at the cell border (i.e. SNR=0dB at the cell edge). The metric for determining the resource sharing mode is the sum rate of the connection between UE2 and UE3, and of the cellular connection between BS and UE1. The sum rate takes into account either DL or UL resources depending on which one is shared with the D2D users. The sum rate is calculated by the Shannon capacity formula [12] according to the following equations

$$R_{\text{ULre}} = \log_2\left(1+\frac{P_{23}}{\max(P_{12},P_{13})+N_0}\right) + \log_2\left(1+\frac{P_1}{\max(P_2,P_3)+N_0}\right)$$

$$R_{\text{DLre}} = \log_2\left(1+\frac{P_{23}}{\max(P_2,P_3)+N_0}\right) + \log_2\left(1+\frac{P_1}{\max(P_{12},P_{13})+N_0}\right)$$

$$R_{\text{SEPre}} = \frac{1}{2}\log_2\left(1+\frac{P_{23}}{N_0}\right)+\frac{1}{2}\log_2\left(1+\frac{P_1}{N_0}\right)$$

$$R_{\text{CellMod}} = \frac{1}{2}\cdot\frac{1}{4}\left(\log_2\left(1+\frac{P_2}{N_0}\right)+\log_2\left(1+\frac{P_3}{N_0}\right)\right) + \frac{1}{2}\log_2\left(1+\frac{P_1}{N_0}\right),$$

where P_i denotes the received power of the link between BS and UE i, and P_{ij} denotes the received power of the link between UE i and UE j. N_0 is the noise power at the receiver. The received power in each link is calculated by Equation (4).

The resource allocation scheme which gives the best sum rate is selected for each UE1 position according to

$$R_{\max} = \max\left(R_{\text{ULre}}, R_{\text{DLre}}, R_{\text{SEPre}}, R_{\text{CellMod}}\right) \tag{5}$$

It should be noted that, without any further constraint, it may happen that either the cellular or the D2D connection is compromised in order to maximize the sum rate. For example, under the condition that UE1 is very close to BS, it is likely that the connection between the BS and the UE1 dominates the selection of the resource allocation scheme. The selected scheme might give the D2D connection little transmission rate. Similarly, when the D2D users are very close to each other and dominate the sum rate, the transmission rate of UE1 may be very limited.

Figure 5 shows the share of cell area where one specific resource allocation scheme is selected as the best one, under different values of D and L. The curves

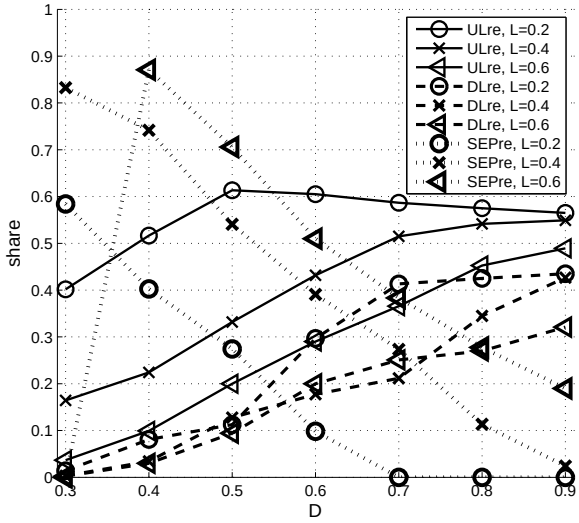


Figure 5. Share of cell area where one specific resource allocation scheme is selected as the best one, for different values of *D* (the distance between BS and the D2D pair) and *L* (the distance between the two D2D users). D2D communication is always beneficial for the system except for the case when the D2D users are at opposite sides with respect to BS (i.e. *D*=0.3 and L=0.6), where none of ULre, DLre and SEPre resource allocation schemes occupies significant percentage area.

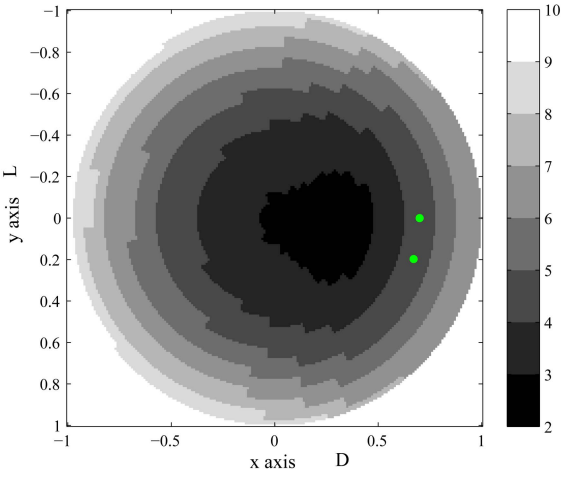


Figure 6. Rate ratio to CellMod mode with *D*=0.7 and *L*=0.2. The two circular spots denote the position of D2D users. The cellular user is at a given position in the cell and the background color at the position displays the rate ratio of the best resource sharing scheme to the CellMod mode.

corresponding to CellMod mode are missing because it is selected only under the condition that the two D2D users are at opposite sides with respect to BS $(L \approx 2D)$. In this special condition, the CellMod mode is the favorable resource sharing scheme. This can be observed from Figure 5 by noticing that the share of all curves with $L = 0.6$ goes to approximately 0 at $D = 0.3$. However, except for this particular case, D2D communication is always beneficial for the system.

When the D2D users are further away from BS (i.e. when D is large), the percentage area where the cellular user experiences strong interference reduces. It suggests the benefit of using non-separate resource sharing schemes (ULre or DLre) which provide higher resource usage efficiency. When D is small, it is more beneficial to use either ULre or SEPre depending on the value of L. In small D and small L scenario, the signal strength between the D2D users is very strong. The ULre mode outperforms DLre mode in this case because the interference observed by the D2D users is smaller in ULre mode, which significantly improves channel quality of D2D communication.

Figure 6 shows the rate ratio of D2D communication to the CellMod mode under the parameter $D = 0.7$ and $L = 0.2$. The two circular spots give the position of two D2D users. The cellular user is at one given position in the cell, and the color at that position represents the rate gain from D2D communication, which is the rate ratio between the rates obtained from the best resource sharing scheme and the CellMod mode. The area outside the unit circle is out of the considered cell and should not be considered. The gain is significant and depends on the position of the cellular user. In Figure 7, we illustrate the rate gain averaged over the considered single-cell with respect to different geometry of D2D users. It is clear that, in average, the larger D and the smaller L are beneficial for the system performance. Consistent with Figure 5, the rate gain vanishes when the D2D users are at opposite sides of BS (e.g. $D = 0.3$ and $L = 0.6$).

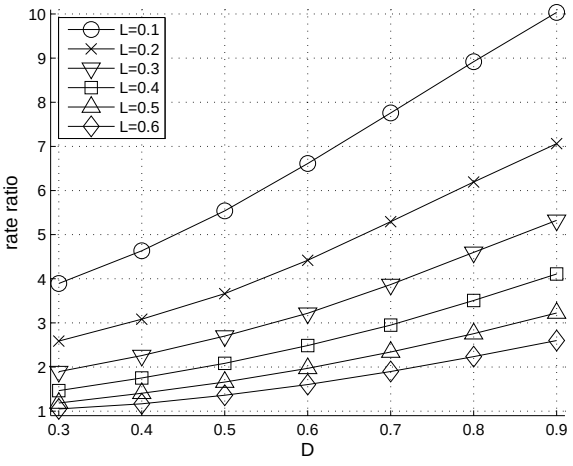
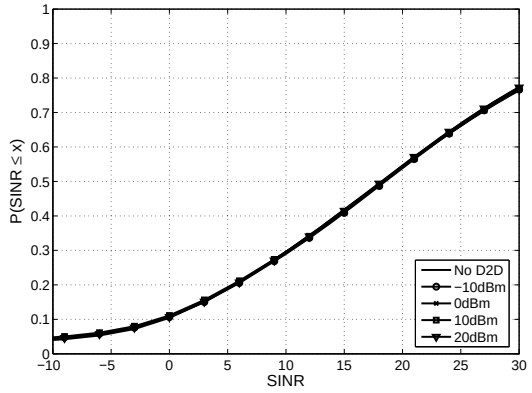


Figure 7. Average rate ratio over CellMod. The rate ratio is obtained by comparing the rate obtained from the best resource sharing scheme and the rate obtained from the CellMod mode. For different values of *D* and *L*, the rate gain is averaged over the whole cell.

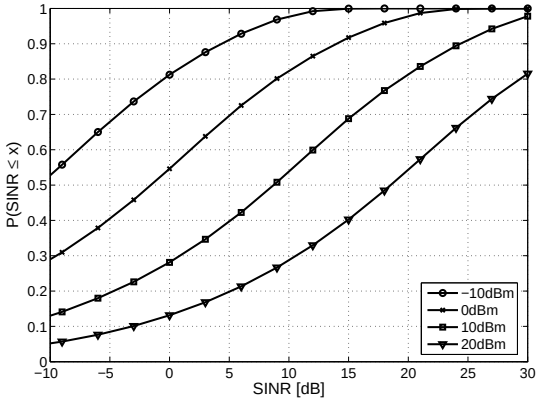
6. Indoor D2D as Underlay to a Metropolitan Area Network

In the previous sections we have demonstrated that D2D communication can take place in an interference-limited network as well as the potential gains from D2D communication in a single cell. Now we consider D2D as an underlay to a metropolitan area network.

The cellular BS are deployed outdoors and outdoor cellular users share downlink resources with indoor D2D connections. The BSs are deployed in a multi-cell environment and the results are obtained from the center cell. The BS deployment is modelled by the well known Manhattan grid and follows the UMTS 30.03 recommendation [13] and the corresponding channel and path-loss models can be found in [9]. The D2D pairs are randomly generated within the same building block and the path-loss between them is below 90dB which corresponds to a distance of up to 25m. The penetration loss of at least 14dB through the outside wall of the building and the favorable propagation between outdoor BS and cellular devices in the same street isolates the indoor D2D connections from the outer cellular network.



(a) Empirical CDF of cellular SINR for different D2D transmission power. There is no visible impact from the D2D communication on the SINR of the cellular network.



(b) Empirical CDF of D2D connection SINR for different D2D transmission power.

Figure 8. SINR of cellular and D2D connections sharing the cellular downlink resources.

Both the cellular devices and the D2D devices operate with full buffers, i.e. both the cellular network and the D2D devices utilize the full bandwidth with 100% load. A single OFDMA resource block (RB) is shared by one cellular user and one D2D pair in the cell.

The transmit power of the cellular BS is set to 37dBm and the transmit power of the D2D devices was limited to 20dBm. Figure 8 illustrates the potential for D2D connection with different transmit power in such a scenario. The cellular SINR is not affected by the indoor D2D connections even when they transmit with 20dBm. About 90% of the D2D connections experience a higher SINR than 0dB which we see as a lower threshold where D2D communication makes sense. In general the BS will also serve indoor users and the example might be overly optimistic. Nevertheless the BS can for example allocate only part of the downlink resources to D2D connections and schedule only outdoor cellular users in these resources. The BS can for example classify outdoor users based on path-loss, spatial signature or location information.

7. Conclusions

In this paper we analyze Device-to-Device (D2D) communications underlaying a cellular network. We show that given proper power control and coordination mechanisms it is possible to have D2D connections that reuse cellular band and still cause only minimal interference to the cellular network.

We propose a power control scheme for the D2D links that share uplink resources with a cellular network. In this case the maximum power that can be used for the D2D link is defined by taking the cellular uplink power control information as reference. We evaluate the proposed power control scheme in system simulations. The results show that by properly defining the maximum power on the D2D link a good D2D link SINR is achieved while at the same time the impact on the cellular network is minor. Thereby D2D communication can take place in interference-limited networks with full load, where a cognitive radio would not be able to detect a white space.

Further, we performed semi-analytical studies on a single-cell scenario to analyze how much gain can be expected from D2D communications. We considered several allocation strategies, including traditional cellular communications. The results show that significant gains in sum rate can be achieved by enabling D2D communications compared to the conventional cellular system.

Finally, we showed in system simulations that indoor D2D communication causes negligible interference to outdoor cellular users in the downlink of a metropolitan area network.

8. References

[1] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," IEEE Journal on Selected Areas in Communications, Vol. 23, No. 2, pp. 201−220, February 2005.

[2] J. Mitola and G. Q. Maguire Jr., "Cognitive radio: Making software radios more personal," IEEE Personal Communications, Vol. 6, No. 4, pp. 13−18, August 1999.

[3] I. F. Akyldiz, W. -Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation dynamic spectrum access cognitive radio wireless networks: A survey," Computer Networks: The International Journal of Computer and Telecommunications Networking, Vol. 50, No. 13, pp. 2127−2159, September 2006.

[4] C. -K. Toh, M. Delwar, and D. Allen, "Evaluating the communication performance of an ad hoc wireless network," IEEE Transactions on Wireless Computing, Vol. 1, No. 3, pp. 402−414, July 2002.

[5] Y. Xue, B. Li, and K. Nahrstedt, "Optimal resource allocation in wireless ad hoc networks: A price-based approach," IEEE Transactions on Mobile Computing, Vol. 5, No. 4, pp. 347−364, April 2006.

[6] ETSI, "BRAN; HIPERLAN2 type 2; data link control (DLC) layer; part 4: Extension for home environment," TS 101 761−4, v1.3.2, 2002.

[7] ETSI, "Terrestrial trunked radio (TETRA); voice plus data (V+D) designers' guide; part 3: Direct mode operation (DMO)," TR 102 300-3 v1.2.1, 2002.

[8] H.-Y. Hsieh and R. Sivakumar, "On using peer-to-peer communication in cellular wireless data networks," IEEE Transactions on Mobile Computing, Vol. 3. No. 1, pp. 57−72, January−February 2004.

[9] WINNER II D1.1.2, "WINNER II channel models," https://www.ist-winner.org/deliverables.html, September 2007.

[10] 3GPP TR 25.814 V7.1.0, "Technical specification group radio access network; physical layer aspects for evolved universal terrestrial radio access (UTRA)," http://www.3gpp.org/, September 2006.

[11] T. S. Rappaport, "Wireless communication principles and practice," New Jersey: Prentice Hall, 1996.

[12] T. M. Cover and J. A. Thomas, "Elements of information theory," New York: Wiley, 1991.

[13] ETSI, "Recommendation TR 30.03 selection procedure for the choice of radio transmission technologies of the UMTS," 1997.

Int. J. Communications, Network and System Sciences, 2009, 3, 169-247
doi:10.4236/ijcns.2009.23020 Published Online June 2009 (http://www.SciRP.org/journal/ijcns/).

An Improved Power Estimation for Mobile Satellite Communication Systems

Byounggi KIM[1], Namgil LEE[2], Sangjin RYOO[3]
[1]*Huneed Technologies, Gunpo-si, Korea*
[2]*Department of Information & Communication System, Ulsan Korea Polytechnic College, Ulsan, Korea*
[3]*Department of Computer Media, Hanyeong College, Yosu, Korea*
Email: *kimbg@huneed.com*, *axtomato@hanmail.net*, *sjryoo@hanyeong.ac.kr*
Received November 8, 2008; *revised March* 28, 2009; *accepted April* 5, 2009

ABSTRACT

In this paper, in order to increase system capacity and reduce the transmitting power of the user's equipment, we propose a efficient power estimation algorithm consisting of a modified open-loop power control (OLPC) and closed-loop power control (CLPC) for mobile satellite communications systems. The improved CLPC scheme, combining delay compensation algorithms and pilot diversity, is mainly applied to the ancillary terrestrial component (ATC). ATC link in urban areas, because it is more suitable to the short round-trip delay (RTD). In the case of rural areas, where ATCs are not deployed or where a signal is not received from ATCs, transmit power monitoring equipment and OLPC schemes using efficient pilot diversity are combined and applied to the link between the user's equipment and the satellite. Two modified power control schemes are applied equally to the boundary areas where two kinds of signals are received in order to ensure coverage continuity. Simulation results show that the improved power control scheme has good performance compared to conventional power control schemes in a geostationary earth orbit (GEO) satellite system utilizing ATCs.

Keywords: Power Control, Pilot Diversity, ATC

1. Introduction

In 4G systems, the major role of satellites will be to provide terrestrial fill-in service and efficient multicasting/broadcasting services [1]. However, it is known that it is difficult for a mobile satellite service (MSS) to reliably serve densely populated areas, because satellite signals are blocked by high-rise structures and/or do not penetrate into buildings. Under these circumstances, in a groundbreaking application to the Federal Communication Commission (FCC) in 2001, Mobile Satellite Ventures LP (MSV) unveiled a bold new architecture for an MSS with an ancillary terrestrial component (ATC) providing unparalleled coverage and spectral efficiency [2]. The main concept of the hybrid MSS/ATC architecture of the MSV proposal is that terrestrial reuse of at least some of the satellite band service link [3] frequencies can eliminate the above-mentioned problem. As the terrestrial fill-in services using ATC [4], satellite systems provide services and applications similar to those of terrestrial systems outside the terrestrial coverage area as much as possible.

This paper examines power control and handover using position information in land mobile satellite communication systems containing an ATC. The MSV's hybrid system architecture is shown in Figure 1.

2. Power Estimation Using Pilot Diversity

SIR estimation is one of the key aspects of the OLPC and CLPC scheme and is typically needed for functions such as power control, handoff, adaptive coding, and modulation. Efficient channel estimation is compared with a channel estimation method using only the pilot symbols of the common pilot channel (CPICH), as well as a channel estimation method combining the pilot symbols of the dedicated physical control channel (DPCCH) and those of the CPICH.

Equation (1) represents a channel estimation using N symbols of the CPICH in one slot after a multipath fading and a dispreading process in a RAKE receiver.

$$x(i)=\alpha(i)+n(i) \quad \text{for } i=1, 2, \ldots, N \tag{1}$$

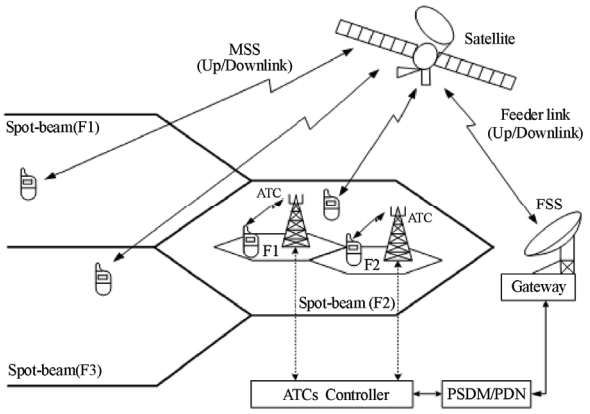


Figure 1. MSV's hybrid system architecture.

in which $\alpha(i)$ is a channel gain to be estimated. Equation (2) shows a method for estimating a channel by combining the pilot symbols of the dedicated physical channel (DPCH) and CPICH. A receiver of a terminal acknowledges a pilot pattern for pilot symbols of the DPCCH in a manner similar to Equation (1).

$$y(j) = \lambda(j)\alpha(j) + m(j) \text{ for } j = 1,2,\ldots, M \quad \textbf{(2)}$$

in which $\lambda(j)=(1/\mu(j))^{1/2}$ refers to the power ratio of the pilot symbols of the DPCH and those of the CPICH. Equation (3) shows a method for estimating a channel by combining the pilot symbols of the DPCH, those of the CPICH, and those of the secondary common control physical channel (S-CCPCH).

$$z(k) = \lambda_1(k)\alpha(k) + l(k) \text{ for } k = 1,2,\ldots, K \quad \textbf{(3)}$$

in which $\lambda_1(k)=(1/\mu_1(k))^{1/2}$ refers to the power ratio of the pilot symbols of the S-CCPCH and those of the CPICH. In Equations (1)−(3), N, M, and K are the number of pilot symbols in one slot of the CPICH, DPCH, and S-CCPCH used in estimating a channel gain, respectively. And $n(i)$, $m(j)$, and $l(k)$ are presumed AWGNs that have zero means and σ^2 variances, respectively. Since it is presumed that the channel gain is not changed during one slot of an estimation period, $\alpha(\cdot)$ and $\lambda_1(\cdot)$ become α and λ_1, respectively. Since channelization codes used in the CPICH, the DPCH, and the S-CCPCH are different from each other, $n(i)$, $m(j)$, and $l(k)$ are independent of each other. A vector of a signal received in a rake receiver of a terminal is as follows:

$$z' = [x(1)x(2)\ldots x(N)y(1)y(2)\ldots y(M)z(1)z(2)\ldots z(L)]^T \quad \textbf{(4)}$$

where T denotes an operator of a transpose matrix. If λ, λ_1, and α are known, a conditional probability density function is as follows:

$$p(z \mid \lambda_1, \lambda, \alpha) = \left(\frac{1}{\sqrt{2\pi\sigma_n^2}}\right)^N \left(\frac{1}{\sqrt{2\pi\sigma_m^2}}\right)^M \left(\frac{1}{\sqrt{2\pi\sigma_k^2}}\right)^K \times \exp\left(-\sum_{i=1}^{N}\frac{(x(i)-\alpha)^2}{2\sigma_n^2} - \sum_{j=1}^{M}\frac{(y(j)-\lambda\alpha)^2}{2\sigma_m^2} - \sum_{k=1}^{K}\frac{(z(j)-\lambda_1\lambda\alpha)^2}{2\sigma_k^2}\right) \quad \textbf{(5)}$$

where α denotes a channel estimation value (case 1) using the CPICH, λ denotes a channel estimation value (case 2) combining the DPCH and the CPICH [5], and λ_1 denotes a channel estimation value (case 3) combining the CPICH, the DPCH, and the S-CCPCH [5].

$$I(\lambda_1,\alpha) = \begin{bmatrix} -E\left(\frac{\partial^2 \ln p(z \mid \lambda_1,\lambda,\alpha)}{\partial^2\lambda_1}\right) & -E\left(\frac{\partial^2 \ln p(z \mid \lambda_1,\lambda,\alpha)}{\partial\lambda_1\partial\alpha}\right) \\ -E\left(\frac{\partial^2 \ln p(z \mid \lambda_1,\lambda,\alpha)}{\partial\alpha\partial\lambda_1}\right) & -E\left(\frac{\partial^2 \ln p(z \mid \lambda_1,\lambda,\alpha)}{\partial^2\alpha}\right) \end{bmatrix} = \begin{bmatrix} \frac{K\lambda^2\alpha^2}{\sigma_k^2} & \frac{2\lambda_1\lambda^2\alpha K}{\sigma_k^2} \\ \frac{2\lambda_1\lambda^2\alpha K}{\sigma_k^2} & \frac{N}{\sigma_n^2}+\frac{M\lambda^2}{\sigma_m^2}+\frac{K\lambda_1^2\lambda^2}{\sigma_k^2} \end{bmatrix} \quad \textbf{(6)}$$

If λ_1 is known, the output Cramer-Rao Lower Bound (CRLB) is that shown by Equation (7) [6].

$$CRLB(\hat{\alpha} \mid \lambda_1) = \frac{1}{I(\lambda_1,\alpha)_{2,2}} = \frac{1}{\frac{N}{\sigma_n^2}+\frac{M}{\mu\sigma_m^2}+\frac{K}{\mu\mu_1\sigma_k^2}} \quad \textbf{(7)}$$

Accordingly, comparing [7] with Equation (7) results in Equation (8), as follows.

$$\frac{1}{\frac{N}{\sigma_n^2}+\frac{M}{\mu\sigma_m^2}+\frac{K}{\mu\mu_1\sigma_k^2}} \le \frac{1}{\frac{N}{\sigma_n^2}+\frac{M\lambda^2}{\sigma_m^2}} \le \frac{\sigma_n^2}{N} \quad \textbf{(8)}$$

As a result, it can be seen that the channel estimation combining the CPICH, the DPCH, and the S-CCPCH is superior to both the channel estimation using only the CPICH and the channel estimation combining the CPICH and the DPCH.

3. Open-Loop Power Control

A modified OLPC and CLPC model is shown in Figure 2.

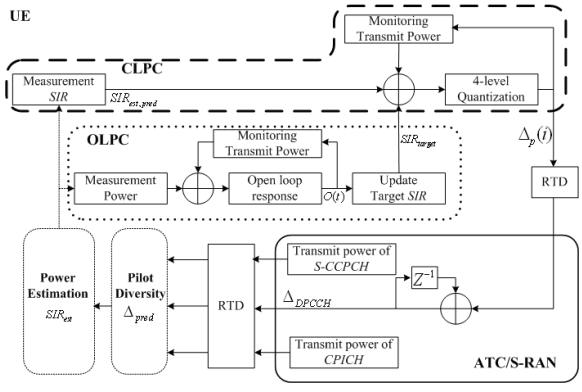


Figure 2. Modified OLPC and CLPC model.

In order to improve the accuracy of the estimation of SIR, we proposed a method to estimate the interference power, which will be presented as follows.

In Figure 3, n, k, l, T_b, and T_c denote n-th slot, k-th symbol, l-th resoluble multi-path, bit duration, and chip duration, respectively. Since the interference noise is Gaussian distributed, the variance of the interference can be found from the sum of the variances of the amplitude of the I channel and Q channel, as follows: [8]

$$I = E|R_I|^2 + E|R_Q|^2 \quad \textbf{(9)}$$

Desired signal S is achieved by calculating the summation of the S_l from the 1 to L tap RAKE receiver.

$$S = \sum_{l=0}^{L-1} S_l \quad \textbf{(10)}$$

According to Friis' free-space propagation-path-loss Formula [9], in order to apply OLPC, the average received power at the mobile station would be:

$$P_r = |E|^2/2\eta_0 = P_0[1/(4\pi d/\lambda)]^2 \quad \textbf{(11)}$$

where $P_0 = P_t G_t G_m$ and η_0, P_t, G_t, and G_m denote intrinsic impedance of free-space, transmitted power, gain of the transmitting antenna, and gain of the receiving antenna, respectively. Path loss and shadowing effects are regarded as slow fading in this work. The general open-loop response of the OLPC can be approximated as follows: [10]

$$O(t) = -\Delta P_{in}(1-\exp(-t/\tau)u(t) \quad \textbf{(12)}$$

in which ΔP_{in}, τ, and $O(t)$ are the step change in mean input power, the time constant of the open-loop response, and the output, respectively.

4. Closed-Loop Power Control

CLPC is a powerful tool to mitigate near-far problems in a DS-CDMA system over Rayleigh fading channels.

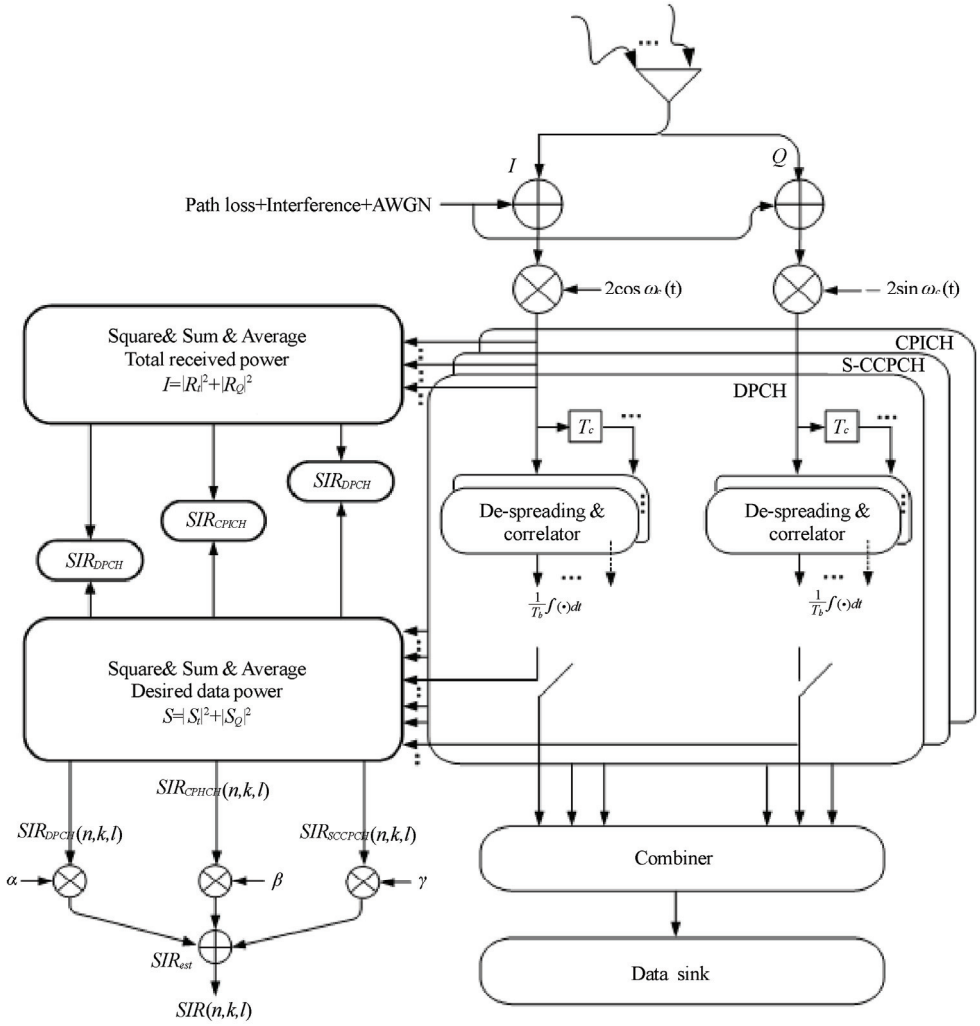


Figure 3. Block diagram of power estimation using pilot diversity.

Because of a significant difference in the RTD, there is serious performance degradation of the CLPC if the power control used for the terrestrial interface is employed as is. In order to reduce power control error, a delay compensation mechanism was selected in the ATC and satellite.

The transmitting power control (TPC) commands are generated as follows. Firstly, let us define power control error of $\Delta_{\varepsilon,c} = SIR_{est} - SIR_{target} + \Delta_{loop\ delay}$, where $\Delta_{loop\ delay}$ and SIR_{est} denote the prediction for the amount of SIR increment/decrement of the received downlink CPICH, S-CCPCH, and the estimated SIR of the received downlink DPCCH during the next time interval equal to the loop delay, respectively. Therefore, $\Delta_{loop\ delay}$ is added to SIR_{est} to result in the predicted SIR value of $SIR_{est,\ pred}$.

$$\Delta_{loop\ delay} = n \times \Delta_{pred} \quad \textbf{(13)}$$

where $n \times \Delta_{pred}$ is the increment (or decrement) of the estimated SIR of CPICH and S-CCPCH in dB during the last frame, and n is the nearest integer to (loop delay)/(frame length).

A four-level quantized power control step, Δ_p, is generated according to the region of Δ, as follows:

$$\begin{aligned}
&\text{if} \quad |\Delta_{\varepsilon,c}| < \varepsilon_T \quad \text{and} \quad \Delta_{\varepsilon,c} < 0, \quad \Delta_p(i) = \Delta_S \\
&\text{if} \quad |\Delta_{\varepsilon,c}| < \varepsilon_T \quad \text{and} \quad \Delta_{\varepsilon,c} > 0, \quad \Delta_p(i) = -\Delta_S \\
&\text{if} \quad |\Delta_{\varepsilon,c}| > \varepsilon_T \quad \text{and} \quad \Delta_{\varepsilon,c} < 0, \quad \Delta_p(i) = \Delta_L \\
&\text{if} \quad |\Delta_{\varepsilon,c}| > \varepsilon_T \quad \text{and} \quad \Delta_{\varepsilon,c} > 0, \quad \Delta_p(i) = -\Delta_L
\end{aligned}$$

in which Δ_S, Δ_L, and ε_T are a small power control step, a large power control step, and the error threshold, respectively. Because of the RTD in the GEO system, the satellite radio access network (S-RAN) can reflect $\Delta_p(i)$ at its transmission power after about 250*ms*, during which time there may be a considerable change in the SIR. The S-RAN adjusts the transmitting power of the downlink DPCCH with an amount of DPCCH using the two most recently received power control steps, $\Delta_p(i)$ and $\Delta_p(i\text{-}1)$, and this can be modeled as a simple FIR filter, as follows: [11]

$$\Delta_{DPCCH} = \Delta_p(i) - \alpha\Delta_p(i\text{-}1) \quad \textbf{(14)}$$

We can rewrite the above equation as follows:

$$\Delta_{DPCCH} = (1\text{-}\alpha)\Delta_p(i) + \alpha(\Delta_p(i) - \Delta_p(i\text{-}1)) \quad \textbf{(15)}$$

which means that Δ_{DPCCH} is determined not only by $\Delta_p(i)$ but also by the difference between $\Delta_p(i)$ and $\Delta_p(i\text{-}1)$ with weighting factors of $(1\text{-}\alpha)$ and α, respectively.

5. Simulation Results

A channel with only fast fading and a channel with path loss, slow fading, and fast fading were simulated to examine the performance of the CLPC with and without an OLPC. The simulation parameters are given in Table 1. We present the simulation results of only the proposed CLPC scheme (SCHEME-II), combining the proposed OLPC and proposed CLPC (SCHEME-I), and only the proposed OLPC (SCHEME-III) over GEO satellite or ATC environments, and we compare the performance of the various conventional- OLPC and CLPC algorithms. For conventional schemes, we used the terrestrial CLPC scheme in the WCDMA system and Gunnarsson's scheme in [12], and they are denoted in the figures as SCHEME-II with a dotted line and without SCHEME-II with a dotted line.

In our simulations, we consider a satellite system with a single beam and ignore the inter-spot interference. We assumed power control begins to work after 250ms due to propagation delay.

Figures 4 and 5 show the average transmitting power consumed at the transmitters of specific users according to mobile speed. It is observed that average UE transmitted power of all schemes is dependent of mobile speed. However, we can see that users with a combination of the modified OLPC and CLPC scheme consume less power. It is also seen that at low vehicle speeds (<40

Table 1. Simulation environment.

Parameter	Value	
Carrier frequency (f_c)	2170 MHz	
Power control sample interval (T_d)	UE serving from GEO satellite	10 *ms*
	UE serving from ATC	6.667E-4 *ms*
Frame length	10 *ms*	
Round trip delay	GEO satellite	250 *ms*
	ATC	< slot duration (≈6.667E-4 *ms*)
Processing gain	256 (≈24 dB)	
Transmit frame	UE serving from GEO satellite	70,000 frames
	UE serving from ATC	60,000 slots
Small step size	1 dB	
Large step size	2 dB	
Fading model	Clarke's model (Classical Doppler spectrum)	
Target SIR	5 dB	
Desired received power	-140 dBW (≅ -110 dBm)	
Rician K-factor	UE serving from GEO satellite	5 dB
	UE serving from ATC	-inf
Power command error probability	0 ~ 0.15	
Interference plus noise power	-123 dBm	
Interference variance	6 dB	
Path loss variance	8 dB	
Maximum transmitting power	GEO satellite	41.8 dBW
	ATC	28 dBm
Minimum transmitting power	GEO satellite	-2.9 dBW
	ATC	-61 dBm
Mobile speed	0 ~ 98 *km/h*	

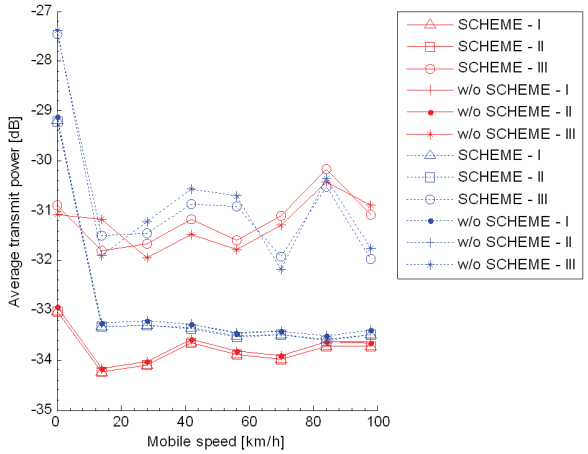


Figure 4. Average transmitting power of UE serving from ATC according to mobile velocity.

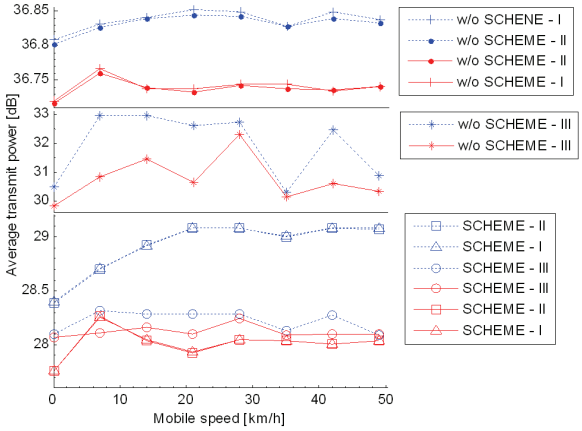


Figure 5. Average transmitting power of UE serving from GEO satellite according to mobile velocity.

km/h), combining modified OLPC and CLPC (SCHEME-I shown with a solid line) is very effective. This is because SCHEME-I compensates slow fading and path loss by monitoring transmitting power of UE and simultaneously archives diversity gain by using efficient channel estimation algorithms.

Figures 6 and 7 show the average received power consumed at the transmitters of specific users according to mobile speed. We can see that the received power of users with a combination of the modified OLPC and CLPC scheme using pilot diversity is settled compared to the other scheme. This highlights the importance of monitoring transmitting power equipment applying for a CDMA-based system. With a RAKE receiver, the dynamic range of the received power decreased as the monitoring equipment decreased.

Figures 8 and 9 show the probability density function of the received SIRs for a mobile speed of 98*km/h* having a probability of power control command error of 0. Intuitively, we turn out that the SCHEME-I using pilot

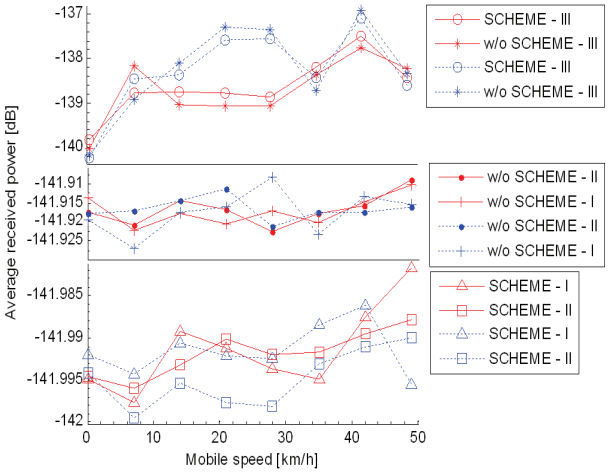


Figure 6. Average received power of UE serving from ATC according to mobile velocity.

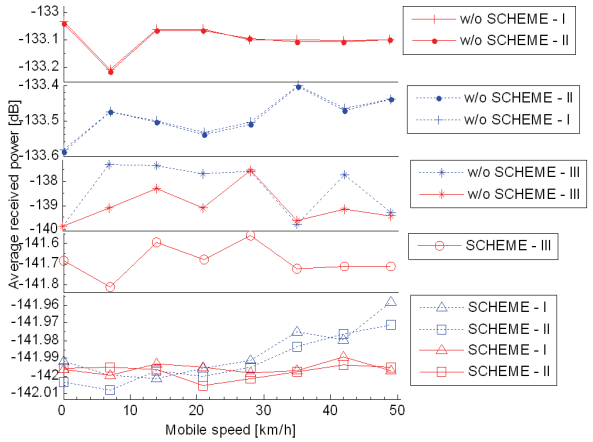


Figure 7. Average received power of UE serving from GEO satellite according to mobile velocity.

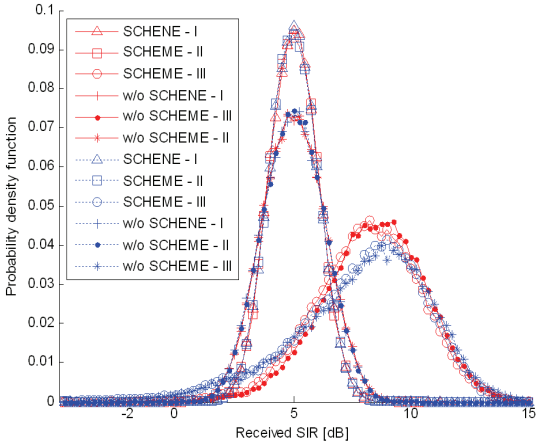


Figure 8. Probability density function of received SIRs of UE serving from ATC: K= -inf and V=98*km/h*.

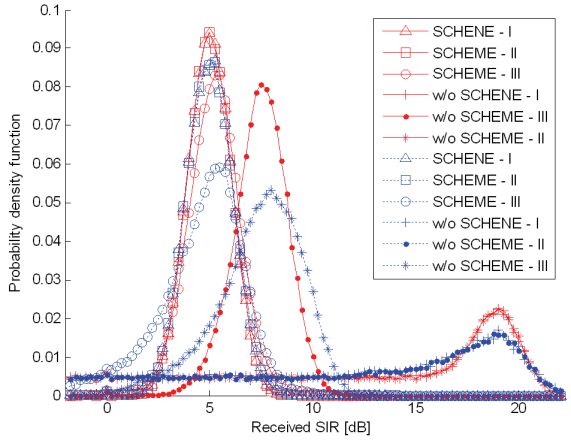


Figure 9. Probability density function of received SIRs of UE serving from GEO satellite: K=5 dB and V=98 *km*/*h*.

diversity has a larger improvement, as confirmed in the simulation results.

6. Conclusions

Conventional channel estimation methods incur many errors in the case of deep fading. In contrast, the proposed channel estimation method using the S-CCPCH to the conventional methods can obtain an improved pilot diversity gain by performing channel estimation using other channels when the first channel does not reach a required level of a received signal. Thus, it is possible to implement an ideal maximum ratio combining method in a RAKE receiver. The channel estimation method described in this paper provides a more improved performance than channel estimation in a receiver of a terminal having conventional pilot symbols of a CPICH or a DPCCH by combining pilot symbols of a CPICH, a DPCCH, and a S-CCPCH, and estimating a channel. In this paper, we have presented satellite access technologies for a future mobile system. We suggested desirable modifications for application to the 4G system. Combining modified CLPC and OLPC with delay compensation algorithms and monitoring equipment proved to provide a good performance in a MSS/ATC hybrid system. In addition, to increase the performance and to keep commonalities between terrestrial standards, more advanced transmission technologies, including multi-carrier transmission, interference cancellation, and highly efficient modulation and coding should be investigated in more detail.

7. References

[1] I. Philipopoulos, S. Panagiotarakis, and A. Yanelli-coralli, "The role of S-UMTS in future 3G markets," ist-2000-25030 SATIN project, SUMTS, P-specific requirement, deliverable No. 2, April 2002.

[2] G. M. Parson and R. Singh, "An ATC primer: The future of communications," MSV, 2006.

[3] Report and order and notice of proposed rulemaking, fcc 03−15, Flexibility for Delivery of Communications by Mobile Satellite Service Providers in the 2 ghz band, the L-Band, and the 1.6/2.4 Bands, IB, Adopted: January 29, 2003, Released: February 10, 2003.

[4] S. Dutta and D. Karabinis, "Systems and methods for handover between space based and terrestrial radioterminal communications, and for monitoring terrestrially reused satellite frequencies at a radioterminal to reduce potential interference," US Patent No. 6879829 b2, April 12, 2005.

[5] T. Luo and Y. C. Ko, "Pilot diversity channel estimation in power-controlled CDMA systems," IEEE Transactions on Vehicular Technology, Vol. 53, No. 2, pp. 559−563, March 2004.

[6] S. M. Kay, "Fundamentals of statistical signal processing: Estimation theory," 2nd Edition, Englewood Cliffs, NJ: Prentice-Hall, 1993.

[7] H. V. Khuong and H. Y. Kong, "BER performance of cooperative transmission for the uplink of TDD-CDMA systems," ETRI Journal, Vol. 28, No. 1, pp. 17−30, February 2006.

[8] C. -C. Lee and R. Steele, "Closed-loop power control in CDMA systems," IEE Proceedings of Communications, Vol. 143, No. 4, pp. 231−239, August 1996.

[9] W. C. Y. Lee, "Mobile communications engineering," 2nd Edition, Mcgraw-Hill, 1998.

[10] S. Choe, "An analytical framework for imperfect DS-CDMA closed-loop power control over flat fading," ETRI Journal, Vol. 27, No. 6, pp. 810−813, December 2005.

[11] K. Lim, K. Choi, K. Kang, S. Kim, and H. J. Lee, "A satellite radio interface for IMT-2000," ETRI Journal, Vol. 24, No. 6, pp. 415−428, December 2002.

[12] F. Gunnarsson, F. Gustafsson, and J. Blom, "Dynamical effects of time delays and time delay compensation in power controlled DS-CDMA," in IEEE Journal of Selected Areas on Communications, Vol. 19, No. 1, pp. 141−151, January 2001.

Int. J. Communications, Network and System Sciences, 2009, 3, 169-247
doi:10.4236/ijcns.2009.23021 Published Online June 2009 (http://www.SciRP.org/journal/ijcns/).

Fast and Noniterative Scheduling in Input-Queued Switches

Kevin F. CHEN, Edwin H.-M. SHA, S. Q. ZHENG
Department of Computer Science, University of Texas at Dallas, Richardson, Texas, USA
Email: {*kchen, edsha, sizheng*}*@utdallas.edu*
Received April 22, 2009; *revised May* 22, 2009; *accepted May* 25, 2009

ABSTRACT

Most high-end switches use an input-queued or a combined input- and output-queued architecture. The switch fabrics of these architectures commonly use an iterative scheduling system such as iSLIP. Iterative schedulers are not very scalable and can be slow. We propose a new scheduling algorithm that finds a maximum matching of a modified I/O mapping graph in a single iteration (hence noniterative). Analytically and experimentally, we show that it provides full throughput and incurs very low delay; it is fair and of low complexity; and it outperforms traditional iterative schedulers. We also propose two switch architectures suited for this scheduling scheme and analyze their hardware implementations. The arbiter circuit is simple, implementing only a FIFO queue. Only half as many arbiters for an iterative scheme are needed. The arbiters operate in complete parallel. They work for both architectures and make the hardware implementations simple. The first architecture uses conventional queuing structure and crossbar. The second one uses separate memories for each queue at an input port and a special crossbar. This crossbar is simple and also has a reduced diameter and distributed structure. We also show that the architectures have good scalability and require almost no speedup.

Keywords: Switch Architecture, Switch Fabric, Fabric Scheduling, SRA

1. Introduction

There has recently been renewed interest in building new switch fabric architectures as line rates go from 10 Gbps to 1 Tbps and beyond. Existing architectures are not very scalable. As memory technology evolves, switching techniques that would otherwise be considered unworkable may now be implemented. New switch fabrics can and should now be fast and highly scalable. In this paper, we propose and analyze two such novel fabric architectures.

By queuing structure, there are input-queued (IQ) switch, output-queued (OQ) switch, and combined input- and output-queued (CIOQ) switch. An OQ switch buffers cells at the output ports. OQ switches guarantee 100% throughput since the outputs never idle as long as there are packets to send. OQ switches are hard to implement. An $N \times N$ OQ switch must operate N times faster than the line rate. Memory technology cannot meet that kind of high-speed requirement. Therefore, IQ and CIOQ switches have gained widespread attention and adoption. The most common architecture is the CIOQ switch in which buffering occurs both at the input and at the output. Output queues are for traffic scheduling which provides fine-tuned service support. Both IQ and CIOQ switches use virtual output queuing by which each input maintains a separate queue for cells destined for each output or of a flow of a certain service requirement. Such a queue is called a *virtual output queue* (VOQ). Virtual output queuing removes head-of-line (HOL) blocking that can severely limit the throughput when only a FIFO queue is used for all the packets at each input.

It is customary to use a crossbar to interconnect the input and output ports due to its simplicity and nonblocking property. A crossbar can either have memory or have no memory at its crosspoints. Our work is for IQ switch architectures using unbuffered crossbars. Crossbar access by the input cells has to be arbitrated by the fabric scheduler. Traffic scheduling manipulates the cells further to meet rate and delay requirements of various services. Fabric and traffic schedulers can be considered as separate identities. They must work in coordination to maximize datapath utilization.

The IQ fabric scheduling problem is key for building efficient switches. Many algorithms have been proposed for scheduling an IQ switch to obtain high throughput. The algorithms all find a matching between the inputs and outputs, but they were derived with different techniques. Under the matching paradigm, the scheduler matches an input with an output and finds the maximal number of those pairs in a time slot. This usually takes a few iterations for one time slot. Numerous algorithms work in this iterative way and are hereof called iterative algorithms. Those pairs are found globally and do not conflict one another. The scheduler uses the information on the states of the input queues and the output readiness to make the matching decision.

The cell scheduling problem for switches is conventionally modeled as bipartite matching over a graph G as follows. In each time slot, G is constructed such that there is an edge from each input port to each output port. The ports are represented as vertices or nodes in G. This implies that at most one cell from a VOQ at an input port can be sent to its destined output port, which corresponds to one edge in G being selected. The bipartite matching over G is the process to find a set of edges such that their vertices do not overlap. A *maximal matching* contains the largest number of edges possible to be selected in a time slot in the number of iterations set according to an iterative algorithm. A *maximum matching* selects the maximum number of edges, i.e. every input port is matched to a distinct output port if the input port has a queued cell that is going to a distinct output port.

Our scheduling algorithm, called SRA, is *noniterative*. Matching is done in a single iteration during a time slot and its efficiency is much higher. SRA runs in $O(1)$ time and always finds a *maximum* matching, although the matching is done over a graph G' modified after G as to be detailed in Section 4. In G', the VOQs at the input ports along with the output ports form the vertices instead of just the input ports plus the output ports as in G. Matching still aims to find the largest number of edges of G' that do not overlap and is done in a single iteration. The basic idea is to allow each input port to send up to multiple cells each from a different VOQ to a different output port in a time slot. Arbitration is implemented by a single round-robin arbiter for each output port. The SRA algorithm is different from the iterative algorithms in terms of the arbitration process. In SRA, an arbiter, consisting of a FIFO queue, is maintained for each output port. The arbiter selects the input port corresponding to the first queued cell. Hence, the arbitration done in iSLIP and other iterative algorithms is not needed.

For hardware implementation, we propose two architectures to support SRA. Both architectures rely on the arbiter construct which is just a FIFO queue. They differ in queuing structure and the crossbar each uses. The first one uses conventional queuing structure and crossbar. The second uses separate memories for VOQs at each input port and a special crossbar.

In this paper, we show that the SRA algorithm is workable, simple, fast, and scalable. We analyze SRA's characteristics. Simulation results also demonstrate that SRA is far more efficient than the popular matching algorithms. We show that the SRA architectures are simple, fast, and effective. We also discuss the hardware implementations. The architectures could be used in switches and routers. No other similar architectures have been proposed. We hope our architectures provide viable alternatives for designing next-generation switch fabrics.

Note that the second architecture assumes a multiple-multiplexer structure and requires VOQs be buffered in separate memories and connected to the custom crossbar differently from the first architecture. Therefore, throughout the paper, we use the term "*input-queued switch*" to refer only to the fact that traffic is buffered at the input ports in VOQs in such a switch, regardless of the memory makeup for the VOQs and the crossbar structure.

The rest of the paper is organized as follows. In Section 2, we review related work including the very latest. We discuss the iterative algorithms in detail. Section 3 gives an overview of the SRA fabric, in comparison with a conventional iterative one. Section 4 contains the SRA algorithm and its complexity analysis. Section 5 contains the analytical results of the SRA algorithm. Section 6 contains the simulation results that show SRA's performance in handling various traffic types, scalability, and cell blocking at input ports. Section 7 shows the hardware aspects of SRA. It covers queuing structure, crossbar, arbiter, and architecture scalability. It also includes a comparison of the architectures to the Knockout Switch. Section 8 concludes the paper, where we highlight the achievements and innovations of this research work.

2. Related Work

The field of IQ switch scheduling boasts of an extensive literature. Many algorithms exist, derived with different techniques. Some of them are of more theoretical import, whereas others are more oriented to implementation. Here we review a representation of the works.

In graph-theoretic terms, the cell scheduling problem for switches can be modeled as a bipartite matching problem as follows. Let I_i and O_j denote input port i and output port j respectively. Let $VOQ_{i,j}$ denote the VOQ at I_i holding cells destined for O_j, and $VOQ_{i,j}(t)$ the length of $VOQ_{i,j}$ at time slot t. In each time slot t, we construct a bipartite graph $G(V,E)$ such that $V = V_1 \cup V_2$, $V_1 = \{I_i | 1 \le i \le N\}$, $V_2 = \{O_j | 1 \le j \le N\}$, and $E = \{(I_i,O_j) | VOQ_{i,j}(t) > 0\}$. Graph G is called an *I/O mapping graph*. A *matching*

is defined as the set $M \subseteq E$ such that no two edges in M are incident to the same node in V. A maximum matching is one with the maximum number of edges, whereas a maximal matching is one that is not contained in any other matching found in an iteration when a matching is done iteratively in a prefixed number of iterations in a time slot.

Work by McKeown *et al*. [1,2] shows that a 100% throughput can be achieved by using a longest queue first (LQF) or an oldest cell first (OCF) scheduling policy for independent identically distributed (i.i.d.) Bernoulli traffic with uniform or non-uniform destinations. LQF and OCF are both maximum weight matching algorithms. McKeown *et al*. proved the result using a linear programming argument and quadratic Lyapunov function. In related work, Mekkittikul and McKeown [3] used the longest port first (LPF) scheduling policy to obtain full throughput. Those scheduling policies appear to be too complex for hardware implementation due to the inherent $O(N^3 \log N)$ complexity of maximum weight matching.

Iterative techniques can be used to find the bipartite matching. Example iterative algorithms include iSLIP [4,5], 2DRR [6], and WRR [7]. These algorithms all use round-robin; the first two are unweighted and the last one is weighted using idling hierarchical round-robin. As discussed in [2], these solutions can get a throughput of more than 90% for uniform traffic but will fare worse when traffic is nonuniform. These algorithms have an $O(N^2)$ worst-case time complexity. Although they can converge in $O(\log N)$ iterations, they tend to incur long delay and have poor scalability.

Yang and Zheng [8] proposed an iterative scheduler that uses space-division multiplexing expansion and grouped inputs/outputs to realize speedup while switch fabric and memory operate at line rate. They formulated packet scheduling as a maximum bipartite k-matching problem. They proved by a fluid model method that full throughput can be obtained when the expansion factor is 2, with only mild restrictions on traffic arrivals. They also proposed the kFRR scheduling algorithm to implement the multiplexing and grouping for full throughput. Since their scheme is iterative, it is prone to long delay and low scalability.

Chao *et al*. proposed and studied another matching architecture called dual round-robin matching (DRRM) [9−11]. DRRM is similar to iSLIP and does request, grant, and accept slightly differently. It uses a bit-sliced crossbar, and a token-tunneling technique to arbitrate contending cells. DRRM can support an aggregate bandwidth of over 1 Tbps using CMOS technology. DRRM can sustain 100% throughput for uniform traffic. It is slightly slower than iSLIP under certain types of non-uniform traffic.

Some other matching techniques guarantee a 100% throughput for both uniform and nonuniform traffic. Chang *et al*. [12,13] developed a scheduling algorithm based on the finding of Birkhoff and von Neumann that a doubly stochastic matrix can be decomposed into a weighted sum of permutation matrices. This algorithm works for traffic of both one priority and two priorities. For the latter, scheduling is optimized for fairness as well as efficiency. In all cases, throughput reaches 100%. Their work is important theoretically, but the switch appears to be too complex (of $O(N^{4.5})$) to be implemented in hardware. In related work, Chang *et al*. [14] generalized the Pollaczek-Khinchin formula to calculate the throughput of input-queued switches. This work is based on an abstraction of input-queued switches and thus offers limited insights into the actual workings of those switches.

Using fluid model techniques, Dai and Prabhakar [15] extended the result of McKeown *et al*. [1,2]. Dai and Prabhakar proved that one can get 100% throughput using a maximum weight matching algorithm in an IQ switch subject to arbitrarily distributed input traffic as long as the traffic obeys the strong law of large numbers and does not oversubscribe any input or output. Dai and Prabahakar's work is theoretical in that it is not scheduling algorithm specific.

When a scheduling algorithm sustains 100% throughput, the IQ switch can emulate an OQ switch. For instance, 2DRR achieves the same saturation throughput as output queuing [6]. There has also been attempt to explicitly emulate an OQ switch by an IQ switch. The work of Gourgy and Szymanski [16] shows that the emulation can be done by tracking the behavior of an ideal OQ switch and matching it to the IQ switch by metrics such as "lag". Based on those metrics, Gourgy and Szymanski designed several algorithms that perform as well as other existing ones in terms of fairness and complexity. OQ emulation studies are theoretical and offer no practical solutions to IQ switching.

In particular, the iSLIP class of iterative matching algorithms, which are designed for finding maximal matchings, is the most widely used in commercial IQ and CIOQ switches. High-end routers of late are Cisco CRS-1 and Juniper TX Matrix. Both are for lumping together multiple smaller routers to form a single larger router. CRS-1 can interconnect up to 72 boxes for a total capacity of 92 Tbps. TX Matrix connects up to 4 T-640 routers for a capacity of up to 2.56 Tbps. While CRS-1 uses a 3-stage Benes switch fabric, TX Matrix uses a standard iterative switch fabric. However, the constituent smaller routers for both CRS-1 and TX Matrix all use a standard iterative switch fabric.

Of latest research work is the π-RGA iterative algorithm proposed by Mneimneh in [17]. This algorithm does request, grant, and accept in every iteration of a time slot. If a maximal matching is found in the first iteration, then the switching is done in one iteration for the time slot. Otherwise, the result is carried over for match-

ing calculation in the next time slot. As any other iterative algorithm, π-RGA needs a speedup of 2. The paper shows that for certain uniform and non-uniform traffic patterns, one iteration is enough to achieve maximal matching, which thus amounts to shortened time slots and increased speed.

However, π-RGA does not appear to have overcome the efficiency hindrances as with other iterative algorithms. In both throughput and delay, π-RGA apparently performs worse than iSLIP under uniform traffic.

A variant of iSLIP itself is DSRR. Matching algorithms also include randomized ones. A precursory randomized matching algorithm is PIM. Later ones include those proposed in [18,19]. As we are to compare SRA to PIM, iSLIP, and DSRR in simulations, here we review these three algorithms.

The parallel iterative matching (PIM) algorithm of Anderson *et al*. [20] is the first randomized iterative algorithm. With enhancements, PIM can ensure certain fairness and throughput. By PIM, each input sends a bid to all the outputs for which it has a buffered cell. An output randomly selects a bid to grant access and notifies each input whether its request was granted. If an input receives any grants, it chooses one to accept and notifies the output. Randomization is relatively expensive. PIM performs poorly when run in one iteration and finds maximal matching in $O(\log N)$ iterations.

The iSLIP algorithm works in iterations each consisting of three steps as described in [4]:

Step 1: Request. Each unmatched input sends a request to every output for which it has a queued cell.

Step 2: Grant. If an unmatched output receives any requests, it chooses the one that a pointer g_i points to in a round-robin schedule in descending priority order. The output notifies each input if its request was granted. Then g_i is advanced (modulo N) one location beyond the granted input if the grant is accepted in Step 3 of the first iteration.

Step 3: Accept. If an unmatched input receives a grant, it accepts the one that a pointer a_i points to in a round-robin schedule in descending priority order. Then a_i is advanced (modulo N) one location beyond the accepted output.

The double static round-robin (DSRR) algorithm [21] is an enhancement to iSLIP and works similarly to iSLIP. It also has request, grant, and accept steps and differs from the iSLIP in the following ways: 1) The *g* pointers at the outputs are set to some initial pattern such that there is no duplication. 2) The *a* pointers at the inputs and the *g* pointers at the outputs are set to the same pattern. 3) In Step 2, g_i is advanced one location no matter the grant is to be accepted or not by the input, i.e., it moves down a location in each iteration. 4) In Step 3, a_i is advanced one location no matter there is an accept or not.

The initialization and pointer assignment peculiar to the DSRR makes a big difference in improving the performance of the iSLIP algorithm as we will see in Section 6.

Iterative matching algorithms like the three above appear to have some drawbacks. First, they are not scalable. They are very sensitive to the problem size. Their performance degrades considerably when N becomes large. Second, they require the fast feedback of the states of both the input and the output ports. As a result, the scheduler is centralized and has to be placed in a central location. This not only impedes scalability, but also worsens the fault-tolerance of the system.

3. Switch Fabric

Here we give a general description of the SRA switch fabrics. We will describe the detailed hardware structures of their components in Section 7. Figure 1(a) shows the switch fabric architecture of an IQ switch according to SRA. For easy comparison and review, Figure 1(b) shows the switch fabric used for a typical iterative scheme.

In both scenarios, the switch consists of N input ports, an $N\times N$ memoryless crossbar interconnect, and N output ports. Ingress traffic in cells is queued at the input ports. There are N VOQs at each input port, one for each output port.

Figure 1(a) is a drawing that applies to both architectures that we are proposing in this paper. A typical feature of the architectures is that the output arbiters are placed in a distributed manner. These N arbiters are independent of each other. Each arbiter implements a single FIFO queue. Of course, the arbiters can be put in a single chip in hardware.

Figure 1(b) represents the conventional input-queued architecture initially studied by such works as [1,5,20]. In this architecture, the output arbiters (actually 2 layers of them) must be placed in a single-chip centralized scheduler. In Figure 1(b), only the scheduler is shown. The circuitries of the arbiters in the two layers are different as are they in the iterative and noniterative types of fabrics. Arbiters will be discussed in detail in Section 7.

The switch fabric excludes other functionalities that may reside in the port cards such as IP lookup, segmentation of cells in input cards, and demultiplexing and reassembly of cells in output cards. The switch fabric operates in a timing reference of its own. If the frequency of the fabric's timing is S times faster than the frequency of the link feed, we say that the *speedup* of the fabric is S. In this paper, we want to discern how much speedup is needed for full throughput. We consider the cells as of equal size. Cells are easier to synchronize and hence simplify scheduling.

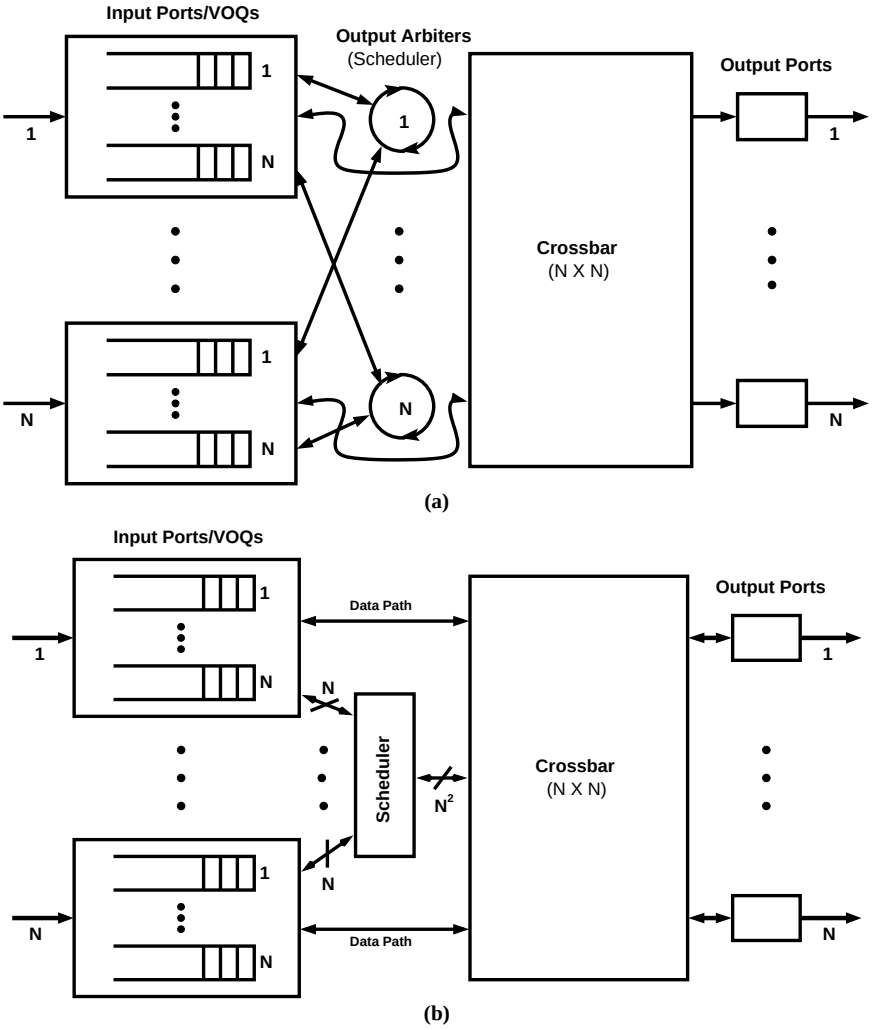


Figure 1. IQ switch fabrics. (a) An SRA architecture. It has distributed arbiters. (b) An iterative architecture. This typically has a centralized scheduler.

Traffic feeds up to one cell into the fabric at each input port every time slot of the line timing. If traffic exits the switch one cell at each output port every line timing slot if the output port is not empty, we say that the switch achieves full throughput (100%) [1].

In an SRA architecture (Figure 1(a)), for each output port, there is an arbiter that keeps track of the input ports having packets destined to it and their order of arrival. Those arbiters can be placed near the input ports to keep track of VOQ status easily. At each time slot, the arbiter grants a send to the input whose cell arrived the earliest. There is no memory existing for arbitration at an output port. Neither is there *backpressure* applied from the output ports. Backpressure is needed by many CIOQ architectures. Backpressure involves sending signals toward input ports when certain congestion thresholds are crossed in the queues at the output ports. Backpressure exerts flow control. No backpressure usage is a distinct feature of our architecture.

A crossbar is nonblocking and has a unified self-routing algorithm to switch cells at its crosspoints as needed. Fabric scheduling is to arbitrate how the cells access the crossbar in an orderly fashion so as to maximize the crossbar utilization in a time slot.

Our first architecture uses the same crossbar as any iterative switch fabric. Our second architecture has a crossbar that is not entirely the same. The architecture is thereof called "*multiple-multiplexer switch*". In addition, the VOQs at an input port are operated and connected to the crossbar differently between the two architectures. Architectural details will be discussed in Section 7.

4. The SRA Algorithm

SRA stands for *single round-robin arbitration*. Each output port uses a round-robin arbiter to select the input port to send in a time slot. PIM, iSLIP, and DSRR all have round-robin arbiters at both the input and output

ports. SRA has several significant advantages. In this section, we describe the algorithm and then analyze its properties that make it simple, fast, and scalable. The IQ switch architecture supporting SRA is illustrated in Figure 1(a).

4.1. Description

Let $G'(V', E')$ denote the bipartite graph defined as follows: $V' = V'_1 \cup V'_2$, $V'_1 = \{VOQ_{i,j} | 1 \le i, j \le N\}$, $V'_2 = \{O_j | 1 \le j \le N\}$, and $E' = \{(VOQ_{i,j}, O_j) \mid VOQ_{i,j}(t) > 0, 1 \le i, j \le N\}$. We call G' the *modified I/O mapping graph*. The SRA algorithm is designed to find a *maximum* matching in G'. Note that G' is different from G defined in Section 1 and the matching is done differently over G' than over G.

SRA is not iterative. It selects a set of up to N cells to send to up to N outputs in a single time slot. Each cell goes to a different output. In theory, these cells can come from one, N, or any other number of inputs. That is, each input can send up to N cells in a time slot. This is where SRA differs from existing algorithms which allow each input to send at most only one cell out in each time slot. Apparently this increases efficiency since there is little reason not to let the input send more than one cell in a time slot when other inputs have no cells to send.

The pseudocode of the SRA algorithm is shown in Algorithm 1. In the pseudocode, the notation qstatus[ip] represents whether the VOQs at input port *ip* is empty or not. The notation IsEmpty(q[op]) represents whether the queue at the arbiter for output port *op* is empty or not. The notation Enqueue(ip, q[op]) means to add the number of input port *ip* to the tail of the queue at the arbiter for output port *op*, whereas Dequeue(q[op]) means to remove the head element from the queue at the arbiter for output port *op*.

The SRA algorithm works as follows:

(1) At the outputs. Each output arbiter maintains a again into the tail of the status queue, else the status element for that input is gone.

(2) At the inputs. Upon receiving a grant, the input checks if the corresponding VOQ is to become empty if the cell has been sent. If yes, it sends a status signal to the output arbiter indicating the VOQ is to be empty, so the output arbiter will not keep an element for this input in its FIFO queue again. Then the input port sends a cell to the crossbar with the designated output information. The input sends status information about any of its VOQs to the corresponding output (arbiter) only when the VOQ changes from being empty to having a cell arrived and from having cells to becoming empty.

Algorithm 1 The SRA Algorithm

Arbiter at output port *op*:

Initialization:

```
1: for ip = 0 to N − 1 do
2:     qstatus[ip] ← 0
3: end for
```

Arbitration:

```
1: // Loop forever
2: // Each iteration represents 1 time slot
3: loop
4:       // Check for newly backlogged VOQs
5:       for ip = 0 to N − 1 do
6:            if VOQ[ip, op] is not empty and qstatus[ip] = 0 then
7:                 Enqueue(ip, q[op])
8:                 qstatus[ip] ← 1
9:            end if
10:      end for
11:      if !IsEmpty(q[op]) then
12:           Get ip of head element of q[op]
13:           Send a grant for op to input port ip
14:           Dequeue(q[op])
15:           if IsEmpty(q[op]) then
16:                qstatus[ip] ← 0
17:           else
18:                Enqueue(ip, q[op])
19:           end if
20:      end if
21: end loop
```


4.2. Complexity

In each time slot (a cell time), a fabric scheduler must perform a matching and synchronously switch on and off the crosspoints of the crossbar to send up to N packets out. High line rates ever stringently require a scheduling action to be prompt. Since SRA finds a maximum matching in a single iteration, it is capable of fast scheduling actions. Its time complexity is only $O(1)$ since all the operations in the algorithm take constant time to finish. On the other hand, an iterative algorithm doing multiple iterations would be too slow to support high line rates.

SRA needs fewer messages to operate than an iterative matching algorithm. The number of exchanged messages a port has to process is equal to the product of N and the number of service levels. Consider the case of best-effort service only. For each matching, an output arbiter sends only one grant message, an input can send up to N status messages out if the status of all N VOQs changes. Over the entire fabric, there are at most N^2 messages exchanged between the inputs and the outputs. Unlike SRA, iSLIP needs N^2 requests, N^2 grant notifications, and N accepts for each iteration. For iSLIP to converge, at least log$N$ iterations are needed. Thus iSLIP needs a total of $(2N^2+N)$ log$N$ messages. PIM and DSRR each need about the same number of messages as iSLIP.

It may be more accurate to compare the information bits exchanged than analyzing the amounts of messages sent by SRA and iSLIP. For SRA, N grants will be sent from output ports to input ports. Hence a total of N^2+N bits are exchanged. For iSLIP, only the information bits from the input ports to the scheduler and back to the input ports should be considered. That will be also N^2+N bits, the same as for the other iterative algorithms. Of course SRA only does one iteration but the others need log$N$ iterations.

In matching over G', in a time slot, each input can be mapped to multiple outputs, but each output is mapped to one input or is not mapped when there is no traffic destined to it. Since only the arbiters decide and send grants, this matching can be regarded as output constrained (and input unconstrained). Yet iterative matching over G involves both input and output ports for actions of request, accept, and grant, and is thus both input and output constrained.

5. Performance Analysis

We first show that SRA matches the maximum number of inputs to the maximum number of outputs in each time slot. We then show that SRA sustains 100% throughput and that it is fair.

Theorem 5.1. *SRA always finds the maximum matching in G′.*

Proof. Let $k(t)$ be the number of nodes in V'_2 of G' with non-zero degree at time slot t, and M' be any matching of G' at time slot t. By the definition of M', $k(t) \le |M'|$. SRA guarantees a matching M^* such that its size is exactly $k(t)$.

Alternatively, because there are N independent (disjoint) subgraphs in G' each of which corresponds to and matches a particular output port, SRA guarantees to find a maximum matching in each time slot.

That an input port can send m ($1 \le m \le N$) cells and the input ports altogether are allowed to send no more than N cells in a time slot is called the *free rule*. There exist analytical studies of throughput under the free rule [22−25]. These studies assume that traffic arrival is i.i.d. Bernoulli with uniformly distributed destinations and found that throughput can be 100% if the load does not exceed 1.0.

Since SRA is a free-rule scheduling policy, so in theory it should sustain 100% throughput. In fact, we can show that SRA does so irrespectively of the traffic arrival patterns.

Theorem 5.2. *SRA sustains* 100% *throughput.*

Proof. Each input port keeps up to N VOQs. Assume that all the VOQs destined to the same output port j at the input ports bid for transmission in a time slot t. Let γ be the throughput of these VOQs. Note that γ is equal to the overall throughput. Let N_j be the number of HOL packets at all the VOQs destined to output j in t. The total number of HOL packets blocked at the VOQs in t is

$$N_b = N_j - \varepsilon(N_j) \quad \textbf{(1)}$$

where $\varepsilon(N_j)=min(1,N_j)$. More specifically, function ε is defined as

$$\varepsilon(x) = \begin{cases} 1, & x \ge 1 \\ 0, & x = 0. \end{cases}$$

In each time slot SRA sends up to N cells out and there can be up to N cells arriving. Thus $E[\varepsilon(N_j)]=\gamma$ in steady state. Taking expectation of (1) gives

$$\gamma = E[N_j] - E[N_b] \quad \textbf{(2)}$$

Let M be the total number of unblocked VOQs in t. Then

$$M = N - N_b \quad \textbf{(3)}$$

By flow conservation, we have

$$E[M]\rho = \gamma \quad \textbf{(4)}$$

where ρ is the probability that one of the M unblocked VOQs gets a new cell to arrive in t. Taking expectation on both sides of (3) and using (4), we obtain

$$E[N_b] = N - \gamma/\rho \quad \textbf{(5)}$$

Let N'_j be the number of HOL packets with destination j in time slot $t + 1$. Let A_j be the number of HOL packets

with destination j arrived at the M sending VOQs. We have the following dynamic equation:

$$N'_j=N_j-\varepsilon(N_j)+A_j \quad \textbf{(6)}$$

Then we can obtain the following mean-value equation as in Appendix A of [22]:

$$E[N_j]=E[A_j]+\frac{E[A_j(A_j-1)]}{2(1-E[A_j])}. \quad \textbf{(7)}$$

For large values of N, A_j can be approximated by a Poisson distributed random variable. This step follows the proof by Karol *et al*. in Appendix A of [26] which shows that as $N \to \infty$, the steady-state number of HOL packets at the VOQs destined for an output in each time slot becomes Poisson. We can obtain

$$E[A_j]=E[\varepsilon(N_j)]=\gamma \quad \textbf{(8)}$$

$$E[A_j(A_j-1)]=\gamma^2 \quad \textbf{(9)}$$

Using (8) and (9) and substituting (5) and (7) into (2), we obtain the throughput formula by setting ρ=1:

$$\gamma=1+N-\sqrt{1+N^2} \quad \textbf{(10)}$$

Equation (10) implies that when $N\to\infty$, throughput is 1. Actually when N is finite, (10) still gives a very close approximation to the optimum value. For instance, when $N=32$, $\gamma=0.992$.

Theorem 5.3. *SRA is fair.*

Proof. We consider a loading scenario more general than uniform i.i.d. Bernoulli. Assume that the loading rate at each input is the same λ and is admissible. Admissible traffic does not oversubscribe any input or output port. Let λ_{ij} be the loading rate of traffic going from input i to output j. We have

$$\lambda_{ij}=\delta\lambda$$

where δ is the fraction of λ for traffic going from input port i to output port j. Let μ_{ij} be the portion of the service rate μ at output port j that serves the traffic of λ_{ij}. Then

$$\mu_{ij}=\delta\mu$$

By the admissible rule, the rate of traffic arriving at port j from all the input ports combined does not exceed μ. Also, the output arbiter works in a round-robin fashion serving each input port that has a cell in turn in each time slot. By Theorem 5.2, traffic of λ_{ij} will receive its fair share of service. Hence we have the above equation. Also, in steady state, $\lambda=\mu$. Thus we obtain

$$\lambda_{ij}=\mu_{ij}$$

The above equation holds for any given time period. In a time period of m time slots, an arbiter j ensures $\lambda_{ij}m$ time slots granted to input port i, since each arbiter works round-robin on the status FIFO queue and any backlogged input port is re-enqueued after it has got a turn to send. Therefore, SRA is able to allocate the available service rate to all input-output pairs in proportion to their offered loads in any given time period.

Note that SRA is fair per VOQ or fabric-wide. Subsequently, per-port fairness is also guaranteed.

6. Performance Evaluation

We simulated SRA against PIM, iSLIP, and DSRR in various traffic conditions. Performance metrics are cell delay and throughput vs. offered load. Offered load is the number of cells per time slot per input. The results show that SRA outperforms the other three and provides high throughput and low delay.

6.1. Uniform Traffic

We first tested the performance of SRA when the incoming traffic is i.i.d. Bernoulli with destinations uniformly distributed. Since SRA works in one iteration, we first ran PIM, iSLIP, and DSRR for only one iteration. We then ran them for four iterations. In all these cases, N is 16. That is, the switch size is 16 × 16. We used the same traffic pattern for all four schemes.

Figure 2 shows cell delay vs. offered load. When the load is 20% or less, all four schemes perform the same. But when the load increases, their performances are significantly different. When the load is less than 60%, PIM, iSLIP, and DSRR show about the same performance, while SRA is 6 times faster than the other three. When the load exceeds 60%, PIM becomes instable and iSLIP performs much better than DSRR. At this time, SRA outperforms the others by many times over. Compared to PIM and DSRR, iSLIP works much better. In terms of throughput, the situation is similar as shown in Figure 3.

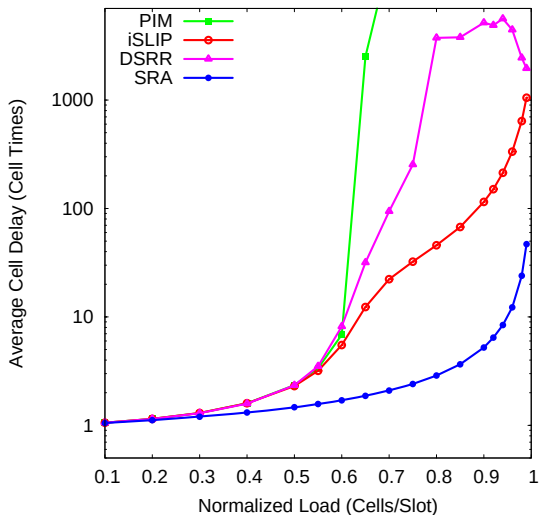


Figure 2. Cell delay under uniform traffic (one iteration).

In the case of four iterations, PIM, iSLIP, and DSRR all perform better than with one iteration. But compared to SRA, they are still off by a few times as shown in Figure 4 and Figure 5. The differentiation becomes the clearest when traffic load reaches 96% and higher. The advantage of DSRR over iSLIP is now obvious. DSRR approaches PIM very closely overall. Both PIM and DSRR perform better than iSLIP. The delay values for PIM, iSLIP, DSRR, and SRA at load 99.5% are 217, 451, 265, and 91 cell times respectively. SRA outperforms the others by 3 to 5 times at all load values.

Although PIM works better than iSLIP, PIM has its problems as discussed in [4]. That is why iSLIP has been adopted in many commercial switches. First, randomness is hard to implement at high speed since random selection has to be made over a time-varying set of elements. Second, PIM can be unfair especially when the inputs are oversubscribed. Third, PIM converges only after several iterations.

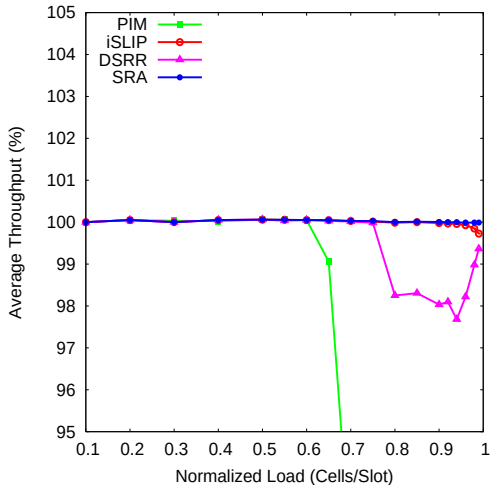


Figure 3. Throughput under uniform traffic (one iteration).

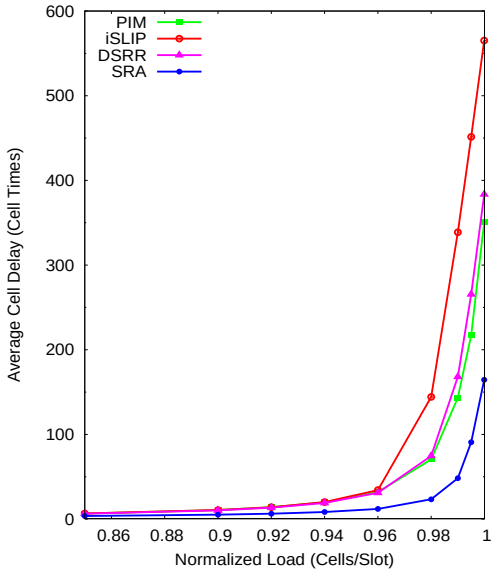


Figure 4. Cell delay under uniform traffic (four iterations).

Work on PIM is recent as in 1999 when Nong *et al.* proposed an analytical model and derived closed-form solutions on throughput, mean cell delay, and cell loss probability [27]. They found that the maximum throughput of the switch exceeds 99% with just four iterations under i.i.d. Bernoulli traffic with cell destinations uniformly distributed over all the output ports. Our simulations show the same throughput performance for PIM. Our simulations also show that on throughput DSRR is about the same as PIM, but iSLIP is off rather markedly at high load. SRA's throughput is closest to 100% among all four algorithms at all times.

6.2. Bursty Traffic

We modeled bursty traffic using interrupted Bernoulli process (IBP). IBP is the discrete version of interrupted Poisson process. IBP is similar to two-state Markov-modulated Bernoulli process, exponential on/off process, and Pareto on/off process. Switch size is again 16×16.

IBP has two states (on and off) and is characterized by three parameters α, p, and q. In each time slot, if the current state is on, the state remains on in the next time slot with probability p; if the current state is off, the state remains off in the next time slot with probability q. In the on state, a cell arrives in a time slot with probability α. The length of on state, X, and the length of off state, Y, have geometric distributions:

$$P\{X=x\}=(1-p)\,p^{(x-1)}$$

$$P\{Y=y\}=(1-q)\,q^{(y-1)}$$

The mean arrival rate or offered load, ρ, is

$$\rho=\frac{\alpha(1-q)}{2-p-q}$$

Average burst length is

$$b=E[X]=\frac{1}{1-p}$$

In each burst period, arrived cells all go to the same destination. Thus b measures how bursty the traffic is. In our simulations, we set b=128 cells and α=1. When α is set, p is set. To get various loads, we just vary the value of q.

As shown in Figure 6, SRA is faster than the other three over all loads. PIM, iSLIP, and DSRR were run for four iterations. At load 95.92%, the delay values are 4453, 5357, 4597, and 2391 cell times for PIM, iSLIP, DSRR, and SRA respectively. In all times, PIM, iSLIP, and DSRR are very close to each other, and SRA works 2 to 3 times faster than them.

Figure 7 shows how each performs in terms of throughput. Again, SRA maintains the highest throughput under all loads. Its throughput dips when load passes 90% but still less than the rest. Thus SRA is the most stable at providing high throughput.

6.3. Effect of Switch Size

The performance of iSLIP degrades considerably as switch size increases as shown in [4]. The switch slows down a time when N doubles. We saw little slowdown with SRA when switch size increases.

We ran a few simulations on an $N \times N$ switch with N being 4, 8, 16, 32, and 64. Traffic is i.i.d. Bernoulli with uniformly distributed destinations. As shown in Figure 8, when $N = 4$, cell delay is the smallest. But when N takes larger values, cell delay remains just about the same. That implies that the SRA scheduling scheme provides the same efficiency regardless of N. Thus SRA is scalable.

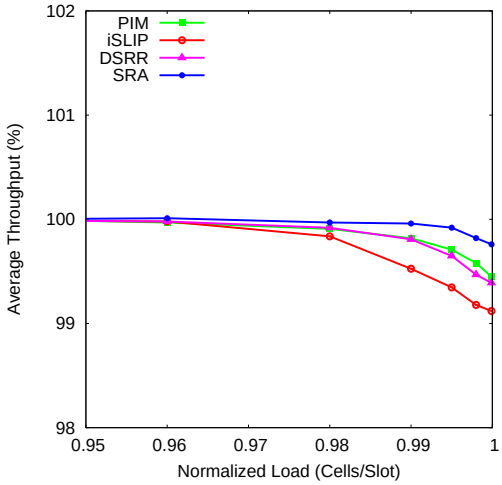


Figure 5. Throughput under uniform traffic (four iterations).

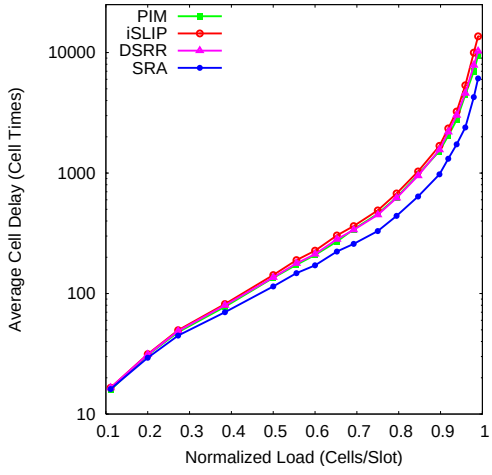


Figure 6. Cell delay under bursty traffic.

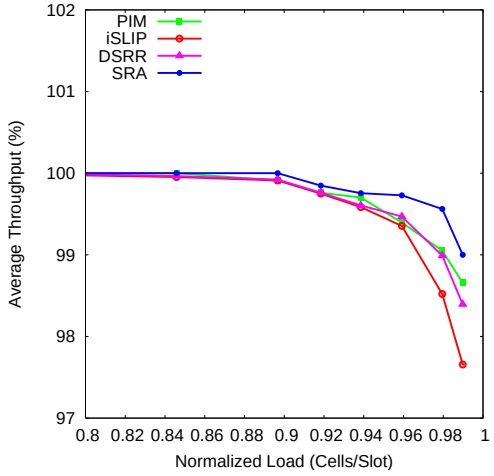


Figure 7. Throughput under bursty traffic.

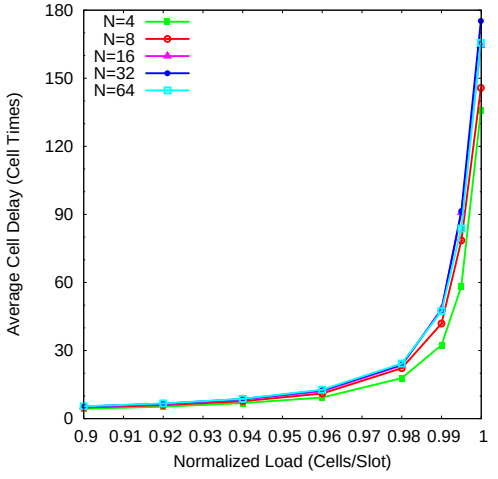


Figure 8. SRA cell delay as a function of switch size (uniform traffic).

6.4. Input Blocking

Consider that the switching fabric used is a conventional unbuffered $N \times N$ crossbar, as in our first hardware architecture described in Section 7. SRA requires an input port to be able to occasionally transmit multiple cells in a time slot, each cell to a different output port. Assume that an input port transmit k ($1 \le k \le N$) cells in a time slot in such a situation. We call k *cell multiplicity*. Contention of reading data at the same time from the same input memory for multiple outputs is called *input blocking*. Our simulations show that the occurrence of input blocking is very rare and indeed negligible.

The maximum input blocking delay occurs when N reads have to be performed at one time. For a given input traffic pattern, the probability of this occurrence is zero under SRA. We did simulations for an $N \times N$ switch under full load of uniform i.i.d. Bernoulli traffic and IBP

traffic when N is 16, 64, and 128. Table 1 shows the frequency data we obtained. An input port can send zero, one, or multiple cells in a time slot. When a send involves multiple cells, multiple reads to the same input memory, hence input blocking, occurs. Frequency of a cell multiplicity is calculated by dividing the number of sends of that particular cell multiplicity with the total number of sends during the simulation duration.

The simulations indicate that k tends to be far smaller than N. The chance for input blocking to happen at all is also low. Specifically, the data in Table 1 show that for $N = 64$ the occurrence of $k > 2$ is about 7% in the worst case. That $k > 5$ virtually does not occur. The probability is 7.81×10^{-7} for $k = 8$ and 0 for $k > 10$. Most input ports send only one cell or send none for a time slot. Some send two. Thus input blocking is slight. Figure 9 shows graphically the frequency of multiple reads when the load is full (1.0) for uniform Bernoulli traffic. The above property of low k holds true indifferently of N. Our simulations show that k is nearly unchanged when N is 16, 64, and 128 under the same uniform and bursty traffic conditions.

Table 1. Input blocking occurrence.

N	Sends	Frequency	
		Bernoulli	IBP
64	0	0.365469	0.371600
	1	0.371215	0.381245
	2	0.185188	0.180119
	3	0.060420	0.053726
	4	0.014478	0.011228
	5	0.002743	0.001824
	6	0.000426	0.000232
	7	0.000056	0.000023
	8	0.000003	0.000003
	9	0.000001	0.000000
	10–64	0.000000	0.000000
128	0	0.366870	0.373318
	1	0.369721	0.379410
	2	0.184585	0.180073
	3	0.060675	0.053620
	4	0.014747	0.011440
	5	0.002877	0.001870
	6	0.000454	0.000241
	7	0.000064	0.000025
	8	0.000007	0.000002
	9	0.000001	0.000000
	10–128	0.000000	0.000000

7. Switch Fabric Hardware

In this section, we discuss how to implement the SRA algorithm in hardware. We propose two alternative architectures that differ in the queuing structures and the crossbars used. Both architectures use the same SRA arbiters. The first architecture uses the same VOQs organization and the same crossbar as in a conventional iterative switch fabric. The second uses a different queuing structure and a crossbar that is designed specifically for implementing the SRA algorithm. Compared to conventional switch architectures that work for iterative matching algorithms, the SRA architectures are simpler and hence more effective in terms of area and power.

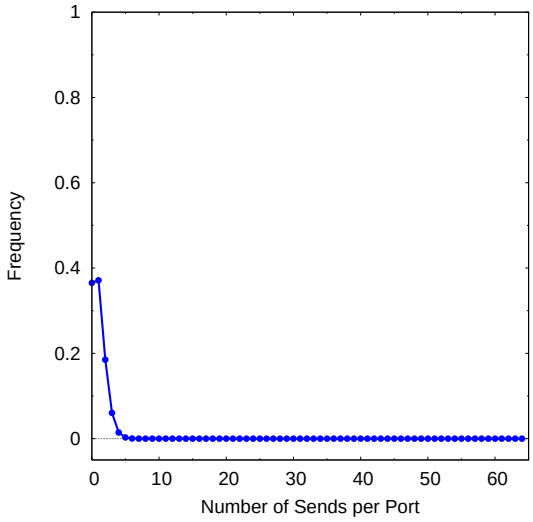


Figure 9. Frequency of input blocking occurrences. *N*=64. Full load of uniform Bernoulli traffic.

SRA has to perform certain operations due to input blocking although their occurrence is rare. The distinctive operations of SRA in each time slot are: 1) each input must do k reads instead of a single read; 2) up to k cells must be carried on a link to the crossbar; and 3) at the crossbar these cells must be split onto separate inputs. As we have discussed in Subsection 6.4, cell multiplicity k incurred by SRA is low. Thus these peculiar operations can be implemented in hardware with adequate simplicity, as in our two architectures.

7.1. Queuing Structure and Crossbar

As we have mentioned already, the two architectures we are proposing differ mainly in the queuing structure and crossbar each uses. Here we discuss the two architectures and how they are distinguished by their queuing structure and crossbar. We call the two uses of queuing structure and crossbar roughly as *designs*.

7.1.1. Design I

This design uses the vintage queuing structure and square crossbar that have been used for iterative matching algorithms since certainly iSLIP [4]. But, of course, it uses the much simpler SRA arbiters. This design can overcome input blocking. The architecture is shown in Figure 10.

With this design, input blocking may simply be solved in this way: Whenever there is more than one cell to be

sent by an input port in a time slot, make that time slot a little bit longer, enough for the multiple reads to complete. In fact, this may be a very feasible solution.

This solution is to allow each input port to send all k cells in a time slot. Thus, SRA inflicts no cell loss which is expensive. This is workable since k is never more than 10, and to make k reads causes a negligible extra delay than one read. It might be worth noting that the delay can be negligible mainly owing to fast memories and the high rate within the switch.

Input blocking would be a nonissue if the input port memory is capable of concurrent read. Should the memory technology be unavailable, other means must exist to mitigate the delay.

Memories supporting concurrent reads are being made. The SigmaRAM^{TM} memories of synchronous SRAMs had been planned for quite some time that will be capable of fast, random, multiple reads [28]. Current speed of existing SRAMs is as fast as 2 ns per operation with a clock rate of 333 MHz and a 24 Gbps throughput. Operations for mostly reads include access to look-up tables and parameter memory (e.g., QoS parameters, congestion avoidance profiles, min/max bandwidth limits). For these operations the common I/O SigmaRAM products provide very high bandwidth per pin and total memory bandwidth. The faster the memory speed, the less the impact of input blocking on fabric delay.

High-speed fabric rate diminishes the delay of input blocking. Memory speed needed to support the high data rate also abates the effect of input blocking. Operations of multiple reads cause only negligible extra delay. For a 40-Gbps fabric, getting one 53-byte cell out of the input memory would take about 11 μs. Simulations done on a real 40-Gbps switch fabric show that extra input blocking delay is merely 0.1 to 0.2 μs [29]. The fabric has 128 ports and carries IMIX traffic. The delay is incurred by doing up to 128 reads from the input memory at the same time. That is about 1% of the delay to perform just one read.

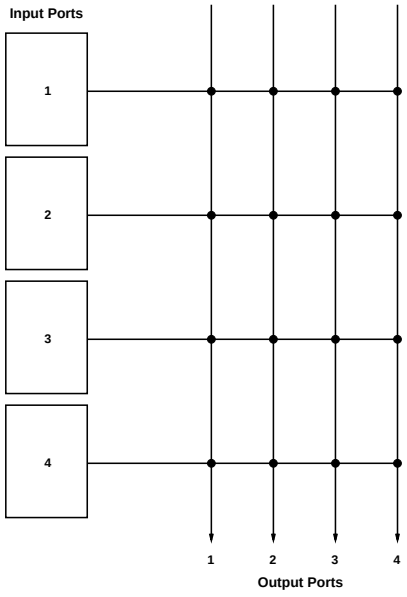


Figure 10. A 4 × 4 switch fabric that uses a square crossbar and contiguous memory for all VOQs at each input port. There is one data link (horizontal line) from an input port to the crossbar.

With SRA, speed gain appears to outweigh speed loss due to possible input blocking. SRA gets N cells sent to the outputs by simply letting the queue head element at the output to send and sending one grant signal back to the inputs. Using an iterative matching scheme, this process would take many iterations of request, grant, and accept with numerous signals sent. SRA removes this complexity at the expense of a much smaller delay of multiple reads. In view of this and the other aforementioned facts, input blocking in this context is a benign tradeoff for simplicity and speed.

With a conventional crossbar, each input port needs only one wire to connect to the crossbar by following the shortest path. The crossbar, being self-routing, will route the cells from different input ports to their destinations. Using one wire is workable and the crossbar is necessary because an input port can send at most one cell to one arbitrary yet distinct output port. The cell on the connecting wire can be going to any of the N output ports.

With SRA, an input port can send multiple cells going to different output ports in a time slot. The crossbar needs to direct the multiple cells arriving to it so that the cells go to separate inputs of the crossbar with correct synchronization.

7.1.2. Design II

This new switch architecture (Figure 11) is different from Design I. It uses separate memories for the VOQs and also a special crossbar. It is so suited to combat input blocking.

At an input port, traffic arrival is normalized to be one cell per time slot. Each input line card has N separate memories, one for each VOQ. An input line is connected to a $1\times N$ demultiplexer, which distributes incoming packets (segmented to cells) to different VOQs. Since each VOQ works over a separate memory, it needs a separate transmitter. Multiple transmitters need to transmit cells from corresponding VOQs simultaneously in a time slot to tackle input blocking. Each input port needs N transmitters.

In this architecture, an input port and hence $VOQ_{k,j}$, where $1 \le k \le N$, are connected to output O_j via an $N \times 1$ multiplexer (MUX). The $N \times N$ crossbar consists of N MUXs of size $N \times 1$. As such, the architecture may be better called a *multiple-multiplexer* (MMUX) switch.

As shown in Figure 11, this fabric has the following feature: At any time, each output port j is connected to at

most one VOQ at input i ($VOQ_{i,j}$) and each input port can have up to N VOQs connected to output ports. Note that such a pattern is exactly a maximum matching of G' found by SRA. Clearly, the MMUX crossbar has $O(N^2)$ crosspoints, resulting in a complexity of $O(N^2)$, which is the same as the complexity of a conventional $N \times N$ crossbar.

Besides being suitable for SRA, this new fabric has two additional advantages compared with conventional crossbar. First, the MMUX crossbar has a reduced diameter, which is the maximum number of crosspoints on an input-output path. With N $N \times 1$ MUXs, the diameter of the crossbar, which depends on the diameter of an $N \times 1$ MUX, is N, whereas the diameter of a conventional crossbar is $2N$. If each of the N $N \times 1$ MUXs were implemented as a (self-routable) binary tree, then the diameter of this crossbar would be $\log N$. Implementing a crossbar by electrical switching elements, the reduced diameter corresponds to smaller signal delay. If each crossing point is implemented by an electro-optic switching element, then the reduced diameter corresponds to less crosstalk and power loss. In fact, crosstalk virtually does not exist in optical implementation of the MMUX crossbar because there do not exist two connection paths in the fabric sharing a crosspoint at any time according to SRA.

The second advantage of this fabric is that it has a distributed control, as a result of the distributed feature of SRA. In the fabric, cell scheduling and transmission for each output port is totally independent of other output ports. Thus, the fabric can be considered as N subsystems, each consisting of all the VOQs designated to a particular output port and a MUX that connects these VOQs to their corresponding output. Then, the problem of synchronizing the entire fabric is reduced to synchronization of independent subsystems. This feature is particularly important when N is large.

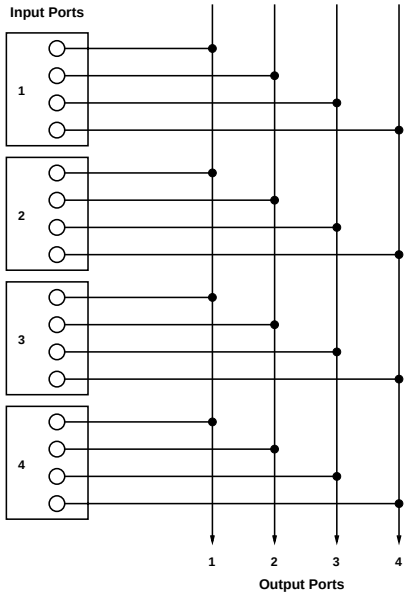


Figure 11. A 4 × 4 MMUX switch fabric in which each VOQ can get a link to an output port. The links (horizontal lines) are data lines.

The MMUX crossbar is different from the standard square crossbar. In this crossbar, each input can send 1 or more cells per time slot, and each output can receive none or 1 cell per time slot. In a standard crossbar each input/output can send/receive none or 1 cell per time slot. The MMUX crossbar is more powerful than a standard crossbar; it can do everything a conventional crossbar can do, but the converse is not true.

The question is how to reduce the usage of transmitters to make the hardware more scalable? As described in Subsection 6.4, simulations indicate that cell multiplicity k is low: an input port virtually never sends more than 5 cells in a time slot even for N=128, a reasonably large switch size. To exploit this property, the following approach can be taken. The resulted structure can be regarded as a variant to Design II proper.

For each input, instead of using N transmitters, use k transmitters. The total number of transmitters is now kN, much smaller than N^2. As before, each input has N single port memory modules, one for each VOQ. But each input needs an $N \times k$ switch and a $k \times N$ switch. The $N \times k$ switch is used to select and connect any k VOQs to the inputs of k transmitters. The $k \times N$ switch is used to connect the outputs of k transmitters to k outputs corresponding to the k selected VOQs.

The advantage of this approach is reduced number of transmitters and no memory access speedup. The disadvantage is additional cost due to the $N \times k$ and $k \times N$ switches. Total cost is $2kN^2 = O(N^2)$ crosspoints. But this may be worthwhile considering how many transmitters are saved.

7.2. Arbiter Structure and Layout

The hardware structure of an iterative matching scheme as that shown in Figure 21 of [4] and Figure 1 of [30] has two layers of arbiters: one consists of N grant arbiters and the other N accept arbiters. The arbiters of both layers have to work together to coordinate and phase the grant or accept actions. As such, all of the arbiters must be placed together, constrained by a state memory and update logic to receive requests from the VOQs next to the input ports, and by a file of decision registers next to the cross bar, thus forming a centralized unit. An arbiter itself typically takes on a round-robin structure (made of priority encoders) as in [4,31,32] or a tree or binary tree structure as in [30,33]. All have a $O(\log N)$ gate delay and consumes $O(N)$ gates.

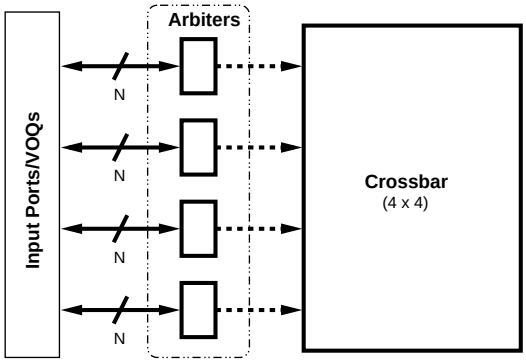


Figure 12. SRA arbiter layout. The dashed arrows indicate that data cells go from the input ports to the crossbar separately.

SRA needs only one layer of N arbiters as shown in Figure 12. The arbiters work in complete parallel, without any coordination or centralized arbitrating hardware. Thus they can be placed in a chip distributedly. Moreover, the arbiter itself implements only a FIFO queue and the associated operations such as enque and deque. The FIFO queue implies round-robin, whereas the discrete arbiters embody overall scheduling. As pointed out in Subsection 4.1, the size of the queue is N.

In Figure 12, the paths from input ports to the arbiters are independent of the data traffic paths as shown in Figure 11. There is no communication between the arbiters and the crossbar. Also, the layout applies to both cases where one wire (as in a conventional iterative fabric) or 4 wires (as for the MMUX fabric shown in Figure 12) are used to link an input port to the crossbar.

In hardware, a FIFO may consist of a set of read and write pointers, storage, control logic, and read and write lines. The read/write pointers are used to track the head and tail of the queue for deque and enque operations and generate queue status flags such as Empty and Full. Read and write lines are used to read out and write in data. In relation to the SRA algorithm, deque corresponds to grant and enque to update of input state. Each will need only one wire (data line) for proper connection to each input port. Storage may be SRAM, flip-flops, and latches. Control logic or FIFO controller contains read and write clocks, enforces synchronization, and drives the read and write pointers. A FIFO "node" needs to store *only* the number of the input port which, for instance, can be 8 bits to support 64 ports. The complexity of the arbiter: gate delay $O(1)$ and number of gates $O(N)$.

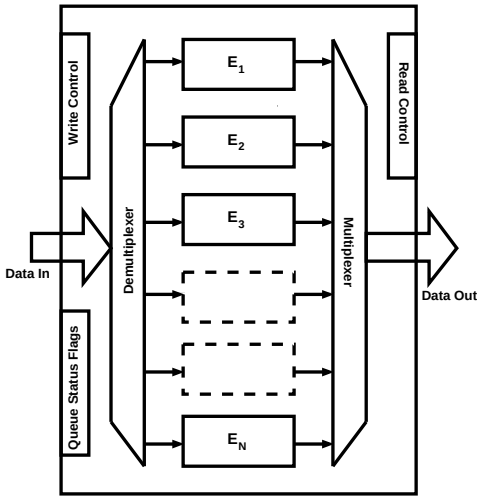


Figure 13. Hardware schematic of a FIFO device that can be used as an SRA arbiter. The elements may actually reside in contiguous memory.

FIFO hardware implementations are numerous. Figure 13 shows a hardware schematic of a FIFO device that fits the design features stated above. It is drawn in reference to Figure 1 of [13]. Work [34] shows a FIFO design that has more features than needed for an SRA arbiter, but is a good source for device details. In Figure 13, the data input port can be considered as consisting of the N wires coming from the N input ports, and the data output port N wires going to the input ports. The data input demultiplexer writes data into an element pointed to by the write pointer, whereas a data output multiplexer can be configured to read data from an element pointed to by the read pointer. Each of the elements marked $E_1, \cdots, E_N$ needs to store log$N$ bits. The elements can be in one contiguous piece of memory. The blocks for read, write, and queue status flags can be integrated into one controller. The design in [34] shows how all N FIFOs (arbiters) can be put in a single device that has nearly the same structure as what is shown in Figure 13.

In essence, the SRA arbiter is simpler. The arbiters needed for SRA scheduling are disparate. They can be placed in one chip with simple supporting components to perform overall scheduling.

7.3. Scalability

In Subsection 4.2, we showed that SRA can support high line rate and uses less auxiliary messages to make arbitration decisions. These properties plus the ones below indicate that SRA is more scalable than conventional iterative fabrics.

First, the SRA scheduler can expand to handle very large N without adversely affecting speed. This is demonstrated by our simulations as shown in Subsection 6.3. In contrast, an iterative algorithm degrades considerably in speed when N increases. The degradation is caused by the very arbitration logic of the algorithm more than by its iterations.

Second, SRA does not need constant feedback from the outputs about their readiness. CIOQ switches need this feedback for the outputs to make granting decisions. Therefore, the SRA scheduler need not be in a central

location. The output arbiters can be placed near the input ports such that the input status updates can be done more easily. These output arbiters can be spatially distributed and execute in parallel (Figure 1(a)).

Third, SRA can more rigorously ensure scalability. Between SRA and a conventional iterative fabric, given the same circuit complexity, SRA has increased scalability, as we elaborate below.

Let T_s be the time of one round (iteration) of arbitration. An iterative algorithm takes $T_s \log N$ time for one time slot, while SRA takes T_s time for one time slot. Consider that a time slot takes time T_c. In order for an iterative algorithm to work, it must meet the following condition:

$$T_s \log N \leq T_c. \quad \textbf{(11)}$$

Since T_c is constant, when N is large, (11) cannot be satisfied. At high line rates, T_s must decrease, (11) becomes harder to satisfy. But SRA needs only to satisfy $T_s \leq T_c$.

SRA completes an arbitration in a single round, so the time required for scheduling for one time slot is reduced significantly. Thus, SRA is more scalable in the sense of satisfying stringent real-time requirement.

In particular, the SRA arbiter circuit is less complex than the other components in the two SRA fabrics. The fabrics are thus scalable by the criterion of Li, Yang, and Zheng that the arbitration circuit should not be more complex than the interconnect [35].

Many scheduling algorithms we referenced in Section 2 require complex hardware if they are implementable at all. Li *et al*. [35] proposed to measure the complexity of a scheduling algorithm in an IQ switch by its structural complexity in hardware against that of the interconnect. They showed that a scheduler not more complex than any nonblocking interconnect can perform as well as non-scalable schedulers. Structural complexity is measured by the number of links (wires) used in the hardware in terms of switch size N.

7.4. Speedup and Egress Memory

SRA requires little speedup if any at all. Ideally, a cell gets out at an output each cell time of the line rate. There are no holdups anywhere in the fabric. The operation that takes the most time is sending the grants to the inputs and for the inputs to pass out the cells. It takes constant time for an output to dequeue the head element in the FIFO queue.

Factoring in input blocking, the speedup of this architecture $S < 1 + 10^{-6} \Rightarrow S = 1$. Thus it is fair to say that this architecture operates in line rate and needs no speedup. Note that this speedup is lower than the best and commonly believed speedup value of 2 for IQ switching.

In contrast, PIM, iSLIP, and DSRR complete a matching in $\log N$ iterations. This needs complex hardware to sustain small speedup. SRA completes a matching in one iteration with much fewer messages. The hardware is less complex yet to support no speedup.

SRA makes the fabric to be strictly scheduled. The scheduled fabric is a pull-type one as opposed to a push-type one. In this fabric, cells at the inputs wait to be explicitly summoned by the outputs into the fabric. No backpressure from the outputs is needed. Complete states of the VOQs at each input are made available to the output arbiters. Therefore, egress memory is not needed *by arbitration* in an SRA fabric. SRA is a one-hop scheduling scheme. As line rates increase, egress memory adds significant cost and latency in the system's datapath. Egress memory could be removed when fabric and traffic scheduling is all done at the input ports.

The following is particular of the MMUX architecture. As shown in Subsection 7.1, this architecture has separate memories and transmitters at the input ports. There is no conflict in sending cells from VOQs to output ports. In other words, at input line rate, a cell can be read and sent out of its VOQ without speedup. Thus no memory access speedup and no transmission speedup are required. However, there is an added hardware cost of N^2 transmitters for Design II proper and of kN transmitters and $O(N^2)$ switch crosspoints for the variant of Design II.

7.5. Compared to the Knockout Switch

As we have alluded to in Section 1, OQ scheduling requires N writes (plus N reads if the output ports are under one shared-memory) in one time slot. SRA transforms the N writes of OQ into N reads. A read operation is much easier to perform and needs minimal hardware support. Thus, SRA reduces the complexity while approaching OQ in speed. Note that actual OQ switches do exist despite the multiple-writes problem.

The Knockout Switch [36] is an example OQ switch which combats the multiple-writes problem by using concentration circuit. The multiple-writes problem here is seemingly similar to input blocking. In fact, the frequencies of cell contention accrued by traffic flow in the Knockout and SRA switches are strikingly close. Coincidentally, this helps validate the correctness of our algorithm and simulations. Below we discuss the design of the Knockout and compare its performance to SRA's.

The Knockout uses an $N \times L$ concentrator at each output port. The concentrator connects N input ports and fan them in to L outlets at the output port such that in a time slot at most L cells can be admitted although there can be N cells arriving each from one input port. The possible remaining $N - L$ cells are dropped, causing a loss probability. The purpose of selecting only L cells in each time slot is to reduce the number of output FIFO buffers and the control circuit complexity. However, the output port has to be able to do L *writes* to buffer the L cells since only one cell can exit the output port in a time slot. It has

been shown that a cell loss probability of only 10^{-6} is achieved with L as small as 8, regardless of switch load and size [36].

In comparison to the Knockout, under SRA, an input port only needs to do *k reads* in a time slot. Moreover, our simulations indicate that the probability for k=8 is 7.81×10^{-7} and the probability for $k > 10$ is zero, regardless of switch load and N. In the mean time, SRA and the scheduling scheme used by the Knockout both have constant time complexity. Therefore, SRA has several advantages over the Knockout.

Finally, it is worth concisely noting the two differences between the Knockout and SRA. A fundamental difference is that the Knockout is an OQ switch and SRA is for IQ switches. Secondly, unlike the Knockout, an SRA fabric does not contain concentration circuits and does no concentration because in any given time slot, only one of the N input ports connected to an output port will have a cell going through.

8. Conclusions

This paper proposes a new scheduling scheme which finds a maximum matching of a modified I/O mapping graph in a single iteration and shows that the proposed scheme achieves 100% throughput and has much lower delay than the conventional iterative scheduling schemes. Implementation issues are discussed and two new switch fabrics are presented.

The major innovation of the SRA algorithm is that it considers the matching in G'. Hence, the switch architectures for SRA are different from that for the iterative algorithms which consider the matching in G. Both architectures hinge on the simple SRA arbiter. The SRA arbiter is much simpler than the arbiter used for iterative schemes. Each SRA architecture uses a set of SRA arbiters that operate in parallel with no interaction. It differs from a general IQ switch architecture which typically has the complex scheduler located in a centralized unit. The SRA switch fabrics are therefore simpler in hardware than an iterative fabric.

Other aspects that are new of the SRA architectures include:

- Removal of egress memory for arbitration.
- Use of the free rule in arbiting: an input port can send k cells at a time. Analysis and simulations all show that this makes SRA to find a *maximum* matching in a time slot without iteration.
- No backpressure usage. Iterative schemes use grant and accept functions to exert backpressure.
- Speedup for SRA $S \ll 2$ cumulatively (virtually $S = 1$). This is easily calculated as we have shown in Subsection 7.4. This is a significant improvement over iterative schemes all of which require a speedup $S \geq 2$, although the MMUX option has an added cost of transmitters.
- Hardware implementations are analyzed to be doable, simple, and efficient.

A switch implementing SRA can be regarded as an IQ switch because traffic is queued at the input ports and no egress memory is needed for arbitration. If buffering for packet reassembly is done at the output port, then egress memory for that is needed and the architecture would be better called a CIOQ switch. However, if the MMUX architecture is used, the switch would be better called a "MMUX-based IQ switch" to distinguish it from the common IQ switch that typically uses an iterative scheduler. We alluded to this in Subsection 7.1.2.

The benefits of using SRA include high throughput and low delay. Note that cell delays incurred by the iterative PIM, iSLIP, and DSRR in simulations would be much higher if the time spent on iterating the algorithms were taken into account. In addition, SRA is scalable and reduces the complexity of switching.

We hope SRA could serve as a design reference for IQ (or CIOQ) switches. Whether SRA can be useful to IQ switches with buffered crossbars, the other promising alternative to designing IQ switches, is yet to be investigated. Also, the SRA fabrics are best-effort architectures. Quality of service and multicast support merits further study.

9. Acknowledgements

This work is supported in part by NSF CCR-0309461, NSF IIS-0513669, HK CERG 526007 (HK PolyU B-Q06B), NSFC 60728206, and NSF 0714057.

The code used for the simulations is based on the code authored by Prof. Ken Christensen of the University of South Florida. PIM and iSLIP results obtained with the initial code were validated against results shown in [37].

10. References

[1] N. McKeown, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," in Proceedings of IEEE INFOCOM'96, pp. 296–302, March 1996.

[2] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," IEEE Transactions on Communications, Vol. 47, No. 8, pp. 1260–1267, August 1999.

[3] A. Mekkittikul and N. McKeown, "A practical scheduling algorithm to achieve 100% throughput in input-queued switches," in Proceedings of IEEE INFOCOM'98, pp. 792–799, March 1998.

[4] N. McKeown, "The iSLIP scheduling algorithm for input-queued switches," IEEE/ACM Transactions on Networking, Vol. 7, No. 2, pp. 188–201, April 1999.

[5] N. McKeown, J. Walrand, and P. Varaiya, "Scheduling cells in an input-queued switch," IEE Electronics Letters, Vol. 29, No. 25, pp. 2174–2175, December 1993.

[6] R. O. LaMaire and D. N. Serpanos, "Two-dimensional round-robin schedulers for packet switches with multiple input queues," IEEE/ACM Transactions on Networking, Vol. 2, No. 5, pp. 471– 482, October 1994.

[7] A. Hung, G. Kesidis, and N. McKeown, "ATM input-buffered switches with the guaranteed-rate property," in Proceedings of IEEE ISCC'98, pp. 331–335, June 1998.

[8] M. Yang and S. Q. Zheng, "An efficient scheduling algorithm for CIOQ switches with space-division multiplexing expansion," in Proceedings of IEEE INFOCOM 2003, pp. 1643–1650, March 2003.

[9] H. J. Chao and J.-S. Park, "Centralized contention resolution schemes for a large-capacity optical ATM switch," in Proceedings of IEEE ATM Workshop'98, pp. 11–16, May 1998.

[10] J. Chao, "Saturn: A terabit packet switch using dual round-robin," IEEE Communications Magazine, Vol. 38, No. 12, pp. 78–84, December 2000.

[11] Y. Li, S. Panwar, and H. J. Chao, "On the performance of a dual round-robin switch," in Proceedings of IEEE INFOCOM 2001, pp. 1688–1697, April 2001.

[12] C.-S. Chang, W.-J. Chen, and H.-Y. Huang, "On service guarantees for input-buffered crossbar switches: A capacity decomposition approach by Birkhoff and von Neumann," in Proceedings of IEEE/IFIP IWQoS'99, pp. 79–86, May 1999.

[13] C.-S. Chang, W.-J. Chen, and H.-Y. Huang, "Birkhoff-von Neumann input buffered crossbar switches," in Proceedings of IEEE INFOCOM 2000, pp. 1614–1623, March 2000.

[14] C. -S. Chang, D. -S. Lee, and C. -L. Yu, "Generalization of the Pollaczek-Khinchin formula for throughput analysis of input-buffered switches," in Proceedings of IEEE INFOCOM 2005, Vol. 2, pp. 960–970, March 2005.

[15] J. G. Dai and B. Prabhakar, "The throughput of data switches with and without speedup," in Proceedings of IEEE INFOCOM 2000, pp. 556–564, March 2000.

[16] A. Gourgy and T. H. Szymanski, "Tracking the behavior of an ideal output queued switch using an input queued switch with unity speedup," in Proceedings of IEEE HPSR 2004, pp. 61–66, April 2004.

[17] S. Mneimneh, "Matching from the first iteration: An iterative switching algorithm for an input queued switch," IEEE/ACM Transactions on Networking, Vol. 16, No. 1, pp. 206–217, February 2008.

[18] R. Panigrahy, A. Prakash, A. Nemat, and A. Aziz, "Weighted random matching: A simple scheduling algorithm for achieving 100% throughput," in Proceedings of IEEE HPSR 2004, pp. 111–115, April 2004.

[19] V. Tabatabaee and L. Tassiulas, "Max-min fair self-randomized scheduler for input-buffered switches," in Proceedings of IEEE HPSR 2004, pp. 299–303, April 2004.

[20] T. E. Anderson, S. S. Owicki, J. B. Saxe, and C. P. Thacker, "High-speed switch scheduling for local-area networks," ACM Transactions on Computer Systems, Vol. 11, No. 4, pp. 319–352, November 1993.

[21] Y. Jiang and M. Hamdi, "A fully desynchronized round-robin matching scheduler for a VOQ packet switch architecture," in Proceedings of IEEE HPSR 2001, pp. 407–411, May 2001.

[22] H. Kim and K. Kim, "Performance analysis of the multiple input-queued packet switch with the restricted rule," IEEE/ACM Transactions on Networking, Vol. 11, No. 3, pp. 478–487, June 2003.

[23] H. Kim, C. Oh, Y. Lee, and K. Kim, "Throughput analysis of the bifurcated input-queued ATM switch," IEICE Transactions on Communications, E82-B(5), pp. 768–772, May 1999.

[24] C. Kolias and L. Kleinrock, "Throughput analysis of multiple input-queueing in ATM switches," in Proceedings of the International IFIP-IEEE Conference on Broadband Communications, pp. 382–393, April 1996.

[25] K. L. Yeung and S. Hai, "Throughput analysis for input-buffered ATM switches with multiple FIFO queues per input port," IEE Electronics Letters, Vol. 33, No. 19, pp. 1604–1606, September 1997.

[26] M. J. Karol, M. G. Hluchyj, and S. P. Morgan, "Input versus output queueing on a space-division packet switch," IEEE Transactions on Communications, COM-35(12), pp. 1347–1356, December 1987.

[27] G. Nong, J. K. Muppala, and M. Hamdi, "Analysis of nonblocking ATM switches with multiple input queues," IEEE/ACM Transactions on Networking, Vol. 7, No. 1, pp. 60–74, February 1999.

[28] SigmaRAM Consortium, SigmaRAM^{TM} targets high speed networking applications, White paper, 2008. http://www.sigmaram.com/white paper.htm.

[29] G. Bracha, "Removing egress memory from switching architectures," CommsDesign.com, February 2003.

[30] S. Q. Zheng, M. Yang, J. Blanton, P. Golla, and D. Verchere, "A simple and fast parallel round-robin arbiter for high-speed switch control and scheduling," in Proceedings of the 45th IEEE Midwest Symposium on Circuits and Systems (MWSCAS-2002), Vol. 2, pp. 671–674, August 2002.

[31] P. Gupta and N. McKeown, "Designing and implementing a fast crossbar scheduler," IEEE Micro, Vol. 19, No. 1, pp. 20–28, January/February 1999.

[32] Y. Tamir and H.-C. Chi, "Symmetric crossbar arbiters for VLSI communication switches," IEEE Micro, Vol. 4, No. 1, pp. 13–27, January 1993.

[33] H. J. Chao, C. H. Lam, and X. Guo, "A fast arbitration scheme for terabit packet switches," in IEEE GLOBECOM '99, Vol. 2, pp. 1236–1243, Rio de Janeiro, Brazil, December 1999.

[34] C. A. Karnstedt, B. L. Chin, P. Shamarao, and M. Montana, "Integrated circuit FIFO memory devices that are divisible into independent FIFO queues, and systems and

methods for controlling same," U.S. Patent 6,907,479, June 2005.

[35] C. Li, S. Q. Zheng, and M. Yang, "Scalable schedulers for high-performance switches," in Proceedings of IEEE HPSR 2004, pp. 198–202, April 2004.

[36] Y.-S. Yeh, M. G. Hluchyj, and A. S. Acampora, "The knockout switch: A simple, modular architecture for high-performance packet switching," IEEE Journal on Selected Areas in Communications, Vol. 5, No. 8, pp. 1274–1283, October 1987.

[37] N. McKeown and T. E. Anderson, "A quantitative comparison of iterative scheduling algorithms for input-queued switches," Computer Networks and ISDN Systems, Vol. 30, No. 4, pp. 2309–2326, December 1998.

Int. J. Communications, Network and System Sciences, 2009, 3, 169-247
doi:10.4236/ijcns.2009.23022 Published Online June 2009 (http://www.SciRP.org/journal/ijcns/).

On the Performance of Traffic Locality Oriented Route Discovery Algorithm with Delay

Mznah AL-RODHAAN[1], Lewis MACKENZIE[2], Mohamed OULD-KHAOUA[3]
[1]*Sponsored by King Saud University, Riyadh, Saudi Arabia*
[1,2]*Department of Computing Science, University of Glasgow, Glasgow, UK*
[3]*Department of Electrical & Computer Engineering, Sultan Qaboos University, Muscat, Oman*
Email: {*rodhaan, lewis*}*@dcs.gla.ac.uk, mok@squ.edu.om*
Received January 24, 2009; *revised April* 2, 2009; *accepted April* 5, 2009

ABSTRACT

In MANETs, traffic may follow certain pattern that is not necessarily spatial or temporal but rather to follow special needs as a part of group for collaboration purposes. The source node tends to communicate with a certain set of nodes more than others regardless of their location exhibiting traffic locality where this set changes over time. We introduce a traffic locality oriented route discovery algorithm with delay, TLRDA-D. It utilises traffic locality by establishing a neighbourhood that includes the most likely destinations for a particular source node. The source node broadcasts the route request according to the original routing used. However, each intermediate node broadcasts the route request with a delay beyond this boundary to give priority for route requests that are travelling within their own source node's neighbourhood region. This approach improves the end-to-end delay and packet loss, as it generates less contention throughout the network. TLRDA-D is analysed using simulation to study the effect of adding a delay to route request propagation and to decide on the amount of the added delay.

Keywords: MANETs, On-Demand Routing Protocols, Route Discovery, Delay, Congestion, Simulation Analysis

1. Introduction

When mobile devices such as notebooks and PDAs appeared, users wanted wireless connectivity and this duly become a reality. Wireless networks could be infrastructure-oriented as in access point dependent networks [1] or infrastructure-less multi-hop such as Mobile Ad hoc NETworks (MANETs) [1,2]. Some of the dominant initial motivations for MANET technology came from military applications in environments that lack infrastructure. However, MANET research subsequently diversified into areas such as disaster relief, sensors networks, and personal area networks [2].

The design of an efficient routing strategy is a very challenging issue due to the limited resources in MANETs [1]. MANETs routing protocols can be divided into three categories: proactive, reactive, and hybrid [3]. In proactive routing protocols (table-driven), the routes to all the destinations (or parts of the network) are determined statically at the start up then maintained using a periodic route update process. An example of this class of routing protocols is the Optimized Link State Routing Protocol (OLSR) [4]. However, in reactive routing protocols (on-demand), routes are determined dynamically when they are required by the source using a route discovery process. Its routing overhead is lower than the proactive routing protocols if the network size is relatively small [5]. Examples of this class are Dynamic Source Routing (DSR) [6] and Ad Hoc On Demand Distance Vector (AODV) [7]. Finally, hybrid routing protocols combine the basic properties of the first two classes of protocols; so they are both reactive and proactive in nature. Zone Routing Protocol (ZRP) [8] is an example belonging to this class.

In on-demand routing protocols, when a source node needs to send messages to a destination it initiates a broadcast-based route discovery process looking for one or more possible paths to the destination where the broadcasting of the route request dominates most of the routing overhead.

In this paper, a traffic locality oriented route discovery algorithm that uses delay, TLRDA-D, is introduced.

Moreover, TLRDA-D is analysed using simulation to understand the relationship between congestion and delay and ease the decision on the amount of the added delay.

The rest of the paper is organised as follows: Section 2 presents the related work while Section 3 presents the proposed algorithm; evaluates the performance and describes the simulation environment and observation. Finally, Section 4 concludes this study.

2. Related Work

The principle of locality was first applied in memory referencing behaviour [9] then it was subsequently observed in the use of other resources such as file referencing [10]. The locality of reference concept deals with the process of accessing a single resource more than once. It includes spatial and temporal locality [11,12]. In networking, locality is observed through the fact that devices within the same geographical area tend to communicate more often than those that are further apart, and exhibit both temporal and spatial locality [13]. The importance of traffic locality concept is recognized in networking. Traffic locality concept is a motivation factor behind network clusters and workgroups [14]. While in infrastructure wireless networks, traffic locality is utilized to improve load balancing in base stations [1,15]. In MANETs, locality is observed through the fact that neighbours, nodes in the same geographical area, tend to receive communication from the same sources, highlighting the spatial locality. Also, nodes communicated within the near past have high probability of re-communicating in the near future leading to temporal locality [16]. Sometimes a node communicates with a certain set of nodes more than others within a particular time regardless of their locations, highlighting the traffic locality [17].

3. Traffic Locality Oriented Route Discovery Algorithm with Delay (TLRDA-D)

MANETs are very useful in applications that need immediate collaboration and communication with the absence of network infrastructure where a temporary connection can be established for quick communication. These collaborative jobs demand traffic to be between known source-destination pairs to accomplish specific tasks. So if this pattern of traffic is found in an application then the design of the algorithm should utilize it.

Looking at the traffic behaviour of MANETs, the traffic may follow a certain pattern, not purely spatial or temporal, in which the source node tends to communicate with a set of nodes more than others regardless of their locations in a connected network. The traffic locality of a particular source node is captured in its working set. The working set is a set of nodes that the source node is mostly communicating with, not necessarily neighbours where members of the working set change over time. Moreover, the traffic locality is identified by the intensity of traffic within the working set over some time interval. If a source node exhibits traffic locality with a certain destination, the intermediate node comprising the route in question will also be a member of the source node's working set until one of them moves far away.

MANETs exhibit traffic locality due to the communication requirements of the users carrying and operating them. One common application that exhibits traffic locality in MANETs is a group communication ad hoc network [18] where a group of nodes communicate to accomplish a common goal.

In this paper, traffic locality concept [17] is utilized to improve the route discovery process in on-demand routing protocols for MANETs. It is used to develop a new adaptive route discovery algorithm, TLRDA-D. The algorithm works by gradually building up the node neighbourhood as a region centred at the source node and expected to contain most of the members of its working set where the whole connected network consists of two disjoint regions: *neighbourhood* and *beyond-neighbourhood*.

Establishing this neighbourhood is a challenging endeavour as it must adapt according to the traffic in an effort to build then maintains the neighbourhood region that reflects the current working set. Upon joining the network, the new node needs a start-up period during which it uses the original broadcast algorithm depending on the routing algorithm used.

Since the neighbourhood region contains the source node's working set, no extra delays are imposed in this region to avoid delaying the route discovery process. On the other hand, delaying a fulfilled route request in the beyond-neighbourhood region reduces channel contention without adding any latency to the discovery process.

Due to the scarce resources in MANETs, the algorithm is kept simple by avoiding the collection or manipulation of large amount of data. Furthermore, the global information is avoided because it is unavailable in a real environment that uses no external resources.

Each node has a locality parameter *LP* where $LP \in \mathbb{N}^*$ which corresponds to the current estimated depth of its neighbourhood as it might be defined by the *weighted average* of hop counts between that source node and destinations as in Equation 1 including route finder. The finder of a route is the first node that finds the route in its cache table whether it is the destination or an intermediate node.

Let $s \in N$ be a source node in a network of N nodes and define a function, $h_s : N \to \mathbb{Z}^+ \bigcup \{0\}$ where $h_s(u)$ is the hop count between s and some other node $u \in N$

and $h_s(s) = 0$. A node, *x*, is considered to be part of the working set of a source node, *s*, if $h_s(x) \le LP$. In TLRDA-D algorithm, source node broadcasts route requests after adding the value of its *LP* to the route request packet so intermediate nodes can decide if the route request is within its source node's neighbourhood or not. To avoid ambiguity we will use LP_r to refer to the *LP* stored in the route request. Also to calculate *LP,* the source node needs to store locally the number of its previous route requests.

Formally, we can view the issue as a two tier-partition where the two tiers $\{\tau_1, \tau_2\}$ are the neighbourhood and beyond-neighbourhood respectively in a network that exhibits traffic locality. It is obvious that the two tiers are disjoint sets so $\tau_1 \bigcap \tau_2 = \phi$. Let us consider a source node *s*, any node $v \in \tau_1$ satisfies the condition $h_s(v) \le LP_r$ and any node $u \in \tau_2$ should satisfy the condition $h_s(u) > LP_r$. *LP* is continuously tuned to adapt to the current situation using the values of $h_s(d)$.

The algorithm is adaptive and adjusts its neighbourhood depth, *LP*, to expand or shrink the neighbourhood boundary. If the destination is outside the neighbourhood then this requires the neighbourhood to be adjusted by the following strategy: *LP* is adjusted by taking the weighted average of the current value of *LP* and the new hop count extracted from the received route reply packet.

To illustrate the neighbourhood adjustment process, let us consider the source node *s* at any time after completing its start up phase; when *s* receives a reply answering its current query it updates its *LP* using Equation 1 after extracting $h_s(d)$ from the received route reply packet and *y* is the number of previous route requests that already been sent by *s*. If $h_s(d) \ge LP_{old}$ then the neighbourhood of *s* expands; otherwise it shrinks.

$$LP_{old} = \alpha \times LP_{old} + (1-\alpha) \times h_s(d)$$

$$LP_{new} = \begin{cases} \lceil LP_{old} \rceil & h_s(d) \ge LP_{old} \\ \lfloor LP_{old} \rfloor & h_s(d) < LP_{old} \end{cases} \qquad \alpha = \frac{y}{(y+1)} \qquad \textbf{(1)}$$

Figure 1 shows the steps of updating the locality parameter *LP* by the source node after receiving the route reply so the source node will be ready for next route request. For clarity, the function Ceiling will return the smallest integer greater than or equal to its parameter while the function Floor will return the greatest integer less than or equal to its parameter. To prevent α from approaching 1 as y gets bigger due to $\lim_{\alpha \to \infty}(\alpha) = 1$, where only the function Ceiling or Floor will affect the value of *LP*, we need to reset y to an initial value, *Initial-y*, when y reaches its maximum value, *max-y*. Each time y is initialised to 1, the partial historical information

Algorithm preformed by source node receiving a route reply and y = previous number of route requests.

```
1:  If y >=max-y then
2:      y = Initial-y
3:  End if
4:  α = y/(y+1)
5:  LP_new = αLP_old + (1−α)h_s(d)
6:  If h_s(d) < LP_old then
7:      LP_new = Floor(LP_new)
8:  Else
9:      LP_new = Ceiling(LP_new)
10: End if
11: LP_old = LP_new
12: y = y+1
```

Figure 1. Update procedure for the locality parameter LP at the source node in TLRDA-D.

represented by LP_{old} is given the same weight as the hop count. Alternatively, if *y* initialised by zero all the weight is given to the hop count.

In TLRDA-D, D stands for a delay where TLRDA-k denotes an instant of the algorithm where the delay equals to k units of time. Intermediate nodes in TLRDA-D broadcast route requests according to the on-demand routing algorithm used while route requests propagating within the neighbourhood boundary. However, beyond this boundary TLRDA-D broadcasts route requests with a delay at each node until the route request broadcast fades or the time to live (TTL) reaches zero.

The motive for adding this delay in the beyond-neighbourhood region is to give higher priority to route requests that are broadcasted within their own source node's neighbourhood regions. Moreover, other route requests that are travelling within their source node's beyond neighbourhood regions have higher chance of being already fulfilled thus they are given lower priority. This approach not only improves the average route discovery time but also improves the latency of the whole network, as it generates less contention throughout the network.

The delay should be calculated by monotonic non- decreasing function as the route request propagates further within beyond neighbourhood region, since the chance of route request fulfilment increases with each hop when the route request moves away from the source node's neighbourhood region. The delay increment can be logarithmic, linear, polynomial, or exponential. However, the exponential increase yields a huge amount of delay that may affect the discovery time if route finder is within the beyond-neighbourhood region which makes it unsuitable for resource-sensitive environment like MANETs and

hence ruled out.

The simulation is used to help us decide on the amount of delay that needs to be imposed to the route request dissemination in the beyond-neighbourhood region for TLRDA-D and whether it should be logarithmic, linear or polynomial. TLRDA-D has been implemented using five different amounts of delay (d_i) where d_i at any intermediate node takes the following values:

$$d_i = \begin{cases} \log_2(LP) & i = 0 \\ 2^{i-1} LP & i = 1,2,3 \\ LP^2 & i = 4 \end{cases} \quad \textbf{(2)}$$

In TLRDA-D, upon receiving a route request; each node performs the steps shown in Figure 2. If the route request has been received before then it is considered redundant and thus discarded. Otherwise, the receiving node compares *LP* value from the route request packet with the hop count after counting itself as an extra hop, if the node resides in the beyond-neighbourhood region of the route request initiator then the node holds the route request for *d* units of time then processes it. Otherwise, the node processes the route request according to the routing algorithm used.

If a route reply is not received within an estimated period of time called NETwork Traversal Time (*NETTT*), the source node will try again to discover the route by broadcasting another route request for a maximum number of tries. So the source node waits *NETTT* units of time to receive a reply before trying to search for the destination again. The worst case scenario is assumed and Node Traversal Time (*NTT*) follows the on-demand routing algorithm used in a network with diameter of *D* hops. TLRDA-D calculates this estimated time as:

$$NETTT = 2\{(LP * NTT) + (D - LP)(NTT + d_i)\} \quad \textbf{(3)}$$

Step preformed by each node upon receiving a route request in TLRDA-D
1: If i = 0 then $d = \log_2(LP_r)$
3: End if i = 1 then $d = LP_r$
4: End if i = 2 then $d = 2*LP_r$
5: End if i = 3 then $d = 4*LP_r$
6: End if i = 4 then $d = LP_r * LP_r$ end if
7: End if
8: End if
9: End if
10: End if
11: If route request is a duplicate
12: Discard the route request
13: Else
14: If hop_count > LP_r then
15: Wait *d* units of time
16: End if
17: Process the route request
18: End if

Figure 2. Route request messages processing at each node for TLRDA-D.

In on-demand routing algorithms, when an intermediate node *m* receives a route request for the first time; it stores: the broadcast ID and the route request originator IP address in its routing table, if it has such a table, for a an estimated time Broadcast Cache Time (*BCT*) as part of the route request processing steps. This information is used to distinguish between new and redundant route requests. When *BCT* expires, the route request record is deleted from the routing table. TLRDA-D calculates the time as:

$$BCT = \begin{cases} BCT & h_s(m) \le LP_r \\ BCT + d_i & h_s(m) > LP_r \end{cases} \quad \textbf{(4)}$$

3.1. Simulation Analysis

A simulation has been conducted to evaluate the new algorithm, TLRDA-D, and compare it with AODV. TLRDA-D algorithm was implemented as a modification to AODV implementation in NS2 network simulator, version 2.29 [19]. NS2 was used to conduct extensive experiments for performance evaluation and comparison.

Mobile nodes are assumed to operate in a squared simulation area of 1000m × 1000m. The transmission range is fixed to 100m in all nodes to approximately simulate networks with a minimum hop count of 10 hops between two border nodes one on opposite sides in a connected network. Each run was simulated for 900 seconds of simulation time, ignoring the first 30 seconds as a start-up period for the whole network. For each topology, 30 runs were performed then averaged to produce the graphs shown throughout this paper and a 95% confidence interval is shown as standard error bars in the relevant figures. Table 1 provides a summary of the chosen simulation parameter values.

The comparison metrics include:

• End-to-end delay: the total delay for the application data packet while transmitted from source to destination plus the route discovery time which is the round trip time from sending a route request until receiving the route reply.

• Packet loss: the number of dropped packets in a single run.

• Route request overhead: measured by the number of received route requests in the whole network.

A traffic generator was used to simulate constant bit rate (CBR) with payload of 512 bytes. Moreover, each five communication sessions were simulated between one source and five destinations randomly selected in a group of ten nodes to simulate traffic in an application that exhibit traffic locality. Data packets are transmitted at a rate of four packets per second, assuming nodes are identical, links are bidirectional, and mobile nodes oper-

ate in a flat arena.

In MANETs, the entity mobility models typically represent nodes whose movements are completely independent of each other, e.g. the Random Way Point (RWP) model [20]. However, a group mobility model may be used to simulate a coop erative characteristic such as working together to accomplish a common goal. Such a model reflects the behaviour of nodes in a group as the group moves together, e.g. Reference Point Group Mobility (RPGM) model [21,22].

The RPGM mobility generator was used [23] to generate mobility scenarios for all of our simulations since it models the random motion of groups of nodes and of individual nodes within the group. Group movements are based upon the movement of the group reference point following its direction and speed with Speed Deviation Ratio and Angle Deviation Ratio = 0.5. Moreover, nodes move randomly within their group with a speed randomly selected between 1m/s and 15m/s with 50s as pause time. Each group contains 10 nodes.

In our simulation, we concentrate on varying three major parameters to study their effect on TLRDA-D performance: network size, traffic load, and maximum speed in three different cases by varying one parameter while keeping the other two constant.

Effect of network size: when the network size increases, the average hop length of routes also increases which may increase the error rate and/or increase network latency. Simulation has been performed using nine topologies with different number of nodes, multiples of 10, from 20 (small size network) to 100 (moderate size network) with traffic load of 10 communication sessions and a maximum speed of 15m/s.

Figure 3 shows the superiority of TLRDA-D over AODV in reducing the end-to-end delay due to reducing congestion level especially when d_2, d_3 or d_4 is used as the amount of delay. For instance, in TLRDA-d_2, TLRDA-d_3, and TLRDA-d_4, the end-to-end delay was reduced by nearly 53% in small size network and by 68% in moderate size network compared to AODV. Moreover, this figure clearly shows that d_2, d_3 or d_4 yield in average almost the same end-to-end delay. The amount of delay added in TLRDA-d_2 was adequate to achieve the best discovery time in our scenarios as adding more delay will not yield further contention improvement. In average, route requests in TLRDA-D reside in the network for longer time than in the case of AODV (not shown here). This is due to the added delay which increases overhead yet reduces discovery time.

Figure 4 shows that TLRDA-D loses fewer packets compared to AODV by 1% to 30% in small size network and by 22% to 62% in moderate size network because TLRDA-D reduces congestion level. In TLRDA-D, the number of received route requests is more than that of AODV as shown in Figure 5. Some of the saved packets, gained in TLRDA-D as a result of reducing packet loss, are route requests which justify the increase in route request overhead. Those route requests might be duplicate copies but were dropped because of congestion or/and collision rather than redundancy. The rest of the saved packets can be any kind which might be useful but dropped in AODV due to high channel contention or collision. TLRDA-d_2, TLRDA-d_3, and TLRDA-d_4 lose fewer packets than TLRDA-d_0 and TLRDA-d_1 which improves network performance.

Effect of traffic load: Traffic load of sizes 5 (light traffic) to 35 (heavy traffic) communication sessions incremented by 5 were injected in networks of size seventy nodes and maximum speed of 15m/s. A reasonably incremented amount of traffic was used to test our algorithm meanwhile avoiding saturation.

Also in this analysis, when TLRDA-D uses d_2, d_3 or d_4 as amount of delay, the algorithm yield in average almost the same end-to-end delay as depicted from Figure 4 for these three instances among all experimented instances

Table 1. System parameters.

Parameter	Value
Transmission range	100m
Topology size	1000×1000m
Simulation time	900s
Packet size	512 bytes
Packet rate	4pkt/s
Traffic load	5,10,…,35 sessions
Traffic type	CBR(UDP)
Antenna type	Omni Antenna
MAC protocol	IEEE 802.11with RTS/CTS
Maximum speed	2,5,7,10,13,15m/s
Minimum speed	1m/s
Pause time	50s
Mobility model	RPGM model
SDR, ADR	0.5
Propagation model	Two-Ray Ground model

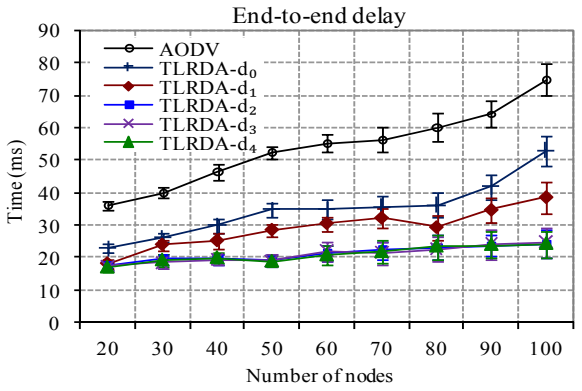


Figure 3. End-to-end delay verses network size for networks of 10 communication sessions and 15m/s as maximum speed.

of TLRDA-D. The end-to-end delay was reduced by nearly 57% in light traffic and 65% in heavy traffic for TLRDA-d_2, TLRDA-d_3, TLRDA-d_4 compared to AODV. So, TLRDA-D has end-to-end delay lower than AODV from traffic load prospective.

Furthermore, TLRDA-d_2, TLRDA-d_3, and TLRDA-d_4 have almost the same end-to-end delay that is lower compare to both TLRDA-d_0 and TLRDA-d_1. This improvement in the end-to-end delay is due to the reduction in channel contention where the application data can travel earlier and quicker which improves the network performance. Moreover, TLRDA-D reduces packet loss in the whole network compared to AODV as shown in Figure 7. This improvement in TLRDA-D over AODV ranges from 3% to 65% in light traffic while it ranges between 10% and 53% in heavy traffic.

The packet loss is nearly the same for the three instances TLRDA-d_2, TLRDA-d_3, and TLRDA-d_4 and better than both TLRDA-d_0 and TLRDA-d_1. Also in this analysis, some of these saved packets in TLRDA-D might be route requests which justify the increment in route request overhead in TLRDA-D over AODV as in Figure 6.

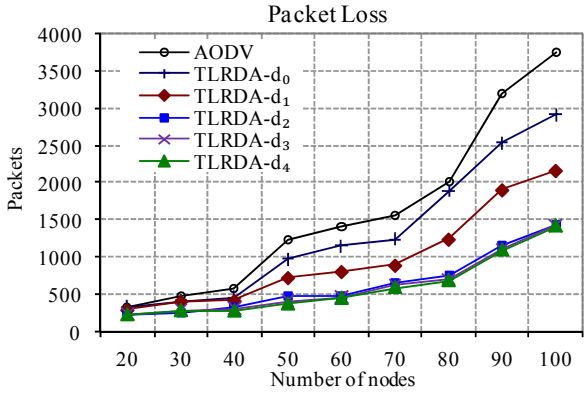


Figure 4. Packet loss verses different number of nodes for networks of 10 communication sessions and 15m/s as maximum speed.

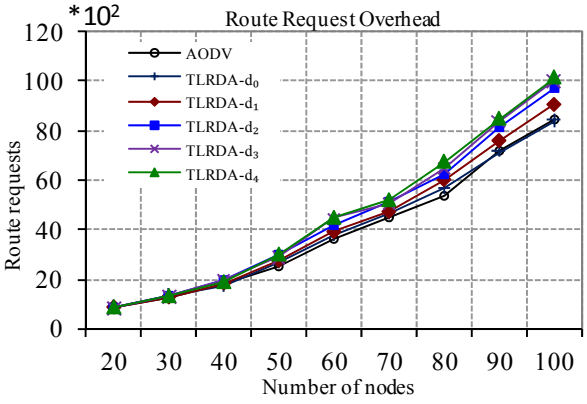


Figure 5. Route request overhead verses different number of nodes for networks of 10 communication sessions and 15m/s as maximum speed.

Effect of mobility: The value of the maximum speed where 2m/s used as slow speed and 15m/s is fast speed. The end-to-end delay in TLRDA-D is reduced com can be 2, 5, 7, 10, 13, or 15m/s with networks of 70 nodes and traffic load of 10 communication sessions where 2m/s used as slow speed and 15m/s is fast speed. The end-to-end delay in TLRDA-D is reduced compared to AODV for different maximum speed as in Figure 9 where discovery time increases in both TLRDA-D and AODV with fast speed because speed affects routes and may result in broken links. This figure reveals the difference in the end-to-end delay among all five instances of TLRDA-D where TLRDA-d_2, TLRDA-d_3, and TLRDA-d_4 reduce end-to-end delay more than TLRDA-d_0 and TLRDA-d_1.

TLRDA-D reduces packet loss compared to AODV as shown in Figure 8. Packet loss increases with faster movements in both algorithms. TLRDA-D improves packet loss over AODV by 14% to 87% in slow speed and by 21% to 62% in fast speed. Moreover, these packets include route requests which increases route request overhead in TLRDA-D over AODV as shown in Figure 9.

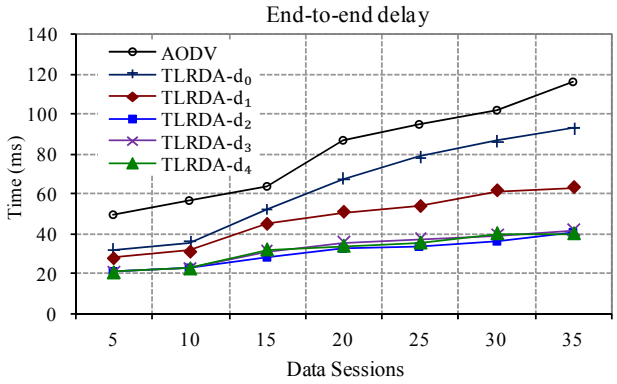


Figure 6. End-to-end delay versus traffic load with a network70 nodes and 15m/s as maximum speed.

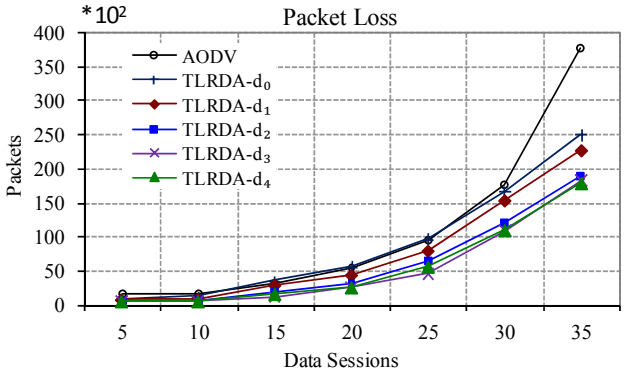


Figure 7. Packet loss versus traffic load with a network70 nodes and 15m/s as maximum speed.

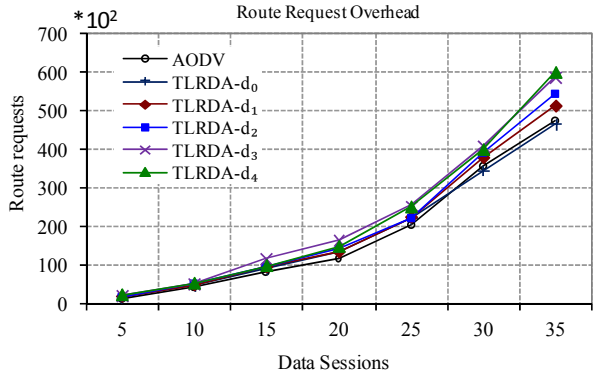


Figure 8. Route request overhead versus traffic load with a network70 nodes and 15m/s as maximum speed.

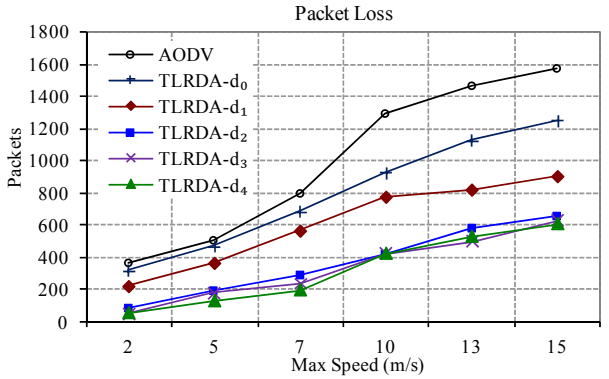


Figure 10. Packet loss versus maximum speed in networks of 70 nodes and 10 communication sessions.

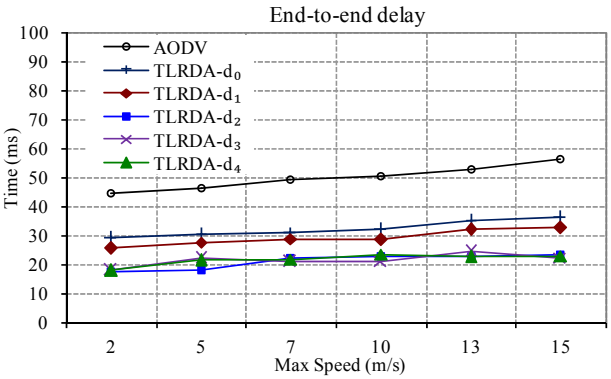


Figure 9. End-to-end delay versus maximum speed in networks of 70 nodes and 10 communication sessions.

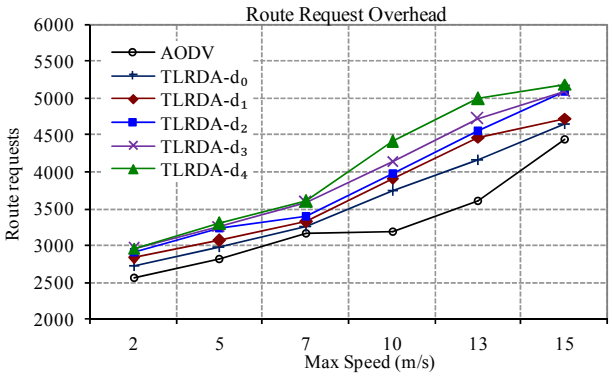


Figure 11. Route request overhead versus maximum speed in networks of 70 nodes and 10 communication sessions.

Both algorithms have almost the same number of transmitted route request; so extra route requests received in TLRDA-D might be duplicate copies but were dropped because of congestion or collision. Furthermore, the number of saved packets is greater than the increment in route requests overhead where the minimum difference ranges from 8% to 70% in slow speed and from 16% to 45% in fast speed. The extra saved packets can be any kind of packets which might be useful but dropped in AODV due to many reasons i.e. contention, congestion or collision. These saved packets in TLRDA-D have a good impact on network performance.

In summary, TLRDA-D reduces discovery time, packet loss, and end-to-end delay over AODV. However, it increases route request lifetime in justifiable manner. The best delay function would be a linear one. In particular, for the considered scenarios in our experimental study the doubling function $d_2 = 2LP_r$ gave the best performance among all scenarios performed in this study. It is worth mentioning that TLRDA-D reduces end-to-end delay despite the fact that it works by delaying, by definition, route request within their source node's beyond-neighbourhood region.

4. Conclusions

When on-demand routing algorithms for MANETs run applications that exhibit traffic locality, the route discovery process can be improved by utilising the traffic locality concept. We introduce a traffic locality oriented route discovery algorithm with delay, TLRDA-D. It works by establishing a neighbourhood that includes the most likely destinations for a particular source node. The source node broadcasts the route request without adding any delay within its neighbourhood boundary. In an effort to improve the route discovery process for MANETs that exhibit traffic locality. This adaptive route discovery algorithm gradually build up the node neighbourhood as a region, with the ability to change, centred at the source node and expected to contain most of the members of its working set. Furthermore, TLRDA-D adds a delay to route requests travelling within their beyond-neighbourhood region to reduce channel contention which reduces the discovery time of other route requests. One of the main advantages of TLRDA-D is improving route discovery process which improves the end-to-end delay as it generates less channel contention throughout the network

which reduces packet loss. We have analysed TLRDA-D using simulation to study the affect of adding a delay to route request propagation and to decide on the proper amount of delay to be added. The simulation analysis showed that when TLRDA-D uses twice the locality parameter as a delay, it gave the best improvement among the experimented scenarios.

5. References

[1] S. Murthy and B. Manoj, "Ad hoc wireless networks: Architectures and protocols," Prentice Hall, 2004.

[2] A. Tanenbaum, "Computer networks," Pearson Education, 2003.

[3] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks," Ad Hoc Networks, Vol. 2, No. 1, pp. 1−22, 2004.

[4] C. Adjih, T. Clausen, P. Jacquet, *et al*., "Optimized link state routing protocol," The Internet Engineering Task Force, IETF, RFC 3626, 2003.

[5] S. R. Das, R. Castaneda, Y. Jiangtao, *et al*., "Comparative performance evaluation of routing protocols for mobile ad hoc networks," pp. 153−161, 1998.

[6] D. Johnson, D. Maltz, and Y. -C. Hu, "The dynamic source routing protocol for mobile ad hoc networks (DSR)," The Internet Engineering Task Force, IETF, draft-ietf-manet-dsr-09.txt, April 2003.

[7] C. Perkins, E. Belding-Royer, and S. Das, "AODV ad hoc on-demand distance vector routing," The Internet Engineering Task Force, IETF, RFC 3561, July 2003.

[8] Z. J. Haas, M. R. Pearlman, and P. Samar, "The Zone Routing Protocol (ZRP) for ad hoc networks," IETF MANET Working Group, INTERNET-DRAFT, July, 2002.

[9] P. Denning, "The working set model for program behavior," Communications of the ACM, Vol. 11, No. 5, pp. 323−333, 1968.

[10] M. Shikharesh and B. B. Richard, "Measurement and analysis of locality phases in file referencing behaviour," Proceedings of the ACM SIGMETRICS Joint International Conference on Computer Performance Modelling, Measurement and Evaluation, Raleigh, North Carolina, United States, 1986.

[11] P. Denning, "The locality principle," Communications of the ACM, Vol. 48, No. 7, pp. 19−24, 2005.

[12] C. Kozierok, "The TCP/IP guide," 1st Edition, No Starch Publishing, 2005.

[13] A. Silberschatz, P. Galvin, and G. Gagne, "Operating systems concepts," 7th Edition, John Wiley & Sons, 2005.

[14] F. Borgonovo, "ExpressMAN: Exploiting traffic locality in expressnet," IEEE Journal on Selected Areas in Communications, Vol. 5, No. 9, pp. 1436−1443, 1987.

[15] M. Prasant and K. Srikanth, "Ad hoc networks: Technologies and protocols," Springer-Verlag New York, Inc., 2004.

[16] J. Y. Li, C. Blake, D. S. J. De Couto, H. I. Lee, and R. Morris, "Capacity of ad hoc wireless networks," Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, Rome, Italy, 2001.

[17] M. Al-Rodhaan, L. Mackenzie, and M. Ould-Khaoua, "A traffic locality oriented route discovery algorithm for MANETs," Ubiquitous Computing and Communication Journal (UBICC), Vol. 2, No. 5, pp. 58−68, 2007.

[18] M. Mosko and J. Garcia-Luna-Aceves, "Performance of group communication over ad hoc networks," Proceedings of the IEEE International Symposium Computers and Communications ISCC, Italy, pp. 545−552, 2002.

[19] K. Fall, "NS notes and documentation," in The VINT Project, 2000.

[20] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in Mobile Computing, I. A. Korth, Ed., Kluwer Academic Publishers, Norwell, MA, Vol. 353, pp. 153−181, 1996.

[21] F. Bai, N. Sadagopan, B. Krishnamachari, *et al*., "Modeling path duration distributions in MANETs and their impact on reactive routing protocols," IEEE Journal on Selected Areas in Communications, Vol. 22, No. 7, pp. 1357−1373, 2004.

[22] X. Y. Hong, M. Gerla, G. Y. Pei, and C.-C. Chiang, "A group mobility model for ad hoc wireless networks," Proceedings of the 2nd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Seattle, Washington, United States, 1999.

[23] F. Bai, N. Sadagopan, and A. Helmy, "IMPORTANT: An evaluation framework to study the impact of mobility patterns on routing in ad-hoc NETworks," University of Southern California, 2005.

Int. J. Communications, Network and System Sciences, 2009, 3, 169-247
doi:10.4236/ijcns.2009.23023 Published Online June 2009 (http://www.SciRP.org/journal/ijcns/).

Mobility Trigger Management: Implementation and Evaluation

Jukka MÄKELÄ, Kostas PENTIKOUSIS, Vesa KYLLÖNEN
VTT Technical Research Centre of Finland, Oulu, Finland
Email: {*jukka.makela, kostas.pentikousis, vesa.kyllonen*}*@vtt.fi*
Received December 11, 2008; *revised April* 21, 2009; *accepted April* 23, 2009

ABSTRACT

Modern mobile devices have several network interfaces and can run various network applications. In order to remain always best connected, events need to be communicated through the entire protocol stack in an efficient manner. Current implementations can handle only a handful of low level events that may trigger actions for mobility management, such as signal strength indicators and cell load. In this paper, we present a framework for managing mobility triggers that can deal with a greater variety of triggering events, which may originate from any component of the node's protocol stack as well as mobility management entities within the network. We explain the main concepts that govern our trigger management framework and discuss its architecture which aims at operating in a richer mobility management framework, enabling the deployment of new applications and services. We address several implementation issues, such as, event collection and processing, storage, and trigger dissemination, and introduce a real implementation for commodity mobile devices. We review our testbed environment and provide experimental results showcasing a lossless streaming video session handover between a laptop and a PDA using mobility and sensor-driven orientation triggers. Moreover, we empirically evaluate and analyze the performance of our prototype. We position our work and implementation within the Ambient Networks architecture and common prototype, centring in particular on the use of policies to steer operation. Finally, we outline current and future work items.

Keywords: Triggering, Mobility Management, Mobile Networks, Handover, Cross-Layer Information Management

1. Introduction

Modern mobile devices, such as smartphones, Internet tablets and PDAs, have several network interfaces and can run various network applications, like web browsers, email clients, and media players. Indeed, it is becoming common that said devices can take advantage of wireless LAN, PAN and cellular connectivity, and we expect that in the coming years mobile WiMAX will be supported as well. In such a multiaccess environment, mobility management support for both horizontal and vertical handovers should be one of the basic functionalities in future devices. Moreover, in order to allow a mobile device to remain always best connected, several events need to be communicated through the entire protocol stack, as we explain in the following section. Nevertheless, current implementations of state-of-the-art mobility management protocols, such as Mobile IP [1] or Host Identity Protocol [2]), can only handle a small set of event notifications that may lead to mobility management actions, including handover execution.

In this paper, we argue for a novel mobility trigger management framework that can handle a much larger set of notifications related to events originating not only from the lower layers of the protocol stack (physical, data link, and network), but also from the upper layers enabling the efficient use of cross-layer information for mobility management. This framework needs to be open, flexible, with low overhead, and incrementally deployable. After describing the main parts of the architecture, we present the implementation of such a framework, which allows mobile devices to manage, on the one hand, conventional mobility events, such as the availability of a new network access, received signal strength indications

(RSSI), network capacity load and, on the other hand, higher level events, such as security alerts, policy violations, end-to-end quality of service deterioration, and network access cost changes. In our framework, event sources can deliver notifications to interested applications and other system entities used in a standardized manner. We will refer to these standardized notifications as triggers in the remainder.

The main elements of our trigger management framework are detailed in [3,4], and include the entities which generate the events (producers) and entities that use the trigger information (consumers). Our trigger management framework is capable of collecting event information from various producers through a specific collection interface. The collected events are then processed and converted into a unified trigger format, described in Section 5, and distributed to interested consumer entities. A trigger consumer can be any entity implementing the collection interface and can be located in the same or in different node in the network. It should be noted also that a same entity can act both as a producer and a consumer.

In this paper we concentrate on the evaluation of the implementation of our framework in the VTT Converging Networks Laboratory. Indeed this paper demonstrates the feasibility of our designed framework over a real testbed network. The concept and architecture behind our framework with some analysis to the similar existing concepts are also summarized below.

The rest of this paper is organized as follows. Section 2 introduces the fundamental elements of our framework for managing triggers, reviews the related work in this area and motivates our evaluation. Section 3 presents our implementation of the triggering framework and Section 4 discusses the role of policies and rules in the system design. Results from our experimental lab evaluation are presented in Section 5. Related work is discussed in Section 6, and Section 7 concludes the paper.

2. A Framework for Managing Mobility Triggers

After surveying the relevant literature (see, for example, [5,6−10]), and based on our own expertise, we identified more than one hundred different types of network events related to mobility management. We cluster triggers, regardless of the underlying communication technology, based on groups of events related to changes in network topology and routing, available access media, radio link conditions, user actions and preferences, context information, operator policies, quality of service (QoS) parameters, network composition [11], and security alerts.

Figure 1 illustrates six different trigger groups as boxes. The "offshoots" on top point to example triggers belonging to each group. The rightmost group includes representative link layer "up/down" triggers (irrespective of the radio access technology). The leftmost group includes triggers originating from the application layer. In this example, certain triggers originate from the node ("System Resources") while others originate from the network ("Macro Mobility"). The "origin" corresponds to the entity that produces the trigger, for example, the radio access component. An advantage of our grouping approach is that it allows us to detect relations between otherwise disparate triggers. This prevents the generation of excessive transient triggers based on, for example, the same root-cause event, such as a link outage, and reduces the number of events that need to be processed.

Event sources need to be able to deliver notifications to interested applications and other system entities in a uniform, concise, and standardized manner. This approach simplifies notification handling considerably, while guaranteeing sufficient diversity for event separation and classification. In order to manage and efficiently propagate triggers originating from a variety

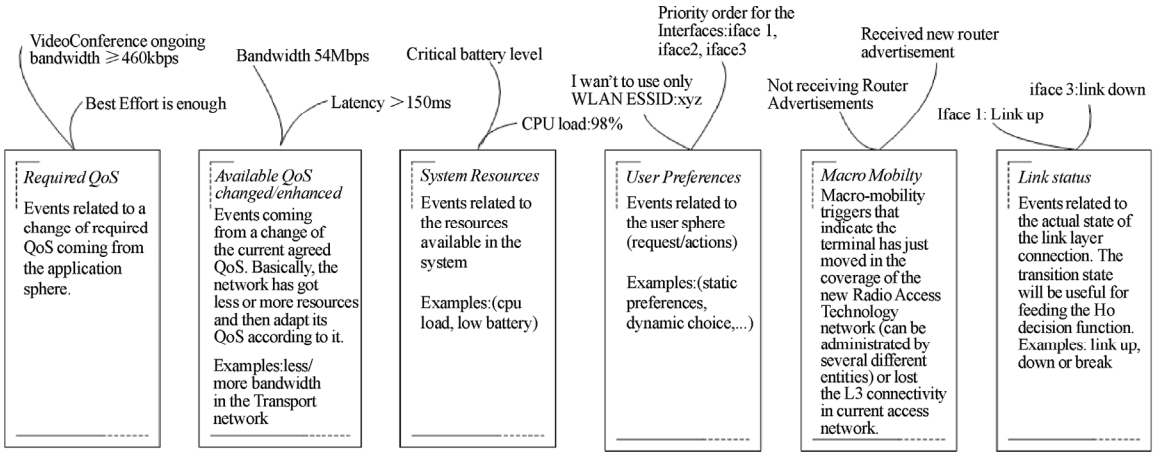


Figure 1. An example of trigger groups.

of sources we developed a trigger management framework, which we call TRG. TRG lays the foundation upon which sophisticated handover (HO) operations can be performed. We aim at establishing an extensible frame- work where new sources of triggers can be defined and included in a system implementation as necessary. Note that this is quite different from other, in our opinion, more closed and specific approaches, such as the one followed in the IEEE 802.21 [12] working group. On the surface, both TRG and the IEEE 802.21 Media Independent Handover Services standard seem quite similar, aiming to improve mobility management performance. However as we argue in [13] the mechanisms and services introduced by the IEEE standard do not include dynamic information elements and any extensions will have to be introduced with lengthy standardization procedures in the future. Moreover, triggers cannot originate from the higher layers of the protocol stack, and system level events are simply out of scope of IEEE 802.21. Finally, 802.21 provides services to command and use the lower layer information to enable seamless handovers and multiaccess, which is not in the domain of TRG, but of the mobility management protocol. Last but not least, TRG is designed to handle much more event sources than MIHF. It is important to highlight that TRG provides the means to disseminate and filter mobility-related information between one or more event sources and several trigger consumers but that HO decisions are still the responsibility of the mobility management protocol, say, Mobile IP [1] or HIP [2]. TRG can also provide hints about moving the communication endpoint from one device to another, as explained in Section 5.

A central part of the design is designating different system entities as producers and consumers of triggers. Policies, described in Section 4, are handled by the Policy Manager. For communicating with different entities, TRG exposes three service access points (SAP). Event sources use the Producer SAP, to register events and emit notifications to TRG when changes occur. Consumers use another SAP, to subscribe with TRG and receive triggers in a single format when they become available. Finally, the Policy Manager uses another SAP to inform TRG about policies. Internally, TRG implements a local trigger repository and functional blocks for processing triggers.

Consumers must state their need to receive triggers and can choose to stop receiving them anytime. For example, the Mobile IP daemon can receive all triggers related to link layer events, but opt to receive only the upper-layer triggers associated with security or policy violations. In the former case, such a consumer takes advantage of the trigger grouping functionality; in the latter, it additionally requests trigger filtering. Consumers can use these triggers to generate their own and, thus, serve as an event producer for other entities. We expect that TRG will be used to guide HO decision making and execution. In particular, consumers can use triggers to derive whether the mobile device is moving within a single network or it is crossing different access technology boundaries, and whether the addressing scheme, trust and provider domains should be changed accordingly.

3. Architecture and Implementation

The core implementation of TRG has three major components: triggering event collection, trigger processing, and the trigger repository [3,4]. Triggering events collection receives events from various sources in the network system via the trigger collection interface. New triggers can be introduced in a straightforward manner by implementing the trigger event collection functionality and supporting the trigger collection interface. The latter allows sources to register their triggers and to make them available to consumers. A specific TRG implementation may contain several event collectors, which may be distributed, and are responsible for collecting different types of events. The trigger repository is designed to meet the stringent requirements placed by mobility management, but can be used to store non-mobility triggers as well. The basic primitives include adding, removing, updating, and disseminating triggers in a standardized format. Each stored trigger has an associated lifetime and is removed automatically once its time-to-live (TTL) expires.

The need for different event collectors arises from the fact that the origin of an event source can be a hardware device, a system component implemented in kernel space, or an application implemented in user space. For example, each device driver could implement its own event collection functionality, which would be capable of handling triggering events produced by the specific device only. Moreover, sources can also be located in the network such as at active network elements or at the user's home network. Finally, a particular TRG implementation can act as a consumer to another TRG located in a different node. Thus, orchestrating the collaboration of, perhaps, several collection entities is needed in order to efficiently gather a larger amount of events.

Having dedicated collectors for different event sources enables the use of TRG in different operating systems as well. The collector can format the events to the format that TRG understands and there is no need to modify the core of TRG functionality; instead the collector can be modified as necessary. This is also one of the key points in the architectural design of TRG that enables it to handle cross-layer information by having a collector at different layers as needed. For example TRG can get simi-

lar information considering the connectivity in FreeBSD through a collector that uses Route Socket and in Linux through a similar collector using RTnetlink socket but obviously these collectors need to have their own implementation. The core of TRG could be implemented in kernel space for performance reasons and allowing for direct access to lower layer information. On the other hand, TRG can be implemented in application space allowing for greater flexibility and easing implementation and code evolution. The prototype described in this paper follows the latter approach. Of course, certain event collectors will have to be implemented closer to the lower layers in the future.

The event sources are connected with TRG via producer SAP, as described also in section 2. The performance of the event collectors is obviously very important. They need to be fast enough to react to all different events, but the collector implementation itself is not part of the TRG framework architecture. TRG provides the interfaces to connect different event producers with the possible consumers by defining the SAP's between TRG and them. TRG core functionality per se provides the mechanisms for distributing, filtering and handling the policies for the whole system of the mobility event handling, but the collectors are out of scope of this paper. Figure 2 illustrates the TRG framework with the different event producers and consumers.

After events are collected from the producers, they are handed over to the trigger processing engine which is responsible for time-stamping and reformatting triggers (if necessary), and assigning them to the appropriate group. Consumers can subscribe by specifying a set of triggers (and, optionally, filtering rules) and are expected to unsubscribe when they do not wish to receive them any longer. For each consumer subscription, TRG makes sure that filters are grammatically and syntactically correct, and accepts or rejects the subscription. Basic rules can also be used as building blocks for crafting more sophisticated rules.

4. Policies and Rules in TRG

TRG supports the application of different triggering policies, defined as a set of classification, filtering, trust, and authorization criteria/rules. This allows our implementation to enforce a different policy at different times or wh en the node operates in different contexts. The availability of a system-wide policy and consumer- supplied filters lies at the centre of our TRG design. These two are orthogonal, providing flexibility and adaptability.

System policies ensure that only designated consumers can receive certain groups or types of triggers. For example, a node may operate under different policies regarding network attachment depending on whether the user is on a business or a leisure trip. Policies can also establish different trigger classification and groupings in different contexts and are typically stored in a separate repository, accessible to the TRG implementation. Filters allow a consumer to focus a trigger subscription. For example, a monitoring application may be interested in receiving all network utilization measurements, while a VoIP application may be interested in receiving a trigger only when utilization exceeds a certain threshold and the user is in a call. In fact, a VoIP application can even opt to be an intermittent trigger consumer, subscribing and unsubscribing to receive certain triggers solely when needed.

Our TRG implementation uses access control policies to define:

- Which producers are allowed to register and send triggers to TRG. Producers are identified by the trigger IDs they register, and can be chosen on a system basis. For example, a policy allows only specific producers to register with TRG.

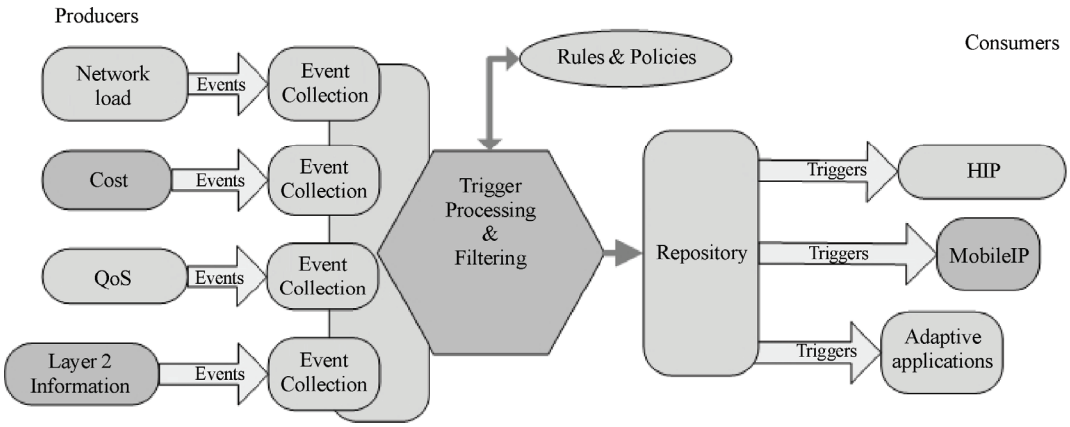


Figure 2. TRG architecture.

- Which consumers are allowed to subscribe to triggers. Policies can be very specific, prescribing which consumers can receive certain triggers and from which producers. Consumers are identified by their locator (typically a host address). For example, in our proof of concept implementation described in Section 5, we can enforce a policy that dictates that triggers from producer with ID=50 are allowed to be subscribed only from "localhost" entities.

The Policy Manager applies access control using policies described in XACML (OASIS eXtensible Access Control Markup Language) [14]. Figure 3 illustrates which Policy Manager functions are called when a producer registers or a consumer subscribes. Typically the decision on whether to allow producer registrations and consumer subscriptions is made immediately based on the system policies and the result is returned to the initiating entity. In the case where a consumer attempts to subscribe to all triggers, the decision may be deferred for when triggers become available. That is, the subscription for "all triggers" effectively becomes a subscription for "all triggers allowed", when system policies dictate so. In our current prototype implementation, policies are described using access control lists read from a configuration file. Policies also define which consumers are allowed to subscribe and for which trigger.

5. Results

We tested our user-space C++ implementation of TRG on laptops running FreeBSD release 6.1, Linux Fedora Core 3 with kernel 2.6.12 and Windows XP, and on a PDA running Linux Familiar v.0.8.4 with kernel 2.4.19. Architecture design with the possibility to use separate event collectors in different environment, as discussed in Section 3, makes our TRG implementation portable and is currently being integrated in several prototypes, including the Ambient Networks [15] prototype [16–18]. For communication between producer, TRG and consumer a Web Service XML-based communication on top of HTTP was used. In this integrated prototype, TRG takes care of the delivery of all mobility-related events. Events were formatted according to the unified trigger format shown in Table 1.

In previous work we presented a proof-of-concept testbed and demonstrated the feasibility of the concepts governing our TRG implementation. These preliminary validation results are summarized briefly in Subsection 5.1; further details are available in [4]. Subsection 5.2 presents the first detailed results of our stress-test empirical evaluation of TRG in the lab.

5.1. Proof of Concept Validation

In [4,19], TRG was employed to enable streaming video session handovers between different mobile devices. In the scenario, the user starts watching a video streamed to his laptop. His GNU/Linux PDA is nearby and the user decides to move to another room but would like to keep watching the video on the way. The commercial, off-the- shelf (COTS) PDA is augmented with a multi-sensor device (detailed in [20]), which was extended to provide "device orientation" triggers. For example, when the user picks up the PDA, a "vertical orientation" trigger is produced, initiating a session HO from the laptop to the PDA. The two devices have to coordinate and arrange for the transfer of the video streaming session. A successful session handover allows the user to receive the streaming video on the PDA over the WLAN seamlessly. The user can also explicitly initiate a session HO by pressing a PDA button. In this example, TRG handles triggers associated with mobility,

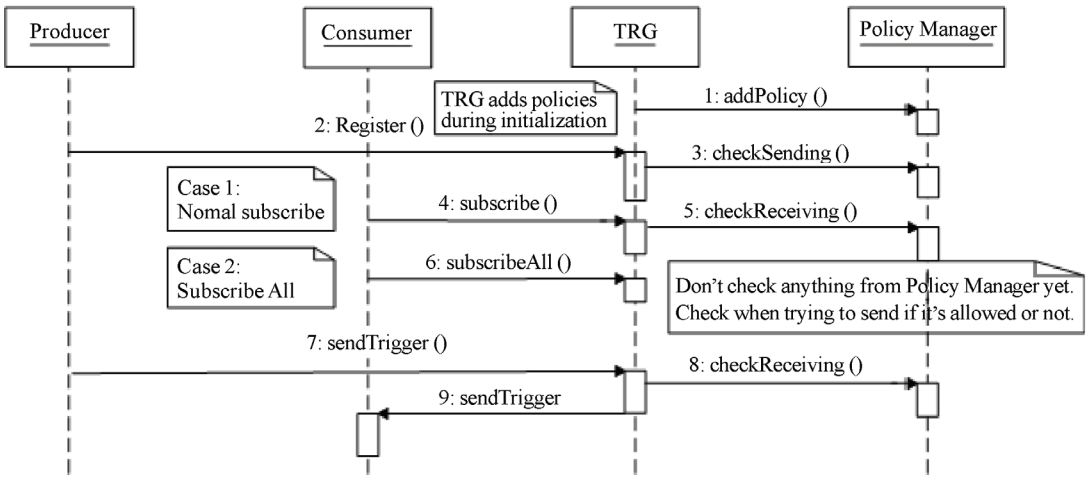


Figure 3. TRG-policy manager message sequence diagram.

Table 1. Trigger format.

Trigger data member	Type	Description
id	integer	Trigger identifier, same as producer identifier. Maps producer name to identifier.
type	integer	Specific to the trigger identifier. Mapping producer information to type.
value	std:string	Specific to trigger type.
timestamp	time_t	Time that a trigger enters the TRG repository.

orientation, and user preferences, keeping the video flowing smoothly while changing the communication end-point. Two logical topologies were evaluated in our lab. First, all devices are connected using IEEE 802.11 in ad-hoc mode, as if the user streamed a video from his digital collection at home. Second, the video streaming server is located in a different network, as would be the case when watching a video from a service provider over the Internet. For both setups in our lab proof-of-concept validation, we stream a 10-minute video encoded at 576 kb/s over UDP. At t = 3 min, a session HO from the laptop to the PDA is triggered, and at t = 7 min, the session is "moved" back to the laptop.

We captured all traffic traces during the experiments using tcpdump and cross-checked all packet IDs sent by the video server with the packet IDs received at the laptop and PDA video clients to confirm that no packet losses occurred. Moreover, the effect of TRG signalling and the actual session handover on packet delay is negligible, compared to packet delays before and after the session handover. Figure 4 illustrates the packet inter- transmission times as recorded by tcpdump at the streaming server and the packet inter-arrival times at (a) the receiving laptop and (b) the receiving PDA. On the left-hand side, the packet inter-arrival time measured at the streaming server, laptop and the PDA during the delivery of the 10-min video stream are shown. The band around 50 ms indicates that the packets are sent and received in an orderly manner. We note a small number of inter-arrival times outside this band. The vast majority of inter-arrival times do not exceed 150 ms; only a handful of packets out of more than 12000 exceed this threshold. On the right-hand side of Figure 4, we zoom in at around t=3 min when the first session HO is triggered from the laptop to the PDA. As the figure illustrates, only a few packets had >0.1 s inter-arrival time. These results are very promising, despite the fact that this is a prototype implementation, especially when taking into consideration that the PDA was running the video client and captured packets using tcpdump throughout the experiment leaving few spare system resources available.

This paper focuses on the empirical validation and evaluation of TRG. The theoretical aspects (scalability, security, reliability) have been partly addressed elsewhere [21] and further analysis is also part of our future work agenda. It is important to note that these set of experiments go beyond showcasing the concept of TRG-assisted session HOs. This is simply a particular application of triggers leading to a HO. Instead, we emphasize that these experiments aim at assessing the feasibility of introducing a TRG implementation in small COTS handheld devices, a result which was not warranted when we embarked in developing TRG.

5.2. Experimental Evaluation

Since we conducted the experiments presented in the previous subsection, we continued the development and evaluation of TRG and used an updated implementation of TRG enhanced with web service interfaces and ran tests where we submitted 100000 triggers from several sources to TRG and delivered those to different consumers. We consider two test cases, with the aim of quan-

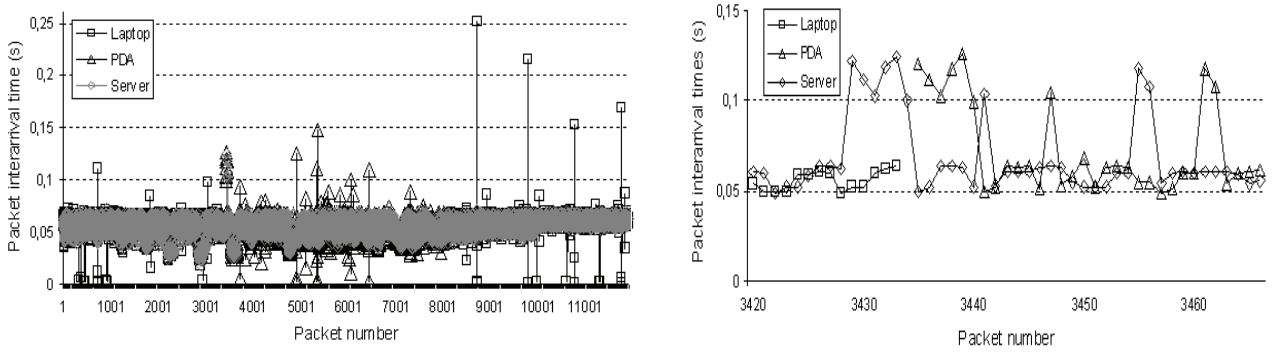


Figure 4. Experimental results when triggering a session HO.

tifying TRG performance under stress (and perhaps clearly unrealistic) conditions. Test Case 1 employs n producers connected with m consumers via TRG. During the test, each producer sends 100000 back-to-back triggers and all triggers are distributed to all m consumers. This means that TRG needs to process $n \times 10^5$ triggers and deliver $n \times m \times 10^5$ triggers. On the left-hand side of Figure 5, we illustrate an example case where n=3 producers A, B and C each send 100000 triggers, with trigger IDs 51, 52 and 53, respectively, to m=4 consumers (labelled I, II, III, IV). That is, in this particular scenario, each of the four consumers will receive 300000 triggers from TRG.

Table 2 shows the number of delivered triggers with average processing times in milliseconds for each trigger received by TRG from the producers in Test Case 1. In this case, only the number of consumers has a significant effect on the processing time of each trigger. This indicates that TRG can cope with several registered producers even when there is no subscribed consumer from certain producers. Moreover, the average trigger processing time is only few milliseconds per subscribed consumer in this stress test of the prototype implementation.

Since there are several possible scenarios about how triggers are distributed between producers and consumers we made also a Test Case 2 setup, illustrated in the right-hand side of Fig. 5, where each consumer has only one dedicated producer. This means that TRG needs to process $n \times 10^5$ triggers and deliver $m \times 10^5$ triggers. If there are more producers than consumers, triggers will be distributed evenly between the available consumers. As mentioned above, all tests were made using a C++ implementation of TRG with a web service interface towards producers and consumers. We used a laptop with an Intel Pentium M 1.70 GHz PC with 1 GB RAM, running FreeBSD release 6.1 in the tests reported in this subsection.

Figure 6 shows the total processing time of Test Case 1, with and without employing the TRG filtering mechanism. It can be seen that when the number of the consumers and producers increases, so does the total processing time. This is expected since the number of processed triggers is increasing when adding more consumers and producers. The costs of adding consumers and producers are both linear. But the cost of adding consumers is greater than the cost of adding producers. For example when comparing the calculated slope $k = \Delta y/\Delta x$ of the curves of total processing time, with and without filtering, we see that the processing time increases faster the more consumers are introduced (slope of the curve with one consumer k = 177,6 and with 5 consumer k = 454,9 in Test Case 1 without filtering), this can be explained as a cost of the duplication of triggers because the number of triggers that have to be duplicated and delivered to consumers increases when adding more consumers. Anyhow this does not increase the average processing time of one trigger. The number of producers has also effect to the total processing time, but not as much as the number of consumers.

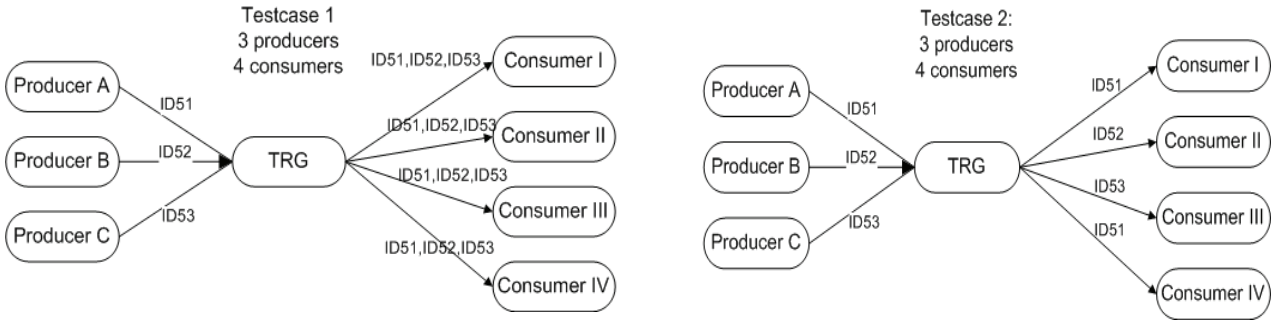


Figure 5. Triggers in Test Cases 1 (left) & 2 (right).

Table 2. Total number of delivered triggers and average processing time (in ms) per trigger in Test Case 1.

	Number of Producers				
Number of Consumers	1	2	3	4	5
1	100k, 1.7 ms	200k, 1.7 ms	300k ,1.8 ms	400k, 1.7 ms	500k, 1.8 ms
2	200k, 2.3 ms	400k, 2.5 ms	600k, 2.4 ms	800k, 2.5 ms	1000k, 2.4 ms
3	300k, 3.2 ms	600k, 3.2 ms	900k, 3.2 ms	1 200k, 3.1 ms	1500k, 3.3 ms
4	400k, 3.7 ms	800k, 3.8 ms	1200k, 3.8 ms	1600k, 3.8 ms	2000k, 3.8 ms
5	500k, 4.5 ms	1000k,4.6 ms	1500k, 4.7 ms	2000k, 4.7 ms	2500k, 4.5 ms

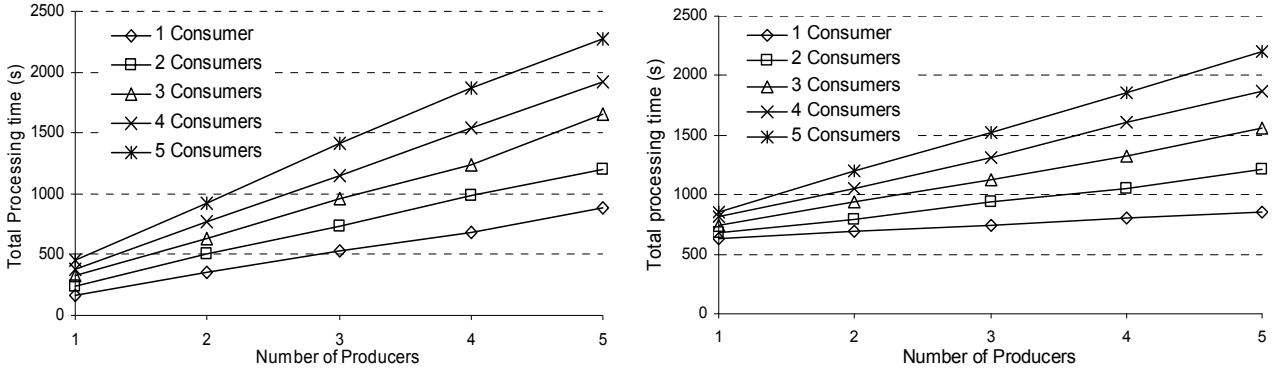


Figure 6. Total processing time in Test Case 1 without (left) and with (right) filter processing.

We also evaluated the cost of using the filtering function of TRG. With Test Case 1 we had all five producers registered and each one was sending 100000 triggers. It follows that TRG was receiving total of 500000 triggers during the test. The right-hand of Figure 6 shows the total processing times when the filtering mechanism was used. When there is one producer, the triggers from the other four producers are filtered away, and the triggers from the sole producer are duplicated and delivered to all four consumers. In the case with two producers the triggers from three producers are filtered away, and so on. The results show that it takes more time to process all triggers but this is not caused by the filtering mechanism itself. When comparing the total processing times, in the case where triggers from 1 producer are delivered to consumers in Figure 6, the total processing time is increased when the filtering mechanism is used, but this is because now there are five times more triggers received by TRG than in the case without the filtering mechanism, since all five producers are sending 100000 triggers all the time during the test. When the filtering mechanism is not used, the number of producers is controlled by making a new registration per producer.

To further quantify system behaviour when filtering is employed, we consider Test Case 2. When evaluating the filtering function in Test Case 2, each consumer had a filtering rule that was true for all triggers, allowing the distribution of all triggers to the subscribed consumers. By having this "receive all triggers" rule we were able to test the effect of the filtering mechanism, since every time a trigger is produced, TRG needs to run the filtering code before disseminating the trigger to consumers even though none of the triggers are in practice going to be filtered away. The purpose was to test the effect and cost of running the filtering function. The TRG filtering mechanism per se does not have a significant effect on the overall processing time, especially when compared to the effect of increasing the number of consumers. When comparing the processing times in Test Cases 1 (Table 2) and Test Case 2 (Figure 7) we see that the duplication of each trigger to every consumer, needed in Test Case 1, increases processing times. In Test Case 2, when the number of producers and consumers are equal, the difference of the processing times can be measured in microseconds, since now there is no need to duplicate triggers.

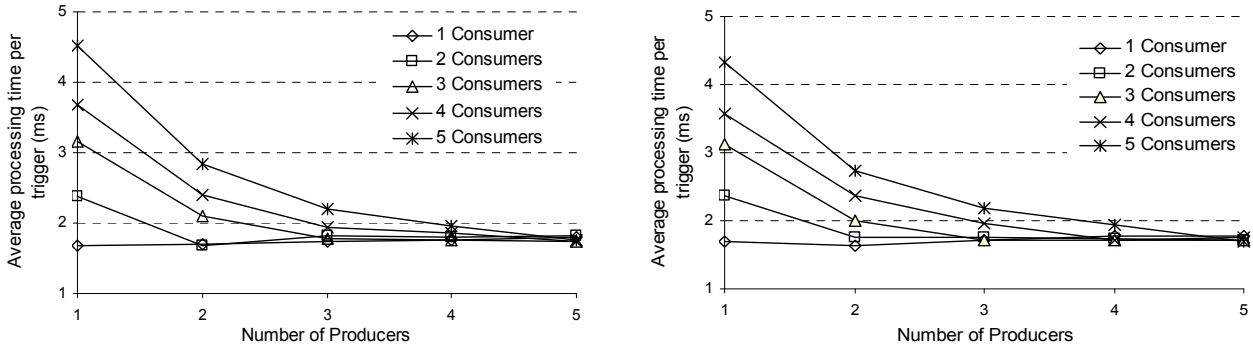


Figure 7. Average processing time in Test Case2 per trigger without (left) and with (right) filtering.

The test and evaluation cases presented in this Section showed that it is in fact the duplication of triggers and number of messages that have a biggest effect on the processing time of triggers. It can be seen in Figure 7, for n = 5 producers and m = 5 consumers, that the processing time does not depend on the number of consumers. There is no duplication of triggers in this case either.

The feasibility of using TRG to process, filter and disseminate a very large amount of triggering events was shown in practice. Each case showed that the processing time of one trigger does not increase, even when processing a huge amount of triggers. Although the stress-test cases are clearly unrealistic, they demonstrate that using TRG does not cause any major delays to handover times. On the contrary, TRG enables handover decision making mechanisms to react more rapidly and to larger set of events. It is also important to note that the TRG filtering mechanism does not have a major effect to processing times and this allows the handover decision making mechanisms to react faster to relevant events. Although the filtering mechanism can be used for the pre-decision about which events are to be collected, the handover decision per se is left to separate mechanisms with the decision algorithm. It was also shown that the cost of adding more consumers and producers increase processing times linearly and the cost of using filtering has only a marginal effect on the processing times. Of course the more triggers there are, the more total processing time is needed for processing and disseminating all triggers. However, by implementing grouping and classification of triggers [4] and having mechanism, e.g. in the TRG source for prioritizing trigger delivery which allows critical triggers to be processed and distributed faster, TRG is ready to process the triggering events.

6. Related work and Discussion

Previously published work [7–9] shows the benefits of using event information, for example, to proactively perform a handover in order to maintain QoS levels. Our goal is to define a framework that supports the event collection and processing, and trigger distribution possibly from hundreds of different sources. We concur with Vidales *et al*. [7] that in heterogeneous network environments several sources of events and context information should be consulted in order to achieve seamless connectivity and develop swift mobility management mechanisms. Furthermore, earlier work in other event/notification systems [22,23], which introduces mechanisms on how to implement such systems, along with the evaluated event generation cases is very encouraging and complimentary to our effort in defining TRG as a specialized notification system for mobility-related events which originate from the entire protocol stack.

The IEEE 802.21 Media Independent Handover (MIH) Services [12] working group is standardizing an information service that will facilitate media independent handovers. The scope of the IEEE 802.21 standard is to provide a mechanism that provides link layer intelligence and other related network information to upper layers to optimize handovers between heterogeneous IEEE 802 systems and facilitates HOs between IEEE 802 and cellular systems. IEEE 802.21 assists in HO Initiation, Network Selection and Interface Activation. The purpose is to enhance the experience of mobile device users. The standard supports HOs for both stationary and mobile users. For mobile users, HOs are usually needed when the wireless link conditions change. For stationary users, HOs are needed when the surrounding environment changes. Both mobile node and network may make decisions about connectivity. The HO may be conditioned by measurements and triggers supplied by the link layers on the mobile node. The IEEE 802.21 standard defines services that enhance HO between heterogeneous access links. Event service, Command service and Information service can be used to determine, manage and control the state of the underlying multiple interfaces. By using the services provided by MIH Function users, like Mobile IP, are able to better maintain service continuity, service adaptation, battery life conservation, networks discovery and link discovery. MIH Function also facilitates seamless handovers between heterogeneous networks.

The IEEE 802.21 Event Service has common characteristics with our TRG design but does not prescribe a particular implementation and stops short of allowing upper-layer entities to provide events that can drive a HO. It was also impossible to compare the performance of the implementations since no MIH implementation was available when these tests were performed. Our approach emphasized standardized ways for consumers to receive trigger from a variety of sources. TRG framework is also fully implemented and tested in a laboratory environment with several operating systems. Easy application registration to TRG permits them to get the information they want from different sources. Event generation, on the other hand, is by its very nature a distributed process and, without a central agent, all sources and consumers are forced to create a fully meshed topology. By introducing TRG, event collection becomes straightforward and trigger distribution standardized. That is why we propose that instead of using only the services provided by the IEEE 802.21 MIH functionality future mobile systems should use also TRG alongside 802.21 services. IEEE 802.21 can be, for example, the source entity that provides the lower layer information to TRG.

7. Concluding Remarks and Future Work

This paper presented a novel TRG framework for managing mobility-related triggers and its functionalities for collecting information from various event sources origi-

nating not only from the lower layers of the protocol stack (physical, data link, and network), but also from the upper layers and processing the collected events in a standardized trigger format. By using the defined mechanism, TRG framework enables easy and efficient use of cross-layer and cross-domain information. This framework was implemented and evaluated by performing tests in a real environment with several operating systems (Linux, FreeBSD, Windows, Linux Familiar for the PDA and Maemo Linux for the Nokia tablet) to prove its robustness and measure its performance.

The TRG framework experimentations with the performance test and evaluations showed that the implemented TRG functionalities are very promising. TRG can run efficiently in small device with very limited processing power and can enable lossless session handovers between devices. Stress tests showed that the TRG filtering mechanism does not cause delay for processing time and TRG can be used to filter and disseminate large numbers of triggers from several information sources.

Our Triggering management framework is currently integrated with Mobile IP [1] and HIP [2] protocols and is also a part of the Ambient Networks Architecture [15] and prototype as discussed in [16−18]. TRG and MIP integration with the use of network information a.k.a cascaded triggering presented in [24] showed the benefits of using TRG for the Mobile IP in the case when networks will be congested. HIP integration with TRG and test evaluations presented in [25] showed as well that TRG processing have only a small factor (less than 9%) to the total; trigger collection, processing and dissemination process.

Next steps will be to run a complete test with these integrated mobility protocols in real heterogeneous environments with the WLAN, 3G/HSDPA, WiMAX access technologies. Tests will benefit of the cascaded functionality of TRG when TRG can be located both at the terminal and network side as discussed in [24]. For example, TRG sources at the network side can monitor the network capacity load and other QoS metrics in overlapping networks and based on this information, the network side TRG can send triggers to the terminal initiating or even forcing a vertical handover.

While the tests and evaluations are made in a real testbed environment, a simulation environment will be built to fulfil the tests and analysis of the performance and scalability. In a forthcoming study we will also map the trigger management framework with the recently finalized standard by the IEEE 802.21 working group [12].

8. References

[1] D. Johnson, C. Perkins, and J. Arkko, "Mobility support in IPv6," Series Request for Comments, No. 3775. IETF, June 2004.

[2] A. Gurtov, "Host Identity Protocol (HIP): Towards the secure mobile Internet," Wiley and Sons, pp. 328, June 2008.

[3] J. Mäkelä, K. Pentikousis, M. Majanen, and J. Huusko, "Trigger management and mobile node cooperation," in M. Katz and F. H. P Fitzek (Editors), Cognitive wireless networks: Concepts, methodologies and visions inspiring the age of enlightenment of wireless communications, Springer-Verlag, pp. 199−211, 2007.

[4] J. Mäkelä and K. Pentikousis, "Trigger management mechanisms," Proceedings of the Second International Symposium on Wireless Pervasive Computing, San Juan, Puerto Rico, pp. 378−383, February 2007.

[5] P. Prasad, W. Mohr, and W. Konhuser, "Third generation mobile communication systems," Boston, MA, Artech House Publishers, 2005.

[6] J. Eisl (Editor), "Ambient networks D4.2: Mobility architecture & framework," EU-project IST-2002-507134-AN/WP4/D4.2, 2005.

[7] P. Vidales, J. Baliosian, J. Serrat, G. Mapp, F. Stajano, and A. Hopper, "Autonomic system for mobility support in 4G networks," IEEE JSAC, Vol. 423, No. 12, pp. 2288−2304.

[8] S. Ishihara, K. Koyama, G. Miyamoto, and M. Kuroda, "Predictive rate control for realtime video streaming with network triggered handover," IEEE WCNC, Vol. 3 No. 13−17, Las Vegas, Nevada, USA, pp. 1335−1340, 2005.

[9] H. Chaouchi and P. Antunes, "Pre-handover signalling for QoS aware mobility management," International Journal of Network Management, Vol. 14, No. 6, pp. 367−374, 2005.

[10] E. Casalicchio, V. Cardellini, and S. Tucci, "A layer-2 trigger to improve QoS in content and session-oriented mobile services," Proceedings of 8th ACM international Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Montral, Quebec, Canada, pp. 95–102, 2005.

[11] C. Kappler, P. Mendes, C. Prehofer, P. Pöyhönen, and D. Zhou, "A framework for self-organized network composition," Lecture Notes in Computer Science, No. 3457, Springer, pp. 139−151, 2005.

[12] IEEE Std 802.21™-2008, IEEE standard for local and metropolitan area networks-Part 21: Media independent handover services, IEEE, January 2009.

[13] R. Giaffreda, K. Pentikousis, E. Hepworth, R. Agüero, and A. Galis, "An information service infrastructure for Ambient Networks", Proc. 25th International Conference on Parallel and Distributed Computing and Networks (PDCN), Innsbruck, Austria, February 2007, pp. 21−27.

[14] OASIS eXtensible Access Control Markup Language, OASIS specification, Available: http://www.oasis-open.org/committees/xacml/.

[15] N. Niebert, A. Schieder, J. Zander, and R. Hancock (Eds.), "Ambient networks; co-operative mobile networking for the wireless world," Wiley & Sons, 2007.

[16] P. Pääkkönen, P. Salmela, R. Aguero, and J. Choque, "An integrated ambient networks prototype," Proceedings

SoftCOM 2007, Split, Croatia, pp. 27−29, September 2007.

[17] C. Simon, R. Rembarz, P. Pääkkönen, et al., "Ambient networks integrated prototype design and implementation," Proceedings 16th IST Mobile Summit, Budapest, Hungary, pp. 1−5, July 2005.

[18] K. Pentikousis, R. Agüero, J. Gebert, J. A. Galache, O. Blume, and P Pääkkönen, "The ambient networks heterogeneous access selection architecture," Proceedings First Ambient Networks Workshop on Mobility, Multiaccess, and Network Management (M2NM), Sydney, Australia, pp. 49−54, October 2007.

[19] J. Makela, R. Aguero, J. Tenhunen, V. Kyllönen, J. Choque, and L. Munoz, "Paving the way for future mobility mechanisms: A testbed for mobility triggering & moving network support," Proceedings 2nd International IEEE/Create-Net Tridentcom, Barcelona, Spain, March 2006.

[20] E. Tuulari and A. Ylisaukko-oja, "SoapBox: A platform for ubiquitous computing research and applications," Lecture Notes in Computer Science 2414, Pervasive Computing, Zurich, CH: Springer, pp. 125−138, August 2002.

[21] C. Pinho, J. Ruela, K. Pentikousis, and C. Kappler, "A protocol for event distribution in next-generation dynamic networks," Proceedings Fourth EURO-NGI Conference on Next Generation Internet Networks (NGI), Krakow, Poland, pp. 123−130, April 2008.

[22] C. H. Lwi, H. Mohanty, and R. K. Ghosh, "Causal ordering in event notification service systems for mobile users," Proceedings of International Conference on Information Technology: Coding and Computing (ITCC), Vol. 2, pp. 735−740, 2004. Conference Distributed Computing Systems Workshops (ICDCS 02), pp. 639−644, 2002.

[23] H. A. Duran-Limon, G. S. Blair, A. Friday, T. Sivaharan, and G. Samartzidis, "A resource and QoS management framework for a real-time event system in mobile ad hoc environments," Proceedings of International Workshop Object-Oriented Real-Time Dependable Systems (WORDS), pp. 217−224, 2003.

[24] M. Luoto and T. Sutinen, "Cross-layer enhanced mobility management in heterogeneous networks," Proceedings of International Conference on Communications, Beijing, China, May 2008.

[25] P. Pääkkönen, P. Salmela, R. Aguero, and J. Choque, "Performance analysis of HIP-based mobility and triggering," Proceedings of IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, 2008.

Int. J. Communications, Network and System Sciences, 2009, 3, 169-247
doi:10.4236/ijcns.2009.23024 Published Online June 2009 (http://www.SciRP.org/journal/ijcns/).

On Approaches to Congestion Control over Wireless Networks

David Q. LIU, Williana Jean BAPTISTE
Department of Computer Science,
Indiana University - Purdue University Fort Wayne, Fort Wayne, IN, USA
Email: {*liud, jeanw*}*@ipfw.edu*
Received January 25, 2008; *revised March* 29, 2009; *accepted April* 6, 2009

ABSTRACT

Congestion control in wireless networks has been extensively investigated over the years and several schemes and techniques have been developed, all with the aim of improving performance in wireless network. With the rapid expansion and implementation of wireless technology it is essential that the congestion control problem be solved. This paper presents a survey of five congestion control schemes which are different in slow start threshold calculation, bandwidth estimation, and congestion window manipulation. A comprehensive comparison of these approaches is given in relation to assumptions, bandwidth estimation, congestion window size manipulation, performance evaluation, fairness and friendliness and improved throughput.

Keywords: Transmission Control Protocol (TCP), Wireless Networks, Bandwidth Estimation, Congestion Window, Slow Start Threshold

1. Introduction

Congestion control in a TCP/IP-based internet is complex and challenging [1] and over the years a lot of effort and resources have been dedicated to the research in this area. TCP provides only end-to-end flow control and relies on packet loss as an indicator of congestion [1−3]. On the other hand, IP is a connectionless stateless protocol and has no provision for any mechanism to detect or control congestion.

TCP limits a sender's transmission rate relative to the network congestion such that if there is little congestion on the path between sender and receiver then the transmission rate will increase, otherwise if there is congestion, the transmission rate will decrease. TCP employs a window-based scheme to control the transmission rate and the size of the window directly limits the transmission rate. With TCP, congestion is avoided by changing the window size which greatly impacts the transmission rate.

Generally with most TCP versions used in the Internet today, if there is little or no congestion, the window size increases by some factor to the predefined size called the slow start threshold, *ssthresh.* After attaining the *ssthresh* size, the window size increases linearly. If a packet is lost or congestion is detected, the window size *is* decreased significantly to allow the network to recover from congestion.

The widely used standard TCP congestion control approach worked well for wired networks since loss of a packet was in most instances due to the congestion in the network. But with the rapid explosion of wireless networks [1,2,4], there is a significant increase in the number of combined wired and wireless networks and congestion control mechanisms previously used for wired networks do not perform well in the wireless links and wireless networks. The main reason for this decrease in performance of the widely used TCP congestion control mechanisms is that for wireless networks packet loss is caused frequently by several factors other than congestion such as noisy channels or fading radio signals, interference, host mobility and disconnection due to limited coverage [1,5,6].

The current TCP mechanisms can not distinguish congestion due to wireless fading channels or bandwidth reduction and therefore make unnecessary reduction in

the congestion window size(*cwnd*) and cause severe performance degradation [1,2,6,7].

Significant efforts and resources have been utilized in researching and developing techniques that would enhance performance in the wireless portion of wired-wireless networks. Two studies [6,7] show that accurate estimation of the available bandwidth for *ssthresh calculation and setting* greatly improves performance. Another study [4] indicates that effectively manipulating the size of the window is essential in improving performance. A combination of both bandwidth calculation and window manipulation is proposed in [2,5] to improve performance.

This paper reviews five approaches to TCP congestion control and review their implementations based on four techniques of managing the send window namely slow start, dynamic window sizing, fast retransmit and fast recovery. It is structured as follows; Section 2 describes five approaches to TCP congestion control for wireless networks including characteristics, algorithms and assumptions. In Section 3, these techniques are compared and contrasted for similarities and differences according to the areas of bandwidth estimation, congestion window calculation, performance, fairness and related results. Concluding remarks are stated in Section 4.

2. Overview of Congestion Control Techniques

In this section, several congestion control techniques over wireless networks are described. The Table 1 lists the terms used in these techniques.

2.1. TCP Enhancement for Transmission in Variable Bandwidth Wireless Environment

Since network bandwidth changes constantly especially in wireless networks, TCP must frequently probe the extra bandwidth of a network to optimally use the available bandwidth by adequately setting the slow start threshold. A scheme is proposed in [4] that dynamically sets the slow start threshold and manipulates the window size in both the slow start phase and the congestion avoidance phase. The slow start threshold is calculated by combining the expected rate with the actual rate to obtain an appropriate rate.

2.1.1. Slow Start Threshold Estimation

The *ssthresh* estimation calculates an appropriate *ssthresh* by combining the expected rate with the actual rate and is defined as follows:

<expected rate>=$cwnd/rtt_{min}$;

<actual rate>=$cwnd/rtt$;

AR=<expected rate>×β+<actual rate>×(1-β);

2.1.2. Congestion Window Estimation

The congestion window is calculated based on the degree of variation of rtt. For three consecutive increases in rtt the congestion window is defined as follows:

if $rtt_{va} < ½$, $cwnd_{next}=cwnd_{cur}+1$else $cwnd_{next} = cwnd_{cur}$

2.1.2.1. Slow Start Phase

TCP enters this phase when a connection is initiated or on timeout.

when timeout {

cwnd = 1; *ssthresh* = AR * rtt_{min}/seg_size;

if *ssthresh* < 2, *ssthresh* = 2;

when an ACK is received {

If *cwnd* < *ssthresh*, *ssthresh* = AR * rtt_{min}/seg_size

else cwnd = ssthresh, enter congestion avoidance phase

2.1.2.2. Congestion Avoidance Phase

For three consecutive increases of *rtt*,

if $var_{rtt} < ½$, $cwnd_{next} = cwnd_{cur} + 1$

For three consecutive decreases in rtt

if (rtt_{va}<1/3), $cwnd_{next} = cwnd_{cur} +1$

else if (1/3<= rtt_{va} <= 2/3) $cwnd_{next} = cwnd_{cur} +3$

else if (rtt_{va} > 2/3) $cwnd_{next} = cwnd_{cur} + 5$

2.1.2.3. Fast Retransmission Phase

This phase is entered when three duplicate ACKs are received. The sender immediately sends the out of sequence packet without waiting for the timer to expire.

ssthresh = AR * rtt_{min} / seg_size.

cwnd = *ssthresh.*

2.1.2.4. Fast Recovery Phase

TCP enters this phase after the fast retransmission phase. In order to reduce transmission, TCP sets *cwnd* = *ssthresh* and enters the congestion avoidance phase.

2.1.2.5. Retransmission Timeout Phase

TCP sets *cwnd* = 1 and enters the slow start phase.

Table 1. Congestion control terms.

Term	Meaning
ACK	acknowledgement
AR	appropriate rate
B_m	measured bandwidth
B_s	smoothed bandwidth
BWE	bandwidth estimation
cwnd	congestion window size
rtt	round trip time
rtt_{acr}	archived rtt
rtt_{var}	variation of rtt
seg_size	segment size
ssthresh	slow start threshold

2.2. Modified TCP Congestion Control Algorithm (Constant TCP)

Modified sender's TCP congestion control [2] uses a constant congestion window. The foundation of the proposal is based on the following assumption: if a TCP sender transmits packets at a rate greater than its fair share then some packets from the previous round would be in the network when the next round of packets are transmitted. If the load of the network is expressed in terms of queue length over some fixed time interval then L load at instant *i* is

$L_i = N + L_{i-1}$,

where N is the average amount of the new arriving traffic and L_{i-1} is the amount of traffic left after the last time interval. If the sender is transmitting packet with its fair share then $L_{i-1} = 0$ *and* $L_i = N$.

In this scheme, a TCP connection is divided into a number of slots such that the fair share of a connection of the network bandwidth remains unchanged for a slot period during the lifetime of the connection. Changes in the available bandwidth are due to connections leaving and joining the network at that time the current slot ends and a new slot starts. The bandwidth calculation algorithm similar to [4] is used to estimate the available fair share for a slot. A new slot triggers the recalculation of the congestion window, which is set according to the available connection bandwidth. In this proposal the modified TCP sender goes through three phases during the lifetime of the connection. They are the start-up, the window recalculation phase and the constant window phase.

2.2.1. Startup Phase

When a connection is initiated, the sender uses the slow start phase for *k rtt* rounds gathering data such as the minimum *rtt* and the network bandwidth in order to calculate the starting *cwnd*.

2.2.2. Window Recalculation Phase

TCP enters this phase when there is a change in the available fair share triggered by a change in *rtt* values. The rtt values are archived as rtt_{acr} for future use and

$cwnd = \text{BWE} * rtt_{\min} / seg\_size$

2.2.3. Constant Window Phase

In this phase the *cwnd* calculated from the start up phase is kept constant regardless of the number of ACKs or DUPACKs received or timeouts, but changes in the *rtt* will trigger a window recalculation by tracking the *rtt* estimates from received segments. If $|rtt_{acr} - rtt_{var}| / rtt_{arc} > \beta$ then a window recalculation phase is entered.

2.3. Two Phase Congestion Control (TCP-TP)

The sender-side control congestion scheme TP-TCP [7] measures the network capacity and uses it to set the congestion window size and the transmission rate in order to optimally utilize the available bandwidth. TCP-TP includes the fair convergence phase and the congestion avoidance phase. In addition, the receiver delays ACKs to reduce the number of packets in the network.

2.3.1. Bandwidth Estimation

The TCP-TP sender measures the bandwidth during the lifetime of the connection and uses this information to calculate the congestion window to optimally utilize the available bandwidth. The bandwidth is measured using the following equation:

B_m=<bytes received between two successive ACKs>/ <time interval between two successive ACKs> This measured bandwidth B_m reflects the network environment at that point in time.

Due to the constant changes of bandwidth in the network, the scheme uses a smoothed bandwidth B_s instead of the measured bandwidth B_m. The smoothed bandwidth is the sum of the previously smoothed bandwidth and the current measured bandwidth according to the following equation:

$$B_{s(i)} = \alpha B_{s(i-1)} + (1-\alpha) B_m$$

2.3.2. Window Calculation

Initially,

ssthresh is calculated as follows:

ssthresh = <*network bandwidth*> * *rtt*

$cwnd = B_s * rtt_{min}$

2.3.2.1. Congestion Control Avoidance Phase

When *cwnd* = *ssthresh*, TCP-TP enters this phase and works like the standard TCP except that the cwnd increases and decreases by N at once for every N packets rtt.

2.3.2.2. Fair Convergence Phase

In this phase TCP-TP measures the network bandwidth and calculates and sets the *cwnd* and *ssthresh.* After each ACK is received, *cwnd* increases by β bytes where

$$\beta = \gamma\, rtt^2\, (ssthresh\text{-}cwnd)$$

2.4. TCP-Westwood Bandwidth Estimation

The sender of the modified TCP-Westwood (TCPW) [5] continuously measures the round trip time of the returning ACKs to calculate a minimum slow start threshold and congestion window which effectively utilizes the bandwidth at the time of congestion. This scheme TCPW uses a two-phase approach to control congestion namely, bandwidth estimation phase and the congestion control phase.

TCPW assumes that on receipt of three DUPACKs, the network capacity has been reached or packets have been dropped due to sporadic loss in wireless networks.

2.4.1. Bandwidth Estimation Phase

In this phase the sender continuously probes the network connection and calculates the available bandwidth and tracks the rtt from the returning ACKs. After a congestion episode the calculated bandwidth is used to determine the *ssthresh* and *cwnd.*

Before congestion episode the TCP sender increases the *cwnd* to determine network capacity and the bandwidth estimation is calculated as follows

$$Bwe= d_k /t_k - t_{k-1}$$

where

d_k: data transferred at time k

t_k: the time ACK was received at source for transmission of data k

t_{k-1}: the time ACK was received at source for transmission of previous data k-1.

Averaging the sample measurements accounts for the low frequency components of the available bandwidth and a low pass filter is used on the estimated bandwidth.

2.4.2. Calculation of Congestion Window

The *cwnd* is calculated using the *ssthresh* after 3 DUPACKs are received or timeout expiration.

2.4.3. Slow Start Phase

At the beginning of a connection:

cwnd =1

ssthresh = BWE x rtt_{min}/*seg_size*

cwnd increases by 1 for each new ACK receipt

until *cwnd* = *ssthresh*

If (timeout expires)

ssthresh = BWE x rtt_{min}/*seg_size*;

if (*ssthresh* < 2*) ssthresh*=2*;*

cwnd =1

2.4.4. Congestion Avoidance Phase

During this phase the sender probes for extra bandwidth and exponentially increases the *cwnd*. Once three DUPACKs are received, the network is at its capacity and this scheme uses the following algorithm for setting the *cwnd* and *ssthresh*:

If (n DUPACKs are received)

ssthresh = BWE x rtt_{min}/*seg_size*;
if (*cwnd*>*ssthresh*) *cwnd*=*ssthresh*

2.5. Enhanced Bandwidth Estimation (TIBET)

A bandwidth estimation scheme, Time Intervals based Bandwidth Estimation Technique (TIBET) [6], modifies the sender side of the TCP congestion control procedure. TIBET is based on the principle that if more information is available, the better is the estimation of the available bandwidth to a connection, leading to better and fair utilization of network resources.

If *n* packets (L_1,L_2,L_3…L_n) are transmitted within a time interval of *T*, then the average bandwidth BWE is given by

$$BWE=1/T *\sum L_i \qquad \text{where i =1 to n}$$

which can be rewritten as

$BWE=L_{mean}/(T/n)$, where L_{mean} is the average packet length in bits and T/n is the average interarrival time. Thus average used bandwidth over a time period is equal to the average packet length in bits transmitted during that time period/average inter arrival time. This scheme also proposes the low-pass filtering of either the packets lengths and their inter departure times.

2.5.1. Bandwidth Estimation Phase

Below is the pseudo code for estimation of bandwidth based on transmitted packets.

```
if(packet is sent)
   sample_length[k] = (packet-size * 8);
   sample_interval[k] = now –last_sending_time;
   avg_packet_length[k] = α * avg_packet _length [k-1]
                          +(1-α)* sample_length[k];
   avg_interval[k]=α * avg_interval[k-1]
                   + (1-α)* sample_interval[k];
  BWE[k] =avg_packet_length[k] / avg_interval[k};
```

where packet size is the *segment size* in bytes, *now* is the current time, *last_sending_time* is the time of the previous packet transmitted, α is the low-pass filter, and *BWE* is the estimated value of the used bandwidth.

A second alternative for calculating the average bandwidth is also proposed for received ACKs. The algorithm used for the ACKs is similar to the stated above except for the calculation of:

```
sample_length[k] = (acked * packet_size * 8)
sample_interval[k] = now – last_acked_time;
```

where last_acked_time is the time the last ACK was received, and acked is the number of segments acknowledged by the last ACK.

2.5.2. Calculation of Congestion Window

The *cwnd* is set to 1 after 3 DUPACKs are received or timer expires, and then the slow start phase is entered. The cwnd grows exponentially as usual until cwnd = ssthresh and at that time, the congestion avoidance phases is entered.

2.5.3. Slow Start Phase

At this phase, the *cwnd* and the *ssthresh* are set as follows:

ssthresh=BWE * rtt_{min}
cwnd=1

2.5.4. Congestion Avoidance Phase

During this phase the sender probes for extra bandwidth and exponentially increases *cwnd* to *ssthresh*. Once the

ssthresh has been reached, the *cwnd* increases by one for each ACK received. If three DUPACKs are received, the network has reached its capacity.

If ($cwnd = ssthresh$) $rtt_{min}=(1-\beta)\ x)\ rtt_{min}$

At this time, slow start phase is entered.

3. Comparison of Various TCP Congestion Control Techniques

The congestion control schemes presented in Section 2 are compared with respect to assumptions, bandwidth estimation, window size manipulation, slow start phase, retransmission phases, congestion avoidance phase and results.

3.1. Comparison of Assumptions

All schemes comply with true end-to-end TCP design principle and do not require the interception of packets by intermediate nodes. Further, for all schemes included in this survey the modifications were made only to the sender side of the traditional TCP congestion control algorithm. Each scheme is based on specific assumptions. For example, the constant TCP assumes that most indications of congestion by the current TCP variants used in the Internet does not necessitate a reduction in the transmission rate of the connection, as such, the *cwnd* size should remain constant until some other factors indicate that true congestion has occurred.

3.2. Comparison of Bandwidth Estimations

Most approaches [2,5−7] described in Section 2 state that the bandwidth estimation algorithm in Reno is inaccurate and causes the under utilization of available bandwidths by TCP entities and propose alternate bandwidth measurements that would optimally utilize the available bandwidth and improve transmission rate. The bandwidth is estimated using the average rate of returning ACKs [2,5]. This estimation more accurately reflects the TCP entity's fair share. The estimated bandwidth [5] is then used to set the *cwnd* and *ssthresh* after congestion episode or timeout expiration.

Another improved bandwidth estimation [7] is the rate of bytes received during immediate successive inter-arrival ACKs. Paper [6] proposes using a low-pass filter rate of average packet length in bytes for inter-arrival times for transmitted packets. This algorithm estimates the used bandwidth by measuring the inter-arrival samples and not the bandwidth samples compared with the algorithm used by [2,5] which directly samples the bandwidth. All bandwidth estimations were smoothed by using a low pass filter to account for the rapidly fluctuating network environment.

None of the papers reviewed compared their bandwidth estimation algorithms with regards to enhancing performance. All assume that their modified bandwidth estimation algorithm would more accurately estimate the available bandwidth resulting in optimal use of the connection's fair share.

3.3. Comparison of Congestion Window Size Manipulation Techniques

Since rate of transmission is indirectly [1,2] related to the congestion window size, effectively manipulating the window size will improve transmission rates because in wireless networks window size is unnecessarily reduced due to loss prone nature of the wireless links and not as a result of congestion. Several approaches manipulate the congestion window size and set the slow start threshold in order to maintain a high transmission rate comparable to the available bandwidth. One of them [2] maintains a constant congestion window and does not react by decreasing the window size when DUACKs are received and timeout expires. Instead, responding only when the network environment becomes sufficiently degraded through monitoring rtt values. Another scheme [4] proposes increasing (decreasing) the congestion window only when the change of three *rtt* values is greater (less) than some predefined factor.

Changing the window size by some fix factor is stated in [7] and that factor is calculated using the current available bandwidth. Other techniques reset the congestion window to the either the previously calculated slow start threshold [5] or to the newly calculated slow start threshold [4] which takes the current network environment into consideration once the network capacity is reached.

All these techniques aim to limit the unnecessary reduction in window size in wireless links thereby improving overall throughput.

3.4.1. Slow Start Phase

In this phase, available bandwidth is probed and the congestion window is increased by some factor. Various approaches are proposed for this phase. Three conditions would cause TCP entities to enter this phase and include starting up a connection, receipt of 3 DUPACKS and timer expiration. For all the proposed approaches at the start of a connection, the *cwnd* is set to either one [4−7] or a fixed value obtained after probing the bandwidth for a fixed number of round trip times [4].

As each new ACK is received, the *cwnd* is increased by one and information such as bandwidth and *rtt* measurements are collected in order to calculate *ssthresh* until the *cwnd* reaches the *sshtresh* value. One approach [2] has no need for a slow start phase since the congestion window once calculated is kept constant irrespective of detection of congestion.

3.4.2. Retransmission Timeout Phase

TCP enters this phase when the timer set on a packet transmitted expires before an ACK is received for that

packet. Several techniques are used to decrease the transmission rate in order to alleviate congestion in the network.

One of these techniques is setting cwnd to one [4,5,7] whilst the ssthresh is set to the value of the bandwidth estimated at that time multiplied by the rttmin [5]. The ssthresh is set to <actual rate> * rttmin/seg_size, which would allow for faster recovery.

Another scheme [2] does not enter this phase since most times the timeout is not an indication of congestion in wireless and there is no need to drastically reduce the transmission rate to reduce congestion.

3.4.3. Congestion Avoidance Phase

Once the congestion window equals the *ssthresh* this phase is entered and *cwnd* is increased or decreased by various functions. Various techniques are employed in this phase. One such technique is monitoring the *rtt* measurements and recalculating *cwnd* when some preset condition is met [2,4].

In [4], if there are three consecutive *rtt* value increases or decreases, then the *cwnd* is decreased or increased by a factor. However, in [2], *cwnd* is kept constant until the measured *rtt* function is greater than some fixed value, and at that time *cwnd* is recalculated and set to *BWE * rttmin/seg_size*. Another technique used in this phase is decreasing and increasing *cwnd* by a fix number of bytes [7].

3.5. Comparison of Performance Evaluation

Most congestion control schemes used NS2 simulations to evaluate their performance except for [7] where experiments were performed by modification of Linus Kernel 2.6.7. Commonly used performance metrics were employed in both the simulations and experiments and include error rate, bandwidth, link capacity, number of connections and rtt length, fairness and friendliness. Table 2 shows metrics used in each scheme.

The percentage increase in throughput compared to the standard TCP congestion control technique implemented in the Internet is presented in Table 3. Various levels in improvement in throughput were observed for all schemes, ranging from 10% to 550%.

Due to the lack of commonality amongst the compared metrics it was generally impossible to compare techniques against each other to determine the best algorithm. However, all techniques compared their improve performance against the TCPW and the results are presented in Figure 1−Figure 4. Overall, the TCP constant, TCP-TP and TCP-EVBWE all out perform TCPW in various network scenarios.

3.6. Comparison of Fairness and Friendliness

Fairness and friendliness are important metrics in evaluating the performance of a scheme. Fairness means that all similar connections have the same opportunity to transfer data and that one connection would not aggressively consume resources at the expense of other connections such that connections with longer round trip times are not at a disadvantage. Friendliness is that connections of different schemes are able to co-exist [8].

Table 2. Shows metrics used to evaluate performance.

Congestion	Error	Number of	Rtt	Band-
TCP-ETVBWE[4]	Yes	No	No	Yes
constant_TCP[2]	Yes	Yes	No	No
TCP_TP[7]	No	No	Yes	No
TCPW[5]	No	Yes	Yes	Yes
TIBET[6]	Yes	Yes	Yes	No

Table 3. Throughput increase for each proposed congestion control scheme.

Congestion Control Schemes	Throughput Increase%
TCP-ETVBWE[4]	10
Constant_TCP[2]	10-20
TCP_TP[7]	~10
TCPW[5]	394-550
TIBET[6]	50

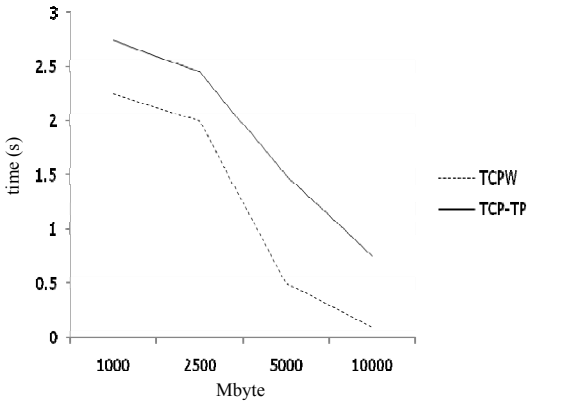


Figure 1. Throughput (Mbps) comparison in wireless networks with varying packet round trip times.

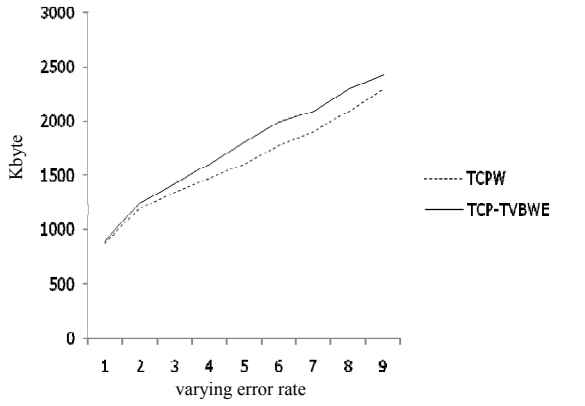


Figure 2. Throughput (Kbps) variation with varying error rates.

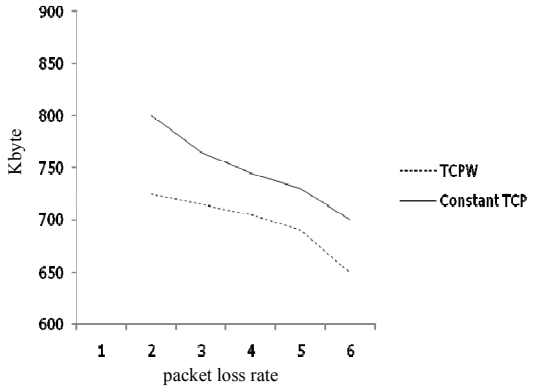


Figure 3. Throughput (Kbps) variation with packet loss rates in wireless links(%).

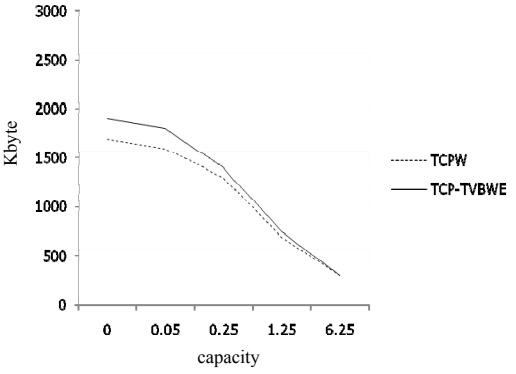


Figure 4. Throughput (Kbps) variation with capacity (Mbps) wireless links.

Table 4. Fairness and friendliness rank (0-3).

Congestion Control Schemes	Fairness	Friendliness
TCP-ETVBEW[4]	0	0
constant TCP[2]	0	0
TCP_TP[7]	3	3
TCPW[5]	2	2
TIBET[6]	3	3

0-not evaluated.

Some of these schemes were evaluated for fairness and friendliness and the results are ranked based on the extent of fairness and friendliness reported. TCP-TP and TIBET are both fair and friendly schemes while TCP constant and TCP-ETVBEW were not evaluated for fairness and friendliness [1,2]. It is difficult to see how TCP constant would be fair or friendly to other TCP entities since the *cwnd* remains constant even when the network environment changes and the TCP entity would continue to transmit at a high rate thereby consuming resources of other entities. The ranking of the proposed schemes in terms of fairness and friendliness is presented in Table 4.

4. Conclusions

Five sender side modification schemes to the standard TCP congestion control algorithm are surveyed in this paper. Their characteristics, algorithms and assumptions were presented. A comparison of their assumption, bandwidth estimation, window size manipulation, slow start phase, retransmission phases, congestion avoidance phase and performance evaluation methods is conducted. The need for an efficient method to optimally utilize available bandwidth is essential in the wireless links of combined wired and wireless networks. Some schemes propose efficient estimation techniques of available bandwidth in a dynamic internet environment while others schemes effectively manipulate the congestion window and set the slow start threshold. With simulations and experiment each of these schemes shows an improvement in throughput between 10−50%.

One of our future research projects is to evaluate these schemes against each other by comparing them in various network scenarios such varying bit error rates, number of connections, link capacity, and bandwidth.

5. References

[1] S. Schmid and R. Wattenhofer, "A TCP with guaranteed performance in networks with dynamic congestion and random wireless losses", In Proceedings of the 2nd Annual International Wireless Internet Conference (WICON'06), August 2006.

[2] R. Roy, S. Das, A. Ghosh, and A. Mukherjee, "Modified TCP congestion control algorithm for throughput enhancement in wired-cum-wireless networks" In Proceedings of the 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2007.

[3] W. Stallings, "High-speed networks and internets performance and quality of service," Prentice Hall Inc., 2002.

[4] N. Wang, C. Chiou, and Y. Huang, "TCP enhancement for transmission in a variable bandwidth wireless environment," Proceedings of IWCMC, pp. 37−42, August 2007.

[5] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, "TCP Westwood: Bandwidth estimation for enhanced transport over wireless links," Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (ACM SIGMOBILE), pp. 287−297, 2001.

[6] A. Capone, L. Fratta, and F. Martignon, "Enhanced bandwisth estimation algorithms in TCP congestion control scheme," in Proceedings of the IFIP Conference on Network Control and Engineering of QoS, Security and Mobility, pp. 469−480, 2002.

[7] J. Lee, H. Cha, and R. Ha, "A two-phase TCP congestion control for reducing bias over heterogeneous networks," in Proceedings of International Conference on Information Networking, Convergence in Broadband and Mobile Networking, (ICOIN), pp. 9−108, 2005.

[8] A. Ghosh, S. Das, R. Roy, and A. Mukherjee, "Constant congestion window approach for TCP-effect on fairness," in Proceedings of 3rd Swedish National Computer Networking Workshop, September 2005.

Int. J. Communications, Network and System Sciences, 2009, 3, 169-247
doi:10.4236/ijcns.2009.23025 Published Online June 2009 (http://www.SciRP.org/journal/ijcns/).

A Novel Approach to Improve the Security of P2P File-Sharing Systems

Cuihua ZUO, Ruixuan LI[+], Zhengding LU
College of Computer Science and Technology,
Huazhong University of Science and Technology, Wuhan, China
E-mail: *zuocuihua@smail.hust.edu.cn, {rxli,zdlu}@hust.edu.cn*
Received February 15, 2009; *revised April* 6, 2009; *accepted April* 16, 2009

ABSTRACT

The recent and unprecedented surge of public interest in peer-to-peer (P2P) file-sharing systems has led to a variety of interesting research questions. How to minimize threats in such an open community is an important research topic. Trust models have been widely used in estimating the trustworthiness of peers in P2P file-sharing systems where peers can transact with each other without prior experience. However, current P2P trust models almost take no consideration for the nature of trust, fuzzy, complex and dynamic, which results in low efficiency in resisting the attacks of malicious nodes. In this paper, a new trust model named NatureTrust that can alleviate the shortage brought by the nature of trust is proposed. In order to cope with the fuzzy characteristic of trust, linguistic terms are used to express trust. Additionally, fuzzy inference rules are employed to evaluate trust of each transaction so as to handle the complex characteristic of trust. Furthermore, risk factor is deployed into NatureTrust to represent and reason with the dynamic characteristic of trust. Both risk and trust factors are considered in evaluating the trustworthiness of each peer. Experimental results show that the trust model analyzed here thus stands against malicious act effectively.

Keywords: Peer-to-Peer (P2P), File-Sharing, Security, Risk, Trust Model, Fuzzy Inference

1. Introduction

In peer-to-peer (P2P) file-sharing systems, all peers are both users and providers of resources and can access each other directly without intermediary agents. Typically, peers are autonomous, anonymous and self-interested, which means individuals seek to maximize their own goal achievement rather than act in a benevolent manner. Consequently, security becomes an open problem in these large and distributed systems since peers can break their commitments or provide sub-standard or even malicious services. Though trust models, like EigenTrust [1] and PeerTrust [2,3], can be used to help P2P systems deal with the security problem, they do not consider the nature of trust: fuzzy, complex and dynamic characteristics [4]. Hence, they can not achieve a preferable effect.

The three types of natural characteristics of trust are as follows. 1) Fuzzy characteristic: the fuzzy nature of trust means it is imprecise and sometimes ambiguous when we express trust or try to explain a trust level. 2) Complex characteristic: the complex nature of trust arises from the fact that there are multiplicity of ways in determining the trust and a variety of views about trust. 3) Dynamic characteristic: the dynamic nature of trust refers to trust not being constant or stable but always changing as time passes. However, Current research seldom considers these three characteristics of trust in peer-to-peer file-sharing systems.

The ultimate goal of our research is to solve the security problem effectively in distributed P2P file-sharing systems, which is incurred by the nature of trust: fuzzy, complex and dynamic. Towards the end, NatureTrust, a new trust model is proposed, which introduces linguistic terms instead of numerical values to express trust and imports fuzzy inference rules to infer trust value of each transaction. The risk factor is also taken into account when evaluating the trustworthiness of each peer.

The rest of this paper is organized as follows. The introduction of related work about trust models is provided in Section 2. Section 3 presents a new trust model which

[+]Corresponding author. E-mail: rxli@hust.edu.cn.

considers risk factor and trust factor separately in order to alleviate the security issues aroused by malicious peers in P2P networks. This section explains how to express trust, how to apply fuzzy inference rules into trust evaluation, and how to compute risk value. In addition, this section also describes the implementation strategies of this new trust model. Then Section 4 evaluates the performance of the proposed trust model with simulation experiments, followed by the conclusion and future work in Section 5.

2. Related Work

In Peer-to-Peer networks, peers cooperate to perform a critical function in a decentralized manner. Among the heterogeneous peers, some might be honest and provide high-quality service, some might be buggy and unable to provide high-quality service, some might be even malicious by providing bad services or harming the consumers. In the current P2P file-sharing systems, there are mainly three types of malicious peers: simple malicious peer, traitor and hypocritical peer.

In order to cope with such malicious behavior, some reputation-based P2P trust models have been proposed. As is well known, centralized reputation systems has been widely applied in e-commerce [5,6], such as eBay [7]. Some researches [8,9,10] suggested reputation based systems as an effective way for protect the P2P network from possible abuses by malicious peers. Reputation systems can help peers establish trust among them based on their past behaviors and feedbacks. Let us see several prominent decentralized reputation systems in the P2P domain. [11,12] proposes a reputation-based approach for P2P file sharing systems (called P2PRep). P2PRep runs in a fully anonymous P2P environment, where peers are identified using self-assigned opaque identifiers. [13] presents a similar approach, called XRep, which extends P2Prep by considering the reputations of both peers and resources. P2PRep and XRep do not consider the credibility of voters. Hence, malicious peers can give bad votes to an honest peer or give good votes to a dishonest peer, which results in a significant decline in the performance of restraining malicious behavior.

EigenTrust is also a reputation-based approach for P2P file sharing systems. In EigenTrust, each peer is assigned a unique global reputation value. However, it is not clear if their approach is feasible for large-scale P2P systems, in which some local reputation values are unreachable for the requesting peers. [14] suggests an approach to trust management for semantic web which is similar to EigenTrust, but ratings are personalized for each user based on his personal experience. Both approaches simply assume that peers are honest and therefore cannot defend some attacks like deceptions and rumors. PeerTrust develops a P2P trust model, so that peers can quantify and compare the trustworthiness of other peers and perform trusted interactions based on their past interaction histories without trusted third parties.

PET [15] proposes a personalized trust model to help the construction of a good cooperation, especially in the context of economic-based solutions for the P2P resource sharing. It designs a risk evaluation to handle the dramatic spoiling of peers. However, only denoting the opinion of the short-term behavior, the risk evaluation does not react on the dynamic nature of trust totally. Unlike the above, the risk evaluation in our trust model represents the fluctuating of peers' trust in the past behavior. In [16,17], ECMBTM is proposed which use cloud-model [18] to model trustworthiness and uncertainty of trust relationships between peers. But the trust aggregation is so complex that it's hard to apply it to practice.

Our work is inspired by these previous works for reputation-based P2P trust models and benefits from the nature of trust. But there are some differences between our effort and the above reputation systems. Firstly, in this paper, we focus on both risk and trust two aspects in evaluating the trustworthiness of peers. In addition, we use linguistic terms to express trust and employ fuzzy inference rules to evaluate trust of each transaction. More importantly, neither of the above reputation systems addresses the strategic behavior by malicious peers.

3. Trust Model

Trust is an accumulative value for the past behavior and reflects the overall evaluation on the valued peer. However, it is not sensitive enough to perceive the suddenly spoiling peer because it needs time to decrease the accumulative score. Meanwhile, it is also hard to perceive traitors who may behave properly for a period of time in order to build up a strongly positive trust, and then begin defecting. What's worse, it is harder to perceive the malicious peers with strategically altering their behavior. Therefore, trust is not enough in evaluating the actions of peers due to its dynamic characteristic.

When a peer involves in a transaction, it is entering into an uncertain interaction, which has an associated risk of failure or reduced performance. For security, the trust model need take risk factor into account. Hence, the main focus of this paper is the design of NatureTrust that is a unique characteristic with the combination of trust and risk factors for evaluating the trustworthiness of peers in P2P file-sharing systems. Here, we use a two-tuples with trust and risk values Tr: (T, R) to express the trustworthiness of peers. Additionally, each peer stores the values of trust and risk of its acquaintances using a XML document. Peers can change their XML documents to achieve some recommendation information.

3.1. Evaluation of an Interaction

Trust is fuzzy and complex when we express it. In P2P file-sharing systems, it is hard to give an accurately nu-

merical value after an interaction, and peers can have different views or policies in evaluating trust, which throws the trust evaluation system into disorder. For instance, one generous peer gives a trust value 0.9 to a certain service, while another one just gives 0.5 to it. Meanwhile, the situation occurs a lot if peers give precise trust values just in accordance with their own standards. In this situation, some malicious peers are easy to give unreasonable evaluations purposely, which may exaggerate the credibility of their conspirators or slander that of the benevolent peers. Besides, the trust evaluation can derive from different measurement criteria, such as "quality", "speed" and so on. This means a trusting agent is hard to explicitly articulate and specify a trust value that he or she has in another trusted agent after a transaction. Therefore, how to evaluate trust value for each transaction becomes an important problem.

In this section, we deal with different measurement criteria of evaluating trust by introducing linguistic terms and fuzzy inference rules. Concretely, we first define the set of measurement criteria for evaluating trust and give different grades for each measurement criterion according to user's satisfaction degree. Furthermore, we classify trust into different grades and establish a series of inference rules from the grades of measurement criteria to the grade of trust, and then use these rules to infer trust grade of each transaction. Finally, we define a map function *h* that maps from each trust grade to a corresponding trust value for each transaction.

Definition 1: Supposing that the set of linguistic terms about the trust grades is $X=\{x_1,x_2,\ldots,x_N\}$, and the set of trust value is $Y=\{y_1,y_2,\ldots,y_N\}$, $Y\subset [0,1]$, where N is the number of the trust grades. The function h(x) maps each element $x_i\in X$ into a value $y_i\in Y$, $h: X\to Y$. So y_i is the corresponding trust value of the trust grade x_i.

For example, we define N=6, X={distrust, a little trust, ordinary trust, a lot of trust, extraordinary trust, absolute trust}, Y={0,0.2,0.4,0.6,0.8,1}. The map function h can be defined as below:

$$h(x)=\begin{cases}1 & (x=\text{"absolute trust"})\\ 0.8 & (x=\text{"extraordinary trust"})\\ 0.6 & (x=\text{"a lot of trust"})\\ 0.4 & (x=\text{"ordinary trust"})\\ 0.2 & (x=\text{"a little trust"})\\ 0 & (x=\text{"distrust"})\end{cases} \quad \textbf{(1)}$$

In this paper, we assume C={C1,C2,…,Cm} as the set of m different measurement criteria, and for each measurement criterion Ci, there is a corresponding set to describe its grade, such as Ci: $\{c_1,c_2,\ldots,c_k\}$. For example, aiming at the service of file download, users can evaluate trust according to two criteria-file quality and download speed, so the set C can be defined as {file quality, download speed}. The set Q={bad quality, normal quality, good quality} can be regarded as the grade of the criterion "file quality" and the set S={slow speed, normal speed, fast speed} that of the criterion "download speed".

Since fuzzy inference is good at handling imprecise inputs, such as assessments of quality or speed, and allows inference rules to be specified by imprecise linguistic terms, such as "good quality" or "slow speed", we use fuzzy inference rules to combine the appraising information from different aspects of trust. The basic form of fuzzy inference rules is as follows:

If C1 is c_1 **and** C2 is c_2 … **and** Cm is c_m
then T is x.

Also using the above example, we might have rules such as the following.

If "file quality" is "good quality" **and** "download speed" is "fast speed"
then trust appraisement is "absolute trust".
If "file quality" is "good quality" **and** "download speed" is "normal speed"
then trust appraisement is "extraordinary trust".

Thus, after a transaction between peer i and peer j, peer i will give the appraisement like this: "good quality" and "normal speed". Through the above rules, we can infer that trust appraisement is "extraordinary trust". Similarly, according to the map function h(x) in the above, the trust value of peer j in view of peer i based on this direct interaction with peer j is 0.8.

3.2. Trust Computation

In this section, we present a general trust metric that combines the direct and indirect factors into a coherent scheme to compute the overall trust value.

Definition 2: we define $t_{ij}^{(n)}$ as the trustworthy of peer j in view of peer i in the n-th direct transaction. The value of $t_{ij}^{(n)}$ can be gained according to the inference method described in Subsection 3.1.

Definition 3: We define t_{ij} as the reliability of peer j in view of peer i based on its direct interactions with peer j.

$$t_{ij}=\frac{\sum_{n=1}^{M} t_{ij}^{(n)} * (1-\mu)^{M-n}}{\sum_{n=1}^{M}(1-\mu)^{M-n}} \quad \textbf{(2)}$$

where μ ($0<\mu<1$) is a time declining constant, and it determines the weights given to the most recent past observations. The bigger μ is, the faster the past observation is forgotten. M is the total number of direct interactions between i and j.

Definition 4: we define r_{ij} as the total recommendation from other peers who has even transacted with peer j.

$$r_{ij}=\lambda*\frac{\sum_{1}^{m} t_{il} * t_{lj}}{\sum_{1}^{m} t_{il}}+(1-\lambda)*\frac{\sum_{1}^{g} t_{zj}}{g} \quad \textbf{(3)}$$

The first part in the above formula is the recommendation from trustworthy references which have transactions with peer i, and the second part is the recommendation from unknown references. m and g are the number of trustworthy references and the number of unknown references respectively. l and z denote the peers of trustworthy references and unknown references respectively. λ is the weight to indicate how the peer i values the importance of the recommendation from trustworthy references and from unknown references. Certainly, comparing to unknown references, the peers who have even transacted with i is more trustworthy. So λ is bigger than 0.5 normally.

Definition 5: we define T_{ij} as the reliability of peer j in view of peer i based on its direct interactions and other peers' recommendation.

$$T_{ij} = w * t_{ij} + (1-w) * r_{ij} \quad \textbf{(4)}$$

From the definitions above, T_{ij} is decided by two factors. One is the reliability of peer j in view of peer i based on its direct interactions with peer j. The other is the total recommendation of peer j from other peers. As we known, peers always trust in themselves than others' recommendation, so w is bigger than 0.5.

3.3. Risk Computation

Peer's behavior can change dynamically, which implies that we need rely on not only the trust factor to evaluate the trustworthiness of peers, but also the risk factor. In NatureTrust, we use entropy of information theory to quantify the risk of each transaction between two peers. In information theory, entropy expresses the uncertainty degree of information. The smaller the entropy is, the lower the uncertainty degree is.

In this paper, the calculation of risk is based on the trust values from the direct interactions in the past which is reliable and self-determined, for risk is used to describe the fluctuation of peers' actions.

Definition 6: We define R_{ij} as the risk value of peer j in view of peer i. The formula of calculating risk value is as follows.

$$R_{ij} = \begin{cases} \dfrac{1}{\log N} * H_{ij} & (M \geq N) \\ R_0 & (M < N) \end{cases} \quad \textbf{(5)}$$

$$H_{ij} = -\sum_{k=1}^{N} p_{ij}^{k} * \log(p_{ij}^{k}) \quad \textbf{(6)}$$

In the above formulae, N is the total number of the classification of trust grades, and H_{ij} is the value of entropy relying on p_{ij}^{1}, p_{ij}^{2}, …, p_{ij}^{N}, which express the probability of N different trust grades appearing in M times direct interactions between peer i and j respectively. R_0 is the initialization value of risk. From the Equation (6), we can educe $0 \leq H_{ij} \leq \log N$, thus $0 \leq R_{ij} \leq 1$.

For example, we also suppose that the trust degree is classified into 6 grades, such as {distrust, a little trust, ordinary trust, a lot of trust, extraordinary trust, absolute trust}, and the corresponding set of trust values is {0, 0.2, 0.4, 0.6, 0.8, 1}. Assuming peer i and j have 10 times transactions in the past and the trust values are {0.6, 0.8, 0.6, 0.4, 0.6, 0.8, 0.8, 0.4, 1, 0.6}, then the probability p_{ij}^{1}, p_{ij}^{2}, …, p_{ij}^{6} are 0, 0, 0.2, 0.4, 0.3, 0.1, respectively. Hence, the values of entropy and risk can be computed according to the above formulae: H_{ij}=−(0.2*log0.2 + 0.4*log0.4 + 0.3*log0.3 + 0.1*log0.1)=0.556, $R_{ij} = H_{ij}$/ log6=0.715.

3.4. Managing Data

Figure 1 gives a sketch of evaluation mechanism for NatureTrust. There is no central database. The data that are needed to compute the trust value and risk value for peers are stored across the network in a distributed manner. Each peer has a data manager that is responsible for trust evaluation and risk evaluation.

The data manager of each peer performs two main functions. On the one hand, it submits recommendation information for other peers. On the other hand, it is responsible for evaluation the peer's trustworthiness. This task is performed in trust and risk two aspects. In risk aspect, the peer only relies on its own direct trust values to computer the risk value. These direct trust values derive from measurement criteria of trust, trust grades, inference rules and mapping function, which described in Subsection 3.1. In trust aspect, the peer needs to collect trust data from other peers in the network, and then combines direct trust to compute the total trust value. Hence, each peer need store the information of trust grades, measurement criteria of trust, inference rules, trust values of direct transactions and mapping function.

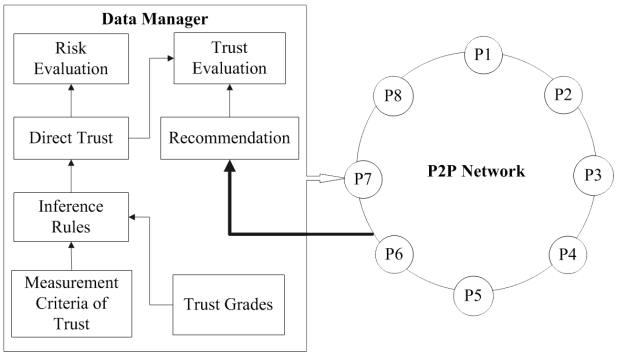


Figure 1. NatureTrust architecture.

3.5. Peer Selection Scheme

A key objective of peer selection scheme is to select one peer or a subset of peers that is or are most qualified to provide service in terms of the trustworthiness. The trust and risk values can help peers to form a trust action on other peers and compare the trustworthiness of a set of peers. A higher trust value T_{pq} and a lower risk value R_{pq} indicate that peer q is more trustworthy in view of peer p in terms of the collection evaluation from other peers and its direct transactions with peer q.

There are two usages of the trust and risk values in P2P file-sharing systems. First, a peer p can rely on a pair of trust and risk values with another peer q to determine whether to perform the next transaction with peer q. Assuming T_{pq} and R_{pq} are the trust value and the risk value of peer q in view of peer p, respectively. A simple rule for peer p to form a trust action on peer q can be $T_{pq} \geqslant T_{threshold}(p)$ and $R_{pq} \leqslant R_{threshold}(p)$, where $T_{threshold}(p)$ and $R_{threshold}(p)$ are the trust threshold value and the risk threshold value for peer p to trust other peers, respectively. The factors that determine these two threshold values include the extent to which peer p is willing to trust others, the importance of the sharing files in peer p. For example, a good file may own both higher trust threshold value and lower risk threshold value. More complex decision rules can be applied, but are not our focus in this paper.

The second usage is to compare the trustworthiness of a list of peers. For example, a peer who issues a file download request can first choose a set of potential peers from the peers who respond to this request according to its two threshold values. Then, it can compare the trustworthiness of the potential peers based on their trust and risk values and select the optimal peer to download the file. By doing this, it can reduce the risk of downloading inauthentic or corrupted files from untrustworthy peers. However, how do we compare the trustworthiness of two potential peers – one with higher trust value, but the other with lower risk value? Hence, we need strike a good balance between trust and risk. For example, if we give the same weight to them, the peer who with the bigger value of (T - R) will be regarded more credible, where T denotes trust value and R means risk value.

From the above analysis, we can see that the peer that has the biggest trust value will not be the optimal choice to provide service all the time. When a peer is suddenly spoiling or intermittently spiteful, although the peer may have a strongly positive trust by a large number of good transactions in the past, its risk is also increase obviously because of the fluctuation of its actions. Hence, the security of systems can be improved effectively by introducing risk factor.

4. Performance Evaluation

We perform a series of experiments to evaluate the NatureTrust approach and show its effectiveness and robustness against different malicious behaviors of peers.

4.1. Simulation Setup

We use the simulator PeerSim [19] for evaluating the performance of NatureTrust. In our simulation, we use BRITE [20,21] to generate P2P network with 100 peers, and the average number of links of each node is 2. We distribute 100 files to these 100 peers and each peer has about 10 different files. In other words, each file has about 10 replicas. We split peers into two types, namely, good peers and malicious peers. The percentage of malicious peers is denoted by k. The behavior pattern for good peers is to always cooperate in transactions, while malicious peers' behavior pattern depends on their types. In this paper, we mainly discuss three types of malicious peers: simple malicious peer who may deceive other peers at random, traitor who may behave properly and attain a high trust for a period of time, but begin defecting suddenly, hypocritical peer who may strategically alter its behavior in a way that benefits itself such as starting to behave maliciously with a certain probability after it builds up a strongly positive trust.

In our simulation, we classify trust grade into 6 types which has been introduced in Subsection 3.1, and the trust criteria are "file quality", "download speed" and "respond time". All peers use the same inference rules which will not be listed here in detail. Besides, in peer selection scheme, we give the same weight to trust and risk.

For each experiment in the following, the experiment environment is initialized by performing 1000 transactions among peers randomly. Then, each peer initializes its trust and risk threshold values according to its own situation. For example, the peer with high trust value can set high trust threshold value for its sharing files, while the peer with low trust value can set low trust threshold value for its sharing files. Finally, every peer, in turn, issues a request for some file to the community until the number of transactions achieves 6000. Important to note that if a peer who initiates a request for some file can not locate an appropriate peer to do transaction, the peer will give up this request and the next request from another peer will be initiated.

For comparison purpose, we also simulate XRep, PeerTrust and PET trust models. In distributed environment, an important issue is increasing the ratio of successful transaction, so we attend to compare our model with XRep, PeerTrust and PET against three types of malicious attacks. All experiment results are averaged over three runs of the experiments. Table 1 summarizes

the main parameters related to the community setting and the computation of trust and risk values. The default values are also listed.

Definition 7: Let TSR denotes transaction successful ratio which is the ratio of the number of successful transactions over the number of total transactions. Nt represents the number of total transactions and Ns denotes the number of successful transactions.

$$TSR = \frac{Ns}{Nt} \quad (7)$$

We use this metric to estimate the effectiveness of trust models against the behavior of malicious peers. The greater TSR is, the more effective the model is.

4.2. Effectiveness against Malicious Peers

In the first set of experiments, we study the transaction success rate with regard to the number of transactions under the attack of simple malicious peers. As to our data set used in experiments, we test different rate (r=0.25,0.5,0.75) of a malicious peer acting maliciously and the result is shown in Figure 2. From the figure we can see, in addition to XRep, the other three approaches have the similar transaction success rate. This is because, even if the computed trustworthiness of peers do not reflect accurately the uncertainty of the peers being cooperative, but they indeed differentiate good peers from simple malicious peers in most cases by the ranking of trust value. XRep is less efficient than other approaches, for it does not consider the credibility of voters. Furthermore, we also observe that the bigger the rate r is, the faster the malicious peers are exposed. Accordingly, the growth of success rate is quicker.

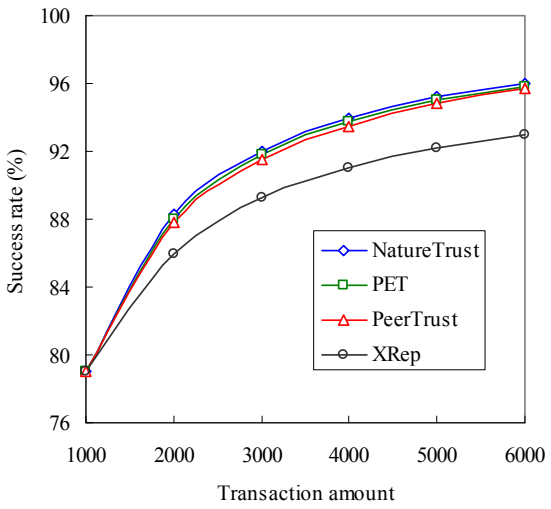


Figure 2. Compare success rate under the attack of simple malicious peers.

The second experiment (Figure 3) shows the variation of the transaction success rate with the increase of transaction amount under the attack of traitors. In this experiment, we presume the malicious peers start deceiving behavior once their trust value is bigger than T_θ=0.8. In this figure, we see an obvious superiority of the transaction success rate in PET and our approach with risk factor. This confirms that supporting risk is an important feature in a P2P community as peers can able to avoid the attack of suddenly spoiling peers. Moreover, another observation is that the success rate firstly decreases, and then increases as the increase of transaction amount. The reason is as follows. At the beginning, malicious peers almost act kindly in order to improve their trust value. Once their trust value is big enough, they can start deception. Hence, the success rate firstly decreases. However, as the malicious peers behaving maliciously, they expose themselves gradually, so the success rate increases subsequently.

In the third experiment (Figure 4), we discuss the variation of the transaction success rate as the number of transaction increasing from 1000 to 6000 under the attack of hypocritical peers. In this experiment, we presume the hypocritical peers strategically alter its behav-

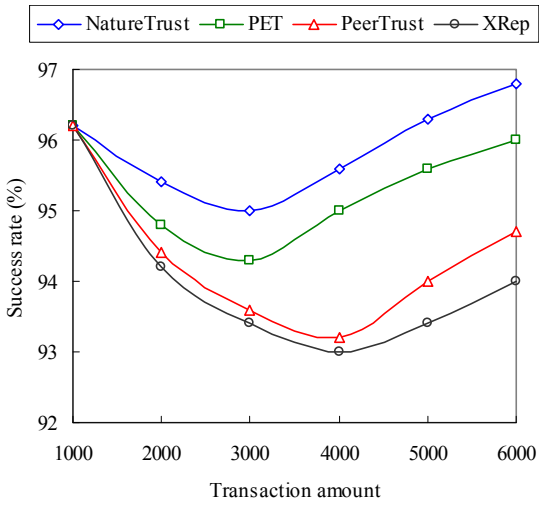


Figure 3. Compare success rate under the attack of traitors.

Table 1. Simulation parameters.

Parameter name	Parameter description	Default value
P	The number of peers in the community	100
k	The percentage of malicious peers	30%
F	The number of files	100
S	The number of replicas for each file	10
N	The number of trust grades	6
R_0	The initial value of risk	0.4
w	The weight factor	0.7
μ	Time declining constant	0.2
λ	The weight factor	0.8

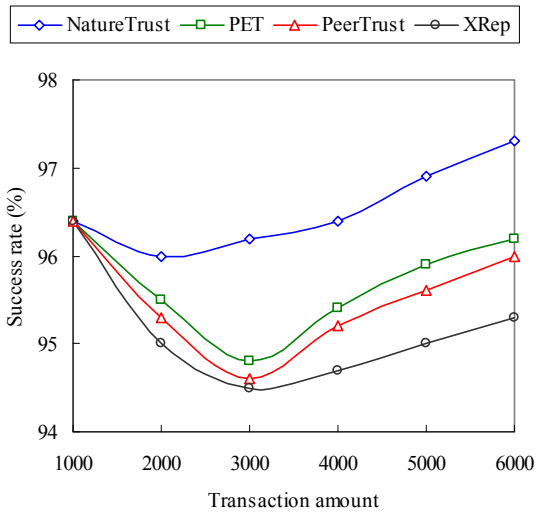


Figure 4. Compare success rate under the attack of hypocritical peers.

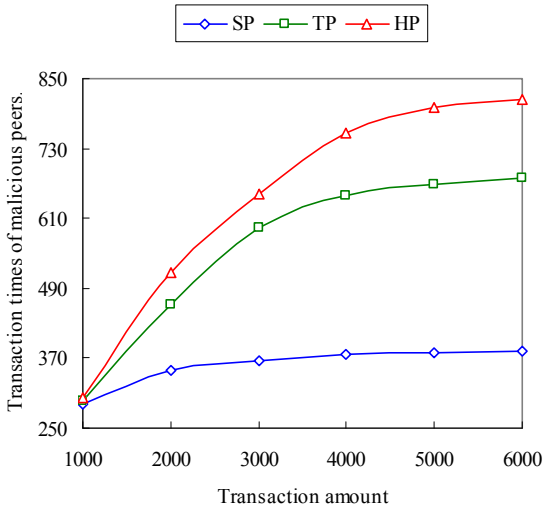


Figure 5. The benefit of isolating malicious peers.

ior in the way that they start to behave maliciously with a certain probability Pr=0.3 after it builds up a strongly positive trust value T_α=0.85. It is clear that the gain of the transaction success rate in NatureTrust is more obvious than that of the other three approaches, which illuminates that the evaluation of peers' trustworthiness in NatureTrust is more effective than others against the attack of hypocritical peers.

In Figure 5, we show the variation of the transaction times of malicious peers as the gain of transaction amount under the attack of three types of malicious peers using our trust model, where SP denotes simple malicious peers with r=0.5, TP denotes traitors with T_θ=0.8 and HP means hypocritical peers with T_α=0.85 and Pr=0.3. We can see that the growth of transaction times of malicious peers nearly stops when the total transaction amount is bigger than 5000 under the attack of three types of malicious peers. This means that three types of malicious peers are isolated quickly in our approach. Therefore, our trust model is beneficial to restraining the malicious behavior of peers.

5. Conclusions

In this paper, we analyze the nature of trust. We apply linguistic terms to express trust and employ fuzzy inference rules to evaluate trust. The fuzzy inference adopted in this paper restrains the unfair appraisements to some extent, for peers can obtain trust values according to the same inference rules. Thus, the security of the system is improved. Furthermore, risk factor is deployed to reason with the dynamic characteristic of trust. The application of the risk scheme aims to solve the security problems, such as traitor and hypocritical behavior, for the risk value increases as soon as the peer defects. Though its trust value can't decrease obviously, we can also detect the malicious act relying on risk value. In the end, the experiments show that the proposed trust model is more efficient than XRep, PET and PeerTrust.

As for our future work, we will continue to perfect the NatureTrust. We will consider other cheating or vicious behaviors in P2P file-sharing systems, and further research other methods to detect such behaviors.

6. Acknowledgements

This work is supported by the National Natural Science Foundation of China (60403027, 60773191, 60873225), the National High Technology Research and Development Program of China (863 Program) (2007AA01 Z403).

7. References

[1] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust algorithm for reputation management in P2P networks," in Proceedings of the 12th International World Wide Web Conference, pp. 640–651, 2003.

[2] L. Xiong and L. Liu, "A reputation-based trust model for peer-to-peer e-commerce communities," in Proceedings of the IEEE International Conference on E-Commerce, 2003.

[3] L. Xiong and L. Liu, "PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities," in Proceedings of IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 7, pp. 843–857, 2004.

[4] E. J. Chang, F. K. Hussain, and T. S. Dillon, "Fuzzy nature of trust and dynamic trust modeling in service oriented environments," in Proceedings of the 2nd ACM

Workshop on Secure Web Services (SWS'05), Fairfax, Virginia, USA, 2005.

[5] D. W. Manchala, "E-commerce trust metrics and models," IEEE Internet Computing, Vol. 4, No. 2, 2000.

[6] L. Xiong and L. Liu, "A reputation-based trust model for peer-to-peer e-commerce communities," in Proceedings of the IEEE Conference on E-Commerce, June 2003.

[7] P. Resnick and R. Zeckhauser, "Trust among strangers in Internet transactions: Empirical analysis of eBay's Reputation system," in Proceedings of NBER Workshop on Empirical Studies of Electronic Commerce, 2000.

[8] F. Cornelli, E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati, "Choosing reputable servents in a P2P network," in Proceedings of the 11th World Wide Web Conference, 2002.

[9] E. Damiani, S. Vimercati, S. Paraboschi, P. Samarati, and F. Violante, "A Reputation-based approach for choosing reliable resources in peer-to-peer networks," in Proceeding of CCS, 2002.

[10] S. Kamvar, M. Schlosser, and H. Garcia-Molina, "EigenTrust: Reputation management in P2P networks," in Proceedings of the 12th WWW Conference, 2003.

[11] F. Cornelli, E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati, "Choosing reputable servents in a P2P network," in Proceedings of the 11th International World Wide Web Conference, pp. 376−386, 2002.

[12] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati, "Managing and sharing servents' reputations in P2P systems," in proceedings of IEEE Transactions on Knowledge and Data Engineering, pp. 840−854, 2003.

[13] E. Damiani, S. D. C. di Vimercati, S. Paraboschi, P. Samarati, and F. Violante, "A reputation-based approach for choosing reliable resources in Peer-to-Peer networks," in Proceedings of the 9th ACM Conference on Computer and Communications Security, pp. 207−216, 2002.

[14] M. Richardson, R. Agrawal, and P. Domingos, "Trust management for the semantic web," in Proceedings of the 2nd International Semantic Web Conference, pp. 351−368, 2003.

[15] Z. Q. Liang and W. S. Shi, "PET: A personalized trust model with reputation and risk evaluation for P2P resource sharing," in Proceedings of the 38th Hawaii International Conference on System Sciences, 2005.

[16] G. W. Zhang, J. H. Kang, and R. He, "Towards a trust model with uncertainty for e-commerce systems," in proceedings of the IEEE International Conference on e-Business Engineering, 2005.

[17] R. He, J. W. Niu, and K. Hu, "A novel approach to evaluate trustworthiness and uncertainty of trust relationships in Peer-to-Peer computing," in proceedings of the 5th International Conference on Computer and Information Technology (CIT'05), 2005.

[18] D. Y. Li, "The cloud control method and balancing patterns of triple link inverted pendulum systems," Chinese Engineering Science, Vol. 1, No. 2, pp. 41−46, 1999.

[19] PeerSim: A peer-to-peer simulator. http://peersim. source forge.net/.

[20] BRITE, http://www.cs.bu.edu/brite/, 2007.

[21] A. Medina, A. Lakhina, I. Matta, et al., "BRITE: Universal topology generation from a user's perspective," Technical Report BUCS-TR-2001-003, Boston University, April 2001.

Int. J. Communications, Network and System Sciences, 2009, 3, 169-247
doi:10.4236/ijcns.2009.23026 Published Online June 2009 (http://www.SciRP.org/journal/ijcns/).

An Improved Analytical Model for IEEE 802.11 Distributed Coordination Function under Finite Load

Rama Krishna CHALLA[1], Saswat CHAKRABARTI[2], Debasish DATTA[3]
[1]*Department of Computer Science, National Institute of Technical Teachers' Training and Research, Chandigarh, India*
[2]*G. S. Sanyal School of Telecommunications, Indian Institute of Technology, Kharagpur, India*
[3]*Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology, Kharagpur, India*
Email: *rkc@ece.iitkgp.ernet.in, rkc_97@yahoo.com*
Received November 27, 2009; *revised March* 26, 2009; *accepted March* 29, 2009

ABSTRACT

In this paper, an improved analytical model for IEEE 802.11 distributed coordination function (DCF) under finite load is proposed by closely following the specifications given in IEEE 802.11 standard. The model is investigated in terms of channel throughput under perfect and slow Rayleigh fading channels. It is shown that the proposed model gives better insight into the operation of DCF.

Keywords: IEEE 802.11, Markov, DCF, Wireless LANs, Backoff, Perfect Channel, Rayleigh Fading Channel, Saturation, Finite Load, Throughput

1. Introduction

IEEE 802.11 has been standardized and widely adopted for wireless local area networks (WLANs) [1]. In this standard, it specifies two fundamental access mechanisms, i.e., point coordination function (PCF) and distributed coordination function (DCF). Since IEEE 802.11 DCF mechanism supports adhoc networking configuration and has been widely adopted in wireless networks, we focus our analysis only on this mechanism. DCF mechanism is based on the carrier sense multiple access with collision avoidance (CSMA/CA) protocol. In DCF, data frames are transmitted via two mechanisms, i.e., basic access mechanism and request-to-send/clear-to-send (RTS/CTS) mechanism.

Performance analysis of DCF has been reported in several research works through either simulation or mathematical modeling [2−13]. The Markov model proposed in [2] for IEEE 802.11 DCF has gained wide acceptance due to its simplicity. However, it exhibits some constraints according to [1]. First, the model is limited to deriving the saturation throughput. It excludes the performance analysis under finite load condition, which is an important practical scenario in a WLAN. Secondly, it does not take into account the loss of frames due to channel contention. This frame loss has been shown to be significant in [10]. Finally, decrementing a backoff value by a station (STA) is not modeled correctly as per 802.11 standard [1].

In the literature, some investigations have been reported on finite load models for IEEE 802.11 DCF, [3,4,11−14]. In [4], the authors extend the model reported in [2], for finite load by introducing a new state accounting for the case in which STA's queue is empty after successful transmission of a packet. Throughput has been expressed as a function of queue utilization under the perfect channel assumption. In [3], a queuing model has been proposed to study delay and queue length characteristics at each STA under finite load conditions. In [11,12], authors propose a Markov model for characterizing the IEEE 802.11 DCF behavior by including transmission states that account for packet transmission failures due to errors caused by propagation through channel. Also, a state has been introduced characterizing the situation when an STA has no packets to transmit. In [13], authors proposed a Markov model for limited load by adding a new state for each backoff stage accounting for the absence of new packets to be transmitted. In [14], a Markov model is proposed to analyze the IEEE 802.11 DCF under finite load. Performance has been analyzed in terms of channel throughput and average packet delay under perfect and a slow Rayleigh fading channel. How-

ever, in all the proposed models, decrementing a backoff value by an STA is not correctly modeled according to [1]. As per the standard [1], an STA in any backoff stage should decrement its backoff counter value only when the channel status is found to be idle for at least DCF inter-frame space (DIFS) duration. Whereas in the proposed models, an STA decrements its backoff counter value irrespective of the channel status i.e., whether the channel is busy or idle, which is not complying with [1]. In this work, we follow the models in [2,14]. Hereafter, we refer to model in [2] as Bianchi's model and model in [14] as Pham's model. Readers are requested to refer [1] for a detailed discussion on IEEE 802.11 DCF operation.

In this paper, we propose an improved Markov model for IEEE 802.11 DCF under finite load by closely following the specifications in [1]. The model is investigated in terms of channel throughput under perfect and slow Rayleigh fading channels for different packet arrival rate and number of STAs in the network.

The rest of the paper is organized as follows: Section 2 describes the proposed Markov model for IEEE 802.11 DCF under finite load, followed by the performance analysis for perfect channel conditions. The model developed in Section 2 is extended for a slow Rayleigh fading channel and the performance is analyzed in Section 3. Finally Section 4 presents the concluding remarks on the work.

2. Markov Model for IEEE 802.11 DCF

As discussed above, the Markov models presented in literature have shortcomings. Complementing the work in [2,14], we focus on the performance analysis of IEEE 802.11 DCF under finite load condition. The saturation throughput considered in [2] is just one particular case of our analysis. We divide our contribution into two parts. In the first part, we propose an improved Markov model for DCF assuming perfect channel conditions. Next, we extend this model for a slow Rayleigh fading channel. For simplicity, we use the same notation as given in [14].

2.1. Proposed Markov Model for IEEE 802.11 DCF

Let n be the number of stations (STAs) in a WLAN contending for channel access. Let $b(t)$ and $s(t)$ be the stochastic processes representing the backoff counter and number of the backoff stages respectively. The backoff states and its transition probabilities for a given STA are shown in Figure 1. The parameters used in our model are described in Table 1. We assume that the channel is perfect (i.e., error free) and the packets are lost only due to collisions.

Under saturation condition, an STA always has a packet for transmission in its queue. Therefore, it enters straight away to state $S_{0,k}$, $0 \le k \le W_0 - 1$. However, under finite load, if the STA's queue is empty, the station enters into one of the states $S_{0',k}$, $0 \le k \le W_0 - 1$, otherwise the STA enters one of the backoff states $S_{0,k}$, $0 \le k \le W_0 - 1$. At state $S_{0',0}$, if there is a packet for transmission, the STA starts transmitting the packet by moving to the state $S_{0,0}$ with probability $P_{0',0}$. Otherwise, the STA enters the state $S_{idle,0}$ with probability $P_{0',idle}$. At state $S_{idle,0}$ once a frame arrives into the queue of a STA and if the channel has been found to be idle for more than DIFS, this frame is transmitted immediately with probability $P_{idle,0}$. Otherwise, the STA goes to backoff state $S_{0,k}$, $0 \le k \le W_0 - 1$ with probability $P_{idle,b}$.

The state of each STA is described by $b(i,k)$, where i indicates the current backoff stage, $0 \le i \le m_1$ and k is the backoff counter value measured in time slots, $0 \le k \le W_i - 1$.

Table 1. Summary of parameters.

P_b	Probability that an STA in the backoff stage senses the channel busy
p	Probability of a packet not received successfully
W_i	Backoff window size at stage $s(t)=i$
W_0	Minimum backoff window size
q	Probability that the queue is empty
$b(i, j)$	Probability of an STA in the backoff state $S_{i,j}$
b(idle, 0)	Stationary probability at idle state $S_{idle,0}$
$S_{idle,0}$	Channel is in idle state
τ	Probability of STA transmitting a packet
$S_{i,j}$	State of STA when $s(t)=i$ and $b(t)=j$
λ	Average packet arrival rate
$P_{0',0}$	Transition probability from state $S_{0',0}$ to $S_{0,0}$
$P_{idle,0}$	Transition probability from state $S_{idle,0}$ to $S_{0,0}$
$P_{idle,b}$	Transition probability from state $S_{idle,0}$ to backoff state
$P_{0',idle}$	Transition probability from state $S_{0',0}$ to $S_{idle,0}$
$P_{tr(n)}$	Probability of at least one out of n STAs transmit
$P_{s(n)}$	Probability of successful transmission for n STAs
$\overline{\sigma}$	Average slot time
$\overline{\sigma_s}$	Average slot time at saturation
U_{sta}	Channel throughput per station
U_{total}	Total normalized channel throughput
Q_l	Queue length
μ_{eff}	Effective packet service rate
μ_{succ}	Rate of successful transmissions
μ_{disc}	Rate at which packets are being discarded
P_{disc}	Probability that the packet is discarded
ρ	Traffic intensity
σ	Channel idle slot
δ	Channel Propagation delay

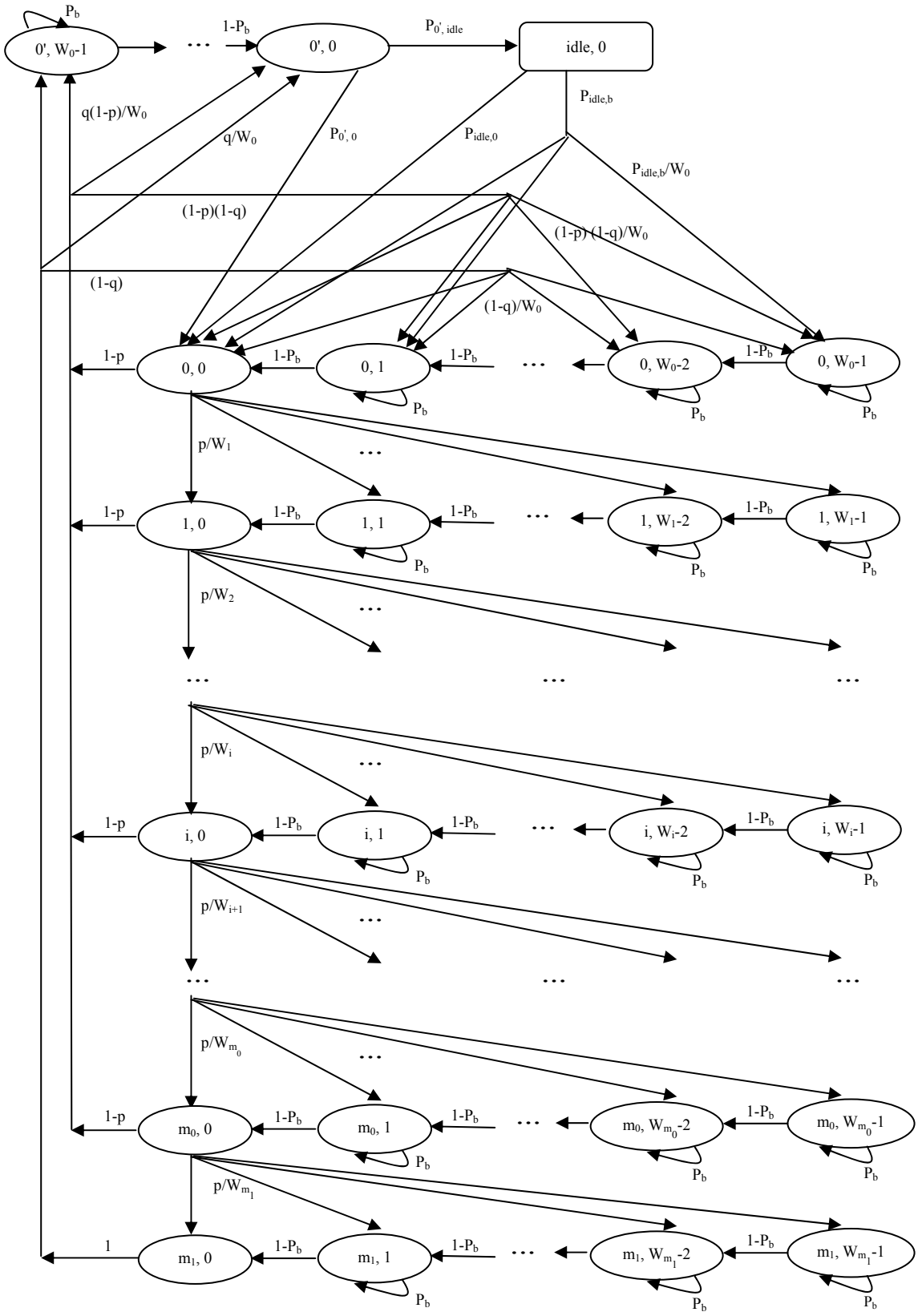


Figure 1. Two dimensional Markov chain for IEEE 802.11 DCF backoff mechanism.

The Markov chain in Figure 1 is described in the following:

1) The backoff counter value is decremented when the STA senses the channel is idle for at least DIFS duration:

$$b\left(i,k\left|i,k+1\right.\right)=1-P_b,\ 0\le i\le m_1 \text{ and } 0\le k\le W_i-2$$

where $P_b=1-(1-\tau)^n$ is the probability that the channel is found to be busy when at least one of the STAs transmits at a given slot time.

2) The backoff counter value is frozen whenever the STA senses that the channel is busy:

$$b\left(i,k\left|i,k\right.\right)=P_b,\ 0\le i\le m_1 \text{ and } 1\le k\le W_i-1$$

The above steps (1) and (2) are not included in the existing Markov models.

3) When at least one new frame has arrived at STA's queue during mandatory backoff stage (i.e., $S_{0',k}$, $0\le k\le W_0-1$) then $b\left(0,0\left|0',0\right.\right)=P_{0',0}$.

4) If no frames have arrived during mandatory backoff stage and the STA enters into the IDLE state:

$$b\left(idle,0\left|0',0\right.\right)=P_{0',\,idle}$$

5) If at least one frame has arrived when an STA is in the IDLE state and the station senses the channel is idle for at least DIFS duration: $b\left(0,0\left|idle,0\right.\right)=P_{idle,0}$.

6) When at least one frame has arrived and the STA senses the channel is busy:

$$b\left(0,k\left|idle,0\right.\right)=\frac{P_{idle,b}}{W_0},\ 0\le k\le W_0-1$$

7) The frame is not transmitted successfully by an STA in a backoff stage:

$$b\left(i,k\left|i-1,0\right.\right)=\frac{p}{W_i},\ 1\le i\le m_0,\ 0\le k\le W_i-1$$

8) The frame has been transmitted successfully and there is at least one more packet in the queue of an STA:

$$b\left(0,k\left|i,0\right.\right)=\begin{cases}\dfrac{(1-p)(1-q)}{W_0}, & 0\le i\le m_0,\ \ 0\le k\le W_0-1\\ \dfrac{(1-q)}{W_0}, & i=m_1,\ 0\le k\le W_0-1\end{cases}$$

9) The frame is transmitted successfully or lost in collision at m_1^{th} backoff stage and there is no packet in the queue:

$$b\left(0',k\left|i,0\right.\right)=\begin{cases}\dfrac{q(1-p)}{W_0}, & 0\le i\le m_0,\ 0\le k\le W_0-1\\ \dfrac{q}{W_0}, & i=m_1,\ 0\le k\le W_0-1\end{cases}$$

Next, we derive the closed-form solution for the proposed Markov model. In the steady-state, let $b(i,k)=\lim_{t\to\infty}\Pr\{s(t)=i,\,b(t)=k\}$ be the stationary distribution of the Markov chain. All steady-state probabilities are expressed as a function of $b(0,0)$.

We observe that,

$$b(i,0)=p^i\,b(0,0),\ 1\le i\le m_1 \qquad \textbf{(1)}$$

Or

$$b(i,0)=p\,b(i-1,0),\ 1\le i\le m_1$$

From (1), it is easy to derive the following,

$$\sum_{i=0}^{m_0}b(i,0)=\frac{b(0,0)(1-p^{m+1})}{1-p} \qquad \textbf{(2)}$$

And also, we can find that,

$$b(m_1,0)=p^{m+1}\,b(0,0) \qquad \textbf{(3)}$$

For $1\le k\le W_i-1,\ 0\le i\le m_1$, we can derive the following,

$$b\,(i,k)=\left(\frac{W_i-k}{W_i}\right)\left(\frac{p}{1-P_b}\right)b(i-1,0) \qquad \textbf{(4)}$$

when $i=k=0$, we obtain,

$$b\,(0,0)=b\,(0',0)\,P_{0',0}+b\,(idle\,,0)\,P_{idle,0}+b\,(idle,0)\,P_{idle,b}+(1-p)\,(1-q)\sum_{i=0}^{m_0}b\,(i,0)+(1-q)\,b\,(m_1,0) \qquad \textbf{(5)}$$

For $1\le k\le W_0-1$, we get,

$$b\,(0,k)=\left(\frac{W_0-k}{W_0}\right)\left(\frac{1}{1-P_b}\right)\left\{b\,(idle,0)P_{idle,\,b}+(1-p)\,(1-q)\sum_{i=0}^{m_0}b\,(i,\,0)+(1-q)\,b\,(m_1,0)\right\} \qquad \textbf{(6)}$$

And also, for $1\le k\le W_0-1$,

$$b\,(0',\,k)=\left(\frac{W_0-k}{W_0}\right)\left(\frac{1}{1-P_b}\right)\left\{q\,(1-p)\sum_{i=0}^{m_0}b\,(i,0)+qb(m_1,0)\right\} \qquad \textbf{(7)}$$

Substituting (2) and (3) in (7), we obtain,

$$b(0',k)=\left(\frac{W_0-k}{W_0}\right)\left(\frac{q}{1-P_b}\right)b(0,0) \qquad \textbf{(8)}$$

And also, we can show that,

$$b(0',0) = q\,b(0,0) \tag{9}$$

We observe that,

$$P_{0',\,idle}\,b(0',0) = b(idle,0)(P_{idle,0} + P_{idle,b}) \tag{10}$$

Substituting (9) in (10) and using the relation $P_{idle,0} + P_{idle,b} = 1$ yields,

$$b(idle,0) = q\,P_{0',idle}\,b(0,0) \tag{11}$$

Under steady state, $b(0,0)$ is determined by imposing the normalizing condition,

$$A + B + C = 1 \tag{12}$$

where $A = \sum_{i=0}^{m_1}\sum_{k=0}^{W_i-1} b(i,k)$, $B = \sum_{k=0}^{W_0-1} b(0',k)$ and $C = b(idle,0)$

$$\begin{aligned} A = \sum_{i=0}^{m_1}\sum_{k=0}^{W_i-1} b(i,k) &= \sum_{i=1}^{m_1}\sum_{k=0}^{W_i-1} b(i,k) + \sum_{k=0}^{W_0-1} b(0,k) \\ &= \sum_{i=1}^{m_1} b(i,0) + \sum_{i=1}^{m_1}\sum_{k=1}^{W_i-1} b(i,k) + \sum_{k=0}^{W_0-1} b(0,k) \\ &= p\left(\frac{1-p^{m+1}}{1-p}\right) b(0,0) + \frac{W_0\,b(0,0)}{2(1-P_b)}\left[\sum_{i=1}^{m}(2p)^i + p(2p)^m\right] \\ &\quad - \frac{p(1-p^{m+1})\,b(0,0)}{2(1-P_b)(1-p)} + b(0',0)\,P_{0',0} + b(idle,0)\,P_{idle,0} \\ &\quad + \left(\frac{1-2P_b+W_0}{2(1-P_b)}\right)\left\{b(idle,0)P_{idle,b} + (1-p)(1-q)\sum_{i=0}^{m_0} b(i,0) + (1-q)\,b(m_1,0)\right\} \end{aligned} \tag{13}$$

$$B = \sum_{k=0}^{W_0-1} b(0',k) = q\,b(0,0)\left(\frac{1+W_0-2P_b}{2(1-P_b)}\right) \tag{14}$$

C can be expressed as:

$$C = b(idle,0) = \frac{q\,P_{0',idle}\,b(0,0)}{P_{idle,0} + P_{idle,b}} \tag{15}$$

Substituting (13), (14) and (15) in (12), we obtain,

$$\begin{aligned} &p\,b(0,0)\left(\frac{1-p^{m+1}}{1-p}\right)\left(\frac{(1-2P_b)}{2(1-P_b)}\right) + \frac{W_0\,b(0,0)}{2(1-P_b)}\left[\sum_{i=1}^{m}(2p)^i + p(2p)^m\right] \\ &+ b(0',0)\,P_{0',0} + b(idle,0)P_{idle,0} + \left(\frac{1-2P_b+W_0}{2(1-P_b)}\right) \\ &\times\left\{b(idle,0)\,P_{idle,b} + (1-p)(1-q)\sum_{i=0}^{m_0} b(i,0) + (1-q)\,b(m_1,0)\right\} \\ &+ q\,b(0,0)\left(\frac{1+W_0-2P_b}{2(1-P_b)}\right) + \frac{q\,P_{0',idle}\,b(0,0)}{P_{idle,0}+P_{idle,b}} = 1 \end{aligned} \tag{16}$$

Therefore, $b(0,0)$ can be obtained as,

$$b(0,0) = \left(\begin{aligned} &\frac{1}{2(1-P_b)}\left[W_0\left(\sum_{i=1}^{m}(2p)^i + p(2p)^m\right) + p(1-2P_b)\left(\frac{1-p^{m+1}}{1-p}\right)\right] \\ &+ q\,P_{0',0} + \frac{q\,P_{0',idle}\,P_{idle,0}}{P_{idle,0}+P_{idle,b}} + \left(\frac{1-2P_b+W_0}{2(1-P_b)}\right)\left\{\frac{q\,P_{0',idle}\,P_{idle,b}}{P_{idle,0}+P_{idle,b}} + (1-q)\right\} \\ &+ \frac{q(1-2P_b+W_0)}{2(1-P_b)} + \frac{q\,P_{0',idle}}{P_{idle,0}+P_{idle,b}} \end{aligned}\right)^{-1} \tag{17}$$

After rearranging some terms, we can write $b(0,0)$ as,

$$b(0,0)=\left(\begin{array}{l}\dfrac{1}{2(1-P_b)}\left[\begin{array}{l}W_0\left(\sum_{i=1}^{m}(2p)^i+p(2p)^m\right)+\\ p(1-2P_b)\left(\dfrac{1-p^{m+1}}{1-p}\right)\end{array}\right]\\ +\\ \dfrac{q(2P_{0',0}-2P_{0',0}P_b+(1+W_0-2P_b))}{2(1-p_b)}+\\ \dfrac{qP_{0',idle}(P_{idle,0}+1)}{P_{idle,0}+P_{idle,b}}+\\ \left(\dfrac{1-2P_b+W_0}{2(1-P_b)}\right)\left\{\dfrac{qP_{0',idle}P_{idle,b}}{P_{idle,0}+P_{idle,b}}+(1-q)\right\}\end{array}\right)^{-1} \quad \textbf{(18)}$$

It is important to note that, if we substitute $P_b = 0$ in (18), proposed model reduces to Pham's model in [14]. Further, by letting q = 0 (i.e., STA's queue is never empty) and introducing the constraint that packet will never be dropped even after reaching maximum retry limit, Equation (18) reduces to the same expression as in [2]. This confirms the fact that we have covered Bianchi's model also.

Using $M/M/1/Q_l$ queuing model in [15], the probability that the queue of any STA is empty is,

$$q=\frac{1-\dfrac{\lambda}{\mu_{eff}}}{1-\left(\dfrac{\lambda}{\mu_{eff}}\right)^{Q_l+1}} \quad \textbf{(19)}$$

where λ is the average packet arrival rate and μ_{eff} is the effective packet service rate.

The probabilities $P_{idle,0}$, $P_{0',idle}$, $P_{idle,b}$ and $P_{0',0}$ are same as in [14]. IEEE 802.11 packet format consists of an actual payload (P_L) and header information (H). We know that the channel can be in any of the three states in a slot, i.e., idle (σ) or busy due to successful transmission (T_s) or busy due to packet collisions (T_c).

Using the basic access mechanism of IEEE 802.11 DCF, T_s and T_c can be calculated as:

$$\begin{aligned}T_s &= H+P_L+SIFS+\delta+ACK+DIFS+\delta\\ T_c &= H+P_L+DIFS+\delta\end{aligned} \quad \textbf{(20)}$$

where SIFS, ACK, DIFS and δ are the short inter-frame space, acknowledgement, DCF inter-frame space and channel propagation delay respectively.

Because each STA transmits with probability τ, we have the following expressions:

$$\begin{aligned}P_{tr(n)} &= 1-(1-\tau)^n\\ P_{tr(n)}P_{s(n)} &= n\tau(1-\tau)^{n-1}\end{aligned} \quad \textbf{(21)}$$

Each time slot has the probability $(1-P_{tr(n)})$ of being idle, $P_{tr(n)}P_{s(n)}$ of having a successful transmission and $P_{tr(n)}(1-P_{s(n)})$ of having a packet collision. Therefore, the average slot time can be calculated as:

$$\overline{\sigma}=(1-P_{tr(n)})\sigma+P_{tr(n)}P_{s(n)}T_s+P_{tr(n)}(1-P_{s(n)})T_c \quad \textbf{(22)}$$

The states $S_{m_{0,j}}$ and $S_{m_{1,j}}$, $0 \le j \le W_m-1$, represent the last two backoff stages as shown in Figure 1. According to [1], the sending STA attempts to send the DATA packet under basic access scheme for station short retry count times before discarding the packet. Therefore, $S_{m_{1,0}}$ is the state where a given packet is either transmitted successfully with probability $(1-p)$ or permanently discarded with probability p. Furthermore, denoting m_0 and m_1 as the indices of the last two backoff stages, we have,

$$\begin{aligned}W_{m_0} &= W_m = 2^m W_0\\ W_{m_1} &= W_m = 2^m W_0\end{aligned} \quad \textbf{(23)}$$

The probability for a given STA to transmit can be easily derived by noticing that the STA can only transmit after its backoff timer expires, that is,

$$\tau=\sum_{i=0}^{m_1}b(i,0)=\frac{b(0,0)(1-p^{m+2})}{1-p} \quad \textbf{(24)}$$

The transmitted packet is not received correctly by the receiver when at least two STAs transmit at the same time. In other words, this is equal to the probability of at least one out of $(n-1)$ remaining STAs transmits, that is,

$$p = 1-(1-\tau)^{n-1} \quad \textbf{(25)}$$

We can rewrite (25) and obtain transmission probability as,

$$\tau = 1-(1-p)^{1/(n-1)} \quad \textbf{(26)}$$

Using (24) and (26), p and τ are readily obtained using numerical analysis.

2.2. Channel Throughput

In this section, we derive the expression for channel throughput which is the performance metric to evaluate our proposed model. For a finite load where the packet arrival rate (λ) is less than effective packet service rate (μ_{eff}), the STA's throughput (U_{sta}) is the portion of traffic that arrived minus the portion that is discarded, i.e., $\lambda(1-P_{disc})$, where P_{disc} is the probability that the packet is discarded. Here, we assume an M/M/1/Q_l model for transmission queue, hence packet arrival rate (λ), throughput at an STA (U_{sta}), and the total normalized throughput for a neighborhood of n identical STAs (U_{total}), are given by (27). Further, $\mu_{eff} = \mu_{succ} + \mu_{disc}$, where μ_{succ} indicates the rate of successful transmissions and μ_{disc} indicates the rate at which packets are being discarded. Therefore, traffic intensity (ρ)= $\frac{\lambda}{\mu_{eff}}$ and,

$$U_{sta} = \begin{cases} \rho \cdot \mu_{succ} & for \ \ \rho < 1 \\ \mu_{succ} & for \ \ \rho \geq 1 \end{cases}$$

$$U_{total} = \frac{n \cdot U_{sta} \cdot P_L}{Channel\ data\ rate} \quad \textbf{(27)}$$

For a large queue length (Q_l) and $\rho < 1$, U_{sta} is taken equivalent to λ since P_{disc} is negligible. Under saturation condition (i.e., $\rho \geq 1$), maximum successful packet transmission rate (μ_{succ}) is given by,

$$\mu_{succ} = \frac{\tau_s(1-\tau_s)^{n-1}}{\overline{\sigma_s}} \quad \textbf{(28)}$$

where $\overline{\sigma_s}$ is the average slot duration and τ_s is the probability of packet transmission by an STA at saturation.

2.3. Performance Analysis of the Proposed Model

In this section, we present the results and discuss the variation of channel throughput with different packet arrival rate and different number of STAs in a network under perfect channel assumption. The parameters used in the evaluation of our proposed analytical model are same as in [14] and are reproduced in Table 2 for ready

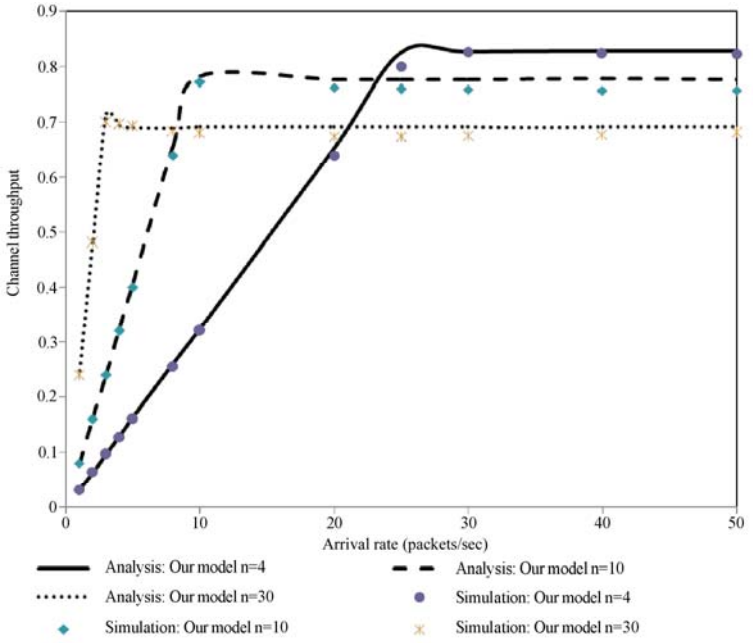


Figure 2. Channel throughput vs. packet arrival rate with varying number of STAs.

reference. In our analysis, we have considered basic access mechanism of DCF. However, it is easy to extend our analysis for RTS/CTS-based access mechanism of DCF as well.

In order to verify our proposed analytical model, we compare theoretical analysis discussed earlier to the simulation results in Figure 2. This figure illustrates that the simulation results agree well with analytical results.

In Figure 3 we observe the impact of packet arrival rate on channel throughput as number of STAs varies. It is clear that for a given number of STAs in the network, increase in the packet arrival rate increases the channel throughput linearly in both models as long as the packet arrival rate is less than packet processing rate. However, as the channel throughput reaches its maximum, further increase in packet arrival rate makes channel throughput to saturate. This is because all STAs have a packet to transmit at any given slot. That is, packet arrival rate is more than the packet processing rate, which causes the saturation of throughput. It is observed that proposed model shows increase in throughput with an improvement of 1 to 4% compared to Pham's model [14] with increase in number of STAs (n) from 4 to 30. This improvement is due to the checking of channel status in our proposed model before decrementing the backoff counter value according to [1], which in turn decreases the probability of collisions and hence increases the probability of successful transmissions. This is also confirmed in Figure 5. However, at low packet arrival rate, performance of the proposed model is similar to Pham's model. This is because the number of competing STAs is small and also the packet arrival rate is less, and hence channel status may be idle for most of the time. We also find that maximum throughput value gets shifted to a lower value with increase in number of STAs. This is obvious as the number of competing STAs increases, probability of packet collisions increases consequently.

Table 2. Summary of IEEE 802.11 DCF parameters.

Payload (P_L)	8000 bits	Channel bit rate	1 Mbps
Headers	576bits	Prop. Delay	2μs
ACK	320bits	Slot Time	20μs
DIFS	50μs	SIFS	10μs
Q_l (Queue length)		50	

Figure 4 shows the variation of channel throughput with number of STAs for a given packet arrival rate. We observe that increasing the number of STAs increases the channel throughput linearly till a maximum value is reached in both the models. However, further increase in number of STAs after the throughput reaches a maximum value; we observe that the throughput decreases. This is because of the fact that increase in number of STAs increases packet collisions and hence reduction in throughput. It is observed that the proposed model shows increase in throughput with an improvement of 2 to 4% compared to Pham's model with increase in number of STAs (n) from 8 to 50 for a fixed packet arrival rate of 15 packets/sec. This im-

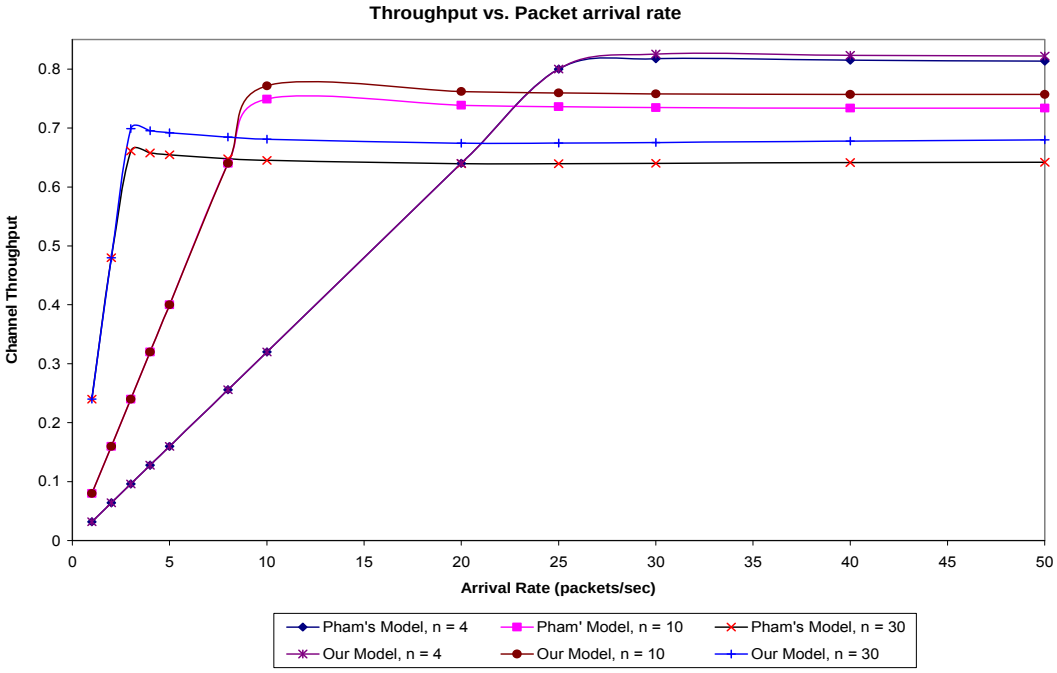


Figure 3. Channel throughput vs. packet arrival rate with varying number of STAs.

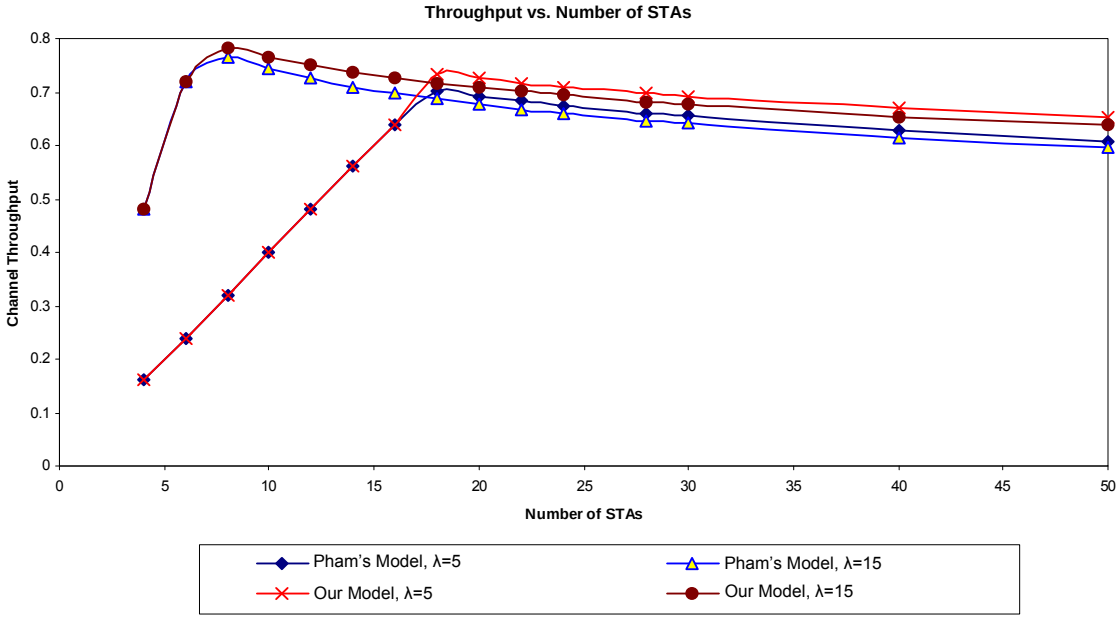


Figure 4. Channel throughput vs. number of STAs with varying packet arrival rate.

provement in throughput is again due to the checking of channel status before decrementing the backoff counter value by an STA. However, at low packet arrival rate and with less number of STAs present in the network, performance of the proposed model is similar to Pham's model. This is because of the reason that the number of competing STAs is small and also the packet arrival rate is less, and hence channel status may be idle for most of the time. We also find that maximum throughput value gets shifted to a higher value with increase in packet arrival rate. This is obvious as there are small number of competing STAs (hence less probability of packet collisions) with a higher packet arrival rate.

From Figure 5 we observe that probability of packet collisions increases with increase in number of STAs in both the models. Due to checking of channel status by an STA before decrementing its backoff counter value, prob-

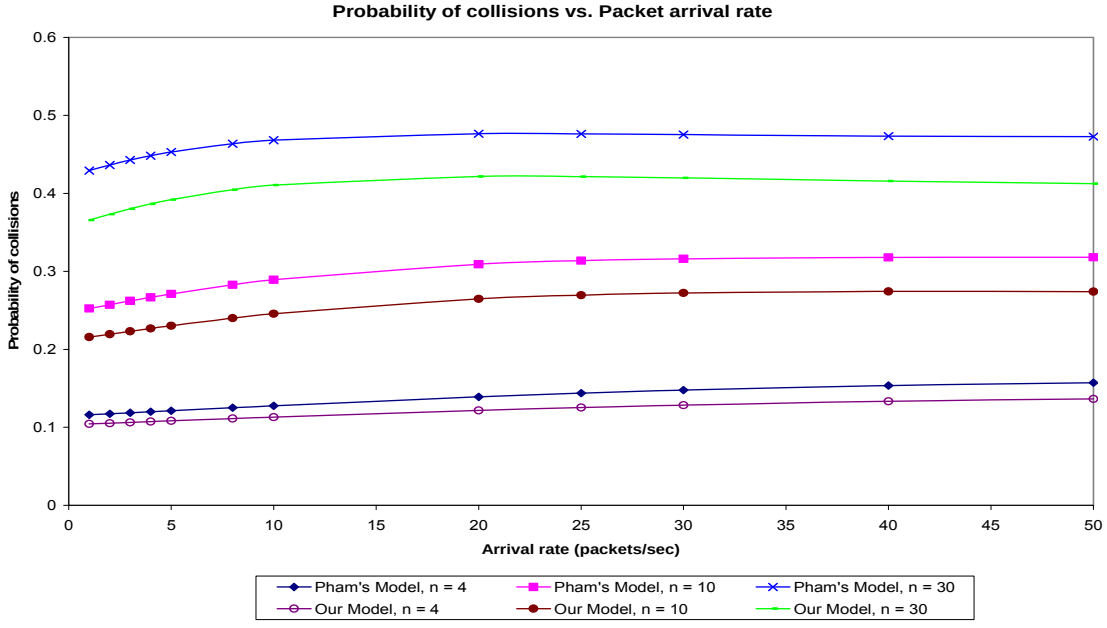


Figure 5. Probability of packet collisions vs. packet arrival rate with varying number of STAs.

ability of collision is small in the proposed model compared to Pham's model. Due to this we observe higher throughput with our model compared to Pham's model (Figures 3 and 4). Further, we find that probability of packet collision increases slowly for smaller packet arrival rate and then remains almost constant in both the models.

3. Performance Analysis under Slow Rayleigh Fading Channel

The Markov model developed for perfect channel case (Figure 1) can be easily extended to capture the behavior of IEEE 802.11 DCF under slow Rayleigh fading channels. However, there are certain differences that must be taken into account. Under the perfect channel assumption, the packet is not received successfully by an STA only when it is destroyed by collision. In a wireless environment, signal between mobile STAs undergoes deep fades [16] due to movement of STAs. The radio link between moving STAs is termed as Rayleigh channel. The signal fades result in packet drops due to low signal-to-noise ratio (SNR). In [14], channel "uptime" and "down time" are defined as two states of the channel. Channel "uptime" is defined as the duration when the received signal power is above a given threshold. Channel "downtime" is defined as the duration when the received signal power is below a given threshold. This can be modeled using a two-state Markov model. Readers are requested to refer [14] for a detailed discussion on Markov model for Rayleigh channels.

However, under the Rayleigh fading channel assumption, packets can be dropped if there is a collision or the channel is "down". Therefore, probability of packet not being received successfully (p) must take into account the above-mentioned causes. By taking this into consideration, it is possible to use the Markov model (Figure 1) to analyze the performance of IEEE 802.11 DCF under the Rayleigh fading channel also.

Under the Rayleigh channel assumption, a packet can be destroyed either by collision or when the channel is "down", that is,

$$p = 1-(1-\tau)^{n-1} + \pi_0(1-\tau)^{n-1} \quad \textbf{(29)}$$

where $\pi_0 = 1-e^{-\varepsilon^2}$ which represents the steady state probability of channel being "down" and ε represents the ratio between the power threshold and the root mean square (rms) value of received power. We can rewrite (29) and obtain transmission probability as,

$$\tau = 1-\left(\frac{1-p}{1-\pi_0}\right)^{1/(n-1)} \quad \textbf{(30)}$$

Using (24) and (30), we can easily obtain p and hence τ for a slow Rayleigh fading channel.

Having obtained the values of τ and p for a Rayleigh channel, the results in Section 2, (i.e., Equations (27) and (28)) can still be applied for the performance analysis of DCF. Next, we present the variation of channel throughput with varying packet arrival rate for a slow Rayleigh fading channel.

From Figure 6 it is evident that throughput increases linearly with increase in packet arrival rate before reaching a maximum value and then saturates even under Rayleigh fading channel. This is because the packet arrival rate is more than the packet processing rate of the system. As expected, saturation throughput decreases further under Rayleigh fading channel conditions as compared to a perfect channel assumption.

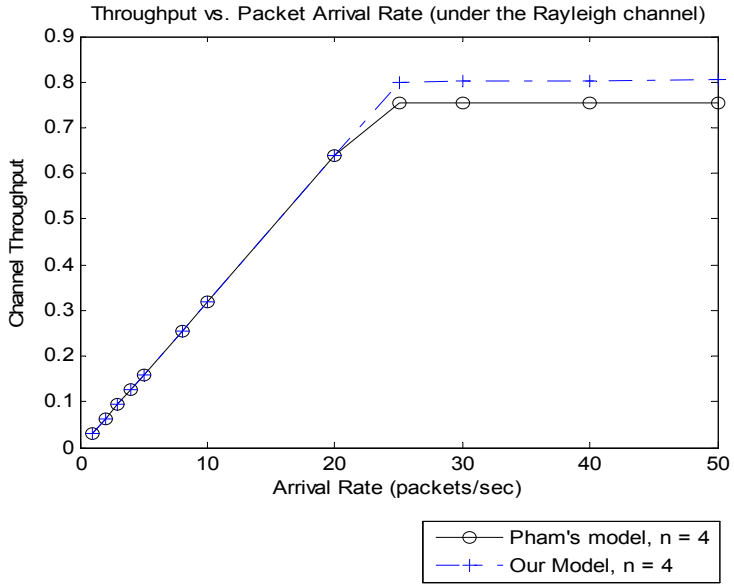


Figure 6. Channel throughput with varying packet arrival rate.

4. Conclusions

Markov Models proposed in the literature for IEEE 802.11 DCF do not comply with the 802.11 standard. In this paper, we have proposed an improved analytical model for DCF under finite load by closely following the specifications in 802.11 standard. Our analysis shown that our Markov model gives better insight into the operation of DCF in terms of channel throughput with varying packet arrival rate and number of STAs in a network under perfect and slow Rayleigh fading channels compared with Pham's model. Though we have shown our analysis for basic access mechanism of DCF, it is easy to extend our analysis for RTS/CTS-based access mechanism as well.

5. Acknowledgments

The first author expresses his sincere thanks to Prof. S.C. Laroiya, Director, National Institute of Technical Teachers' Training and Research, Chandigarh, India for his constant support and encouragement. We express our sincere thanks to anonymous reviewers for their valuable comments which improved the quality of the paper.

6. References

[1] "Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," IEEE Standard, 2007 Edition.

[2] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," IEEE J-SAC, Vol. 18, No. 3, pp. 535−547, March 2000.

[3] O. Tickoo and B. Sikdar, "A queue model for finite load IEEE 802.11 random access MAC," Proceedings of ICC 2004, Vol. 1, pp. 175−179, June 20−24, 2004.

[4] Y. S. Liaw, A. Dadej, and A. Jayasuriya, "Performance analysis of IEEE 802.11 DCF under limited load," Proceedings of IEEE 2005 Asia-Pacific Conference on Communications, Perth, Western Australia, pp. 759− 763, October 3−5, 2005.

[5] R. K. Challa, S. Chakrabarti, and D. Datta, "Modeling of IEEE 802.11 DCF for transient state conditions," Journal of Networks, Vol. 2, No. 4, pp. 14−19, August 2007.

[6] T.-S. Ho and K.-C. Chen, "Performance analysis of IEEE 802.11 CSMA/CA medium access control protocol," Proceedings of 7th IEEE International Symposium on PIMRC 1996, Vol. 2, pp. 407−411, October 15−18, 1996.

[7] H. S. Chhaya and S. Gupta, "Performance modeling of asynchronous data transfer methods of IEEE 802.11 MAC protocol," Wireless Networks, Vol. 3, pp. 217−234, August 1997.

[8] B. P. Crow, "Performance evaluation of the IEEE 802.11 wireless local area networking protocol," Master's thesis, Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ, 1996.

[9] R. K. Challa, S. Chakrabarti, and D. Datta, "A modified backoff algorithm for IEEE 802.11 DCF based MAC protocol in a mobile ad hoc network," Proceedings of IEEE TENCON 2004, Vol. B. 2, pp. 664−667, November 21−24, 2004.

[10] Z. H. Fu, P. Zerfos, K. X. Xu, H. Y. Luo, S. W. Lu, L. X. Zhang, and M. Gerla, "On TCP performance in multihop wireless networks," UCLA, WiNG Technical Report, 2002.

[11] F. Daneshgaran, M. Laddomada, F. Mesiti, and M. Mondin, "Unsaturated throughput analysis of IEEE 802.11 in the presence of non ideal transmission channel and capture effects," IEEE Transactions on Wireless Communications, Vol. 7, No. 4, pp. 1276−1286, 2008.

[12] F. Daneshgaran, M. Laddomada, F. Mesiti, and M. Mondin, "A model of the IEEE 802.11 DCF in the presence of non ideal transmission channel and capture effects," Proceedings of IEEE GLOBECOM'07, pp. 5112−5116, November 26−30, 2007.

[13] D. Malone, K. Duffy, and D. J. Leith, "Modeling the 802.11 distributed coordination function in non-saturated heterogeneous conditions," IEEE ACM Transactions on Networking, Vol. 15, No. 1, pp. 159−172, February 2007.

[14] P. P. Pham, "Comprehensive analysis of the IEEE 802.11," Mobile Networks and Applications, Vol. 10, No. 5, pp. 691−703, 2005.

[15] L. Kleinrock, "Queueing systems," Wiley, New York, 1975.

[16] T. S. Rappaport, "Wireless communication: Principles and practice," Prentice Hall, 1996.

Call for Papers

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.

Editors-in-Chief

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	Sensor networks
UWB technologies	Ad Hoc and mesh networks
Wave propagation and antenna design	Network protocol, QoS and congestion control
Signal processing and channel modeling	Efficient MAC and resource management protocols
Coding, detection and modulation	Simulation and optimization tools
3G and 4G technologies	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

Int. J. Communications, Network and System Sciences, **2009, 3, 169-247**
Published Online June 2009 in SciRes (http://www.SciRP.org/journal/ijcns/).

TABLE OF CONTENTS

Volume 2 Number 3 **June 2009**

ISSN: 1913-3715
9771913371005 07