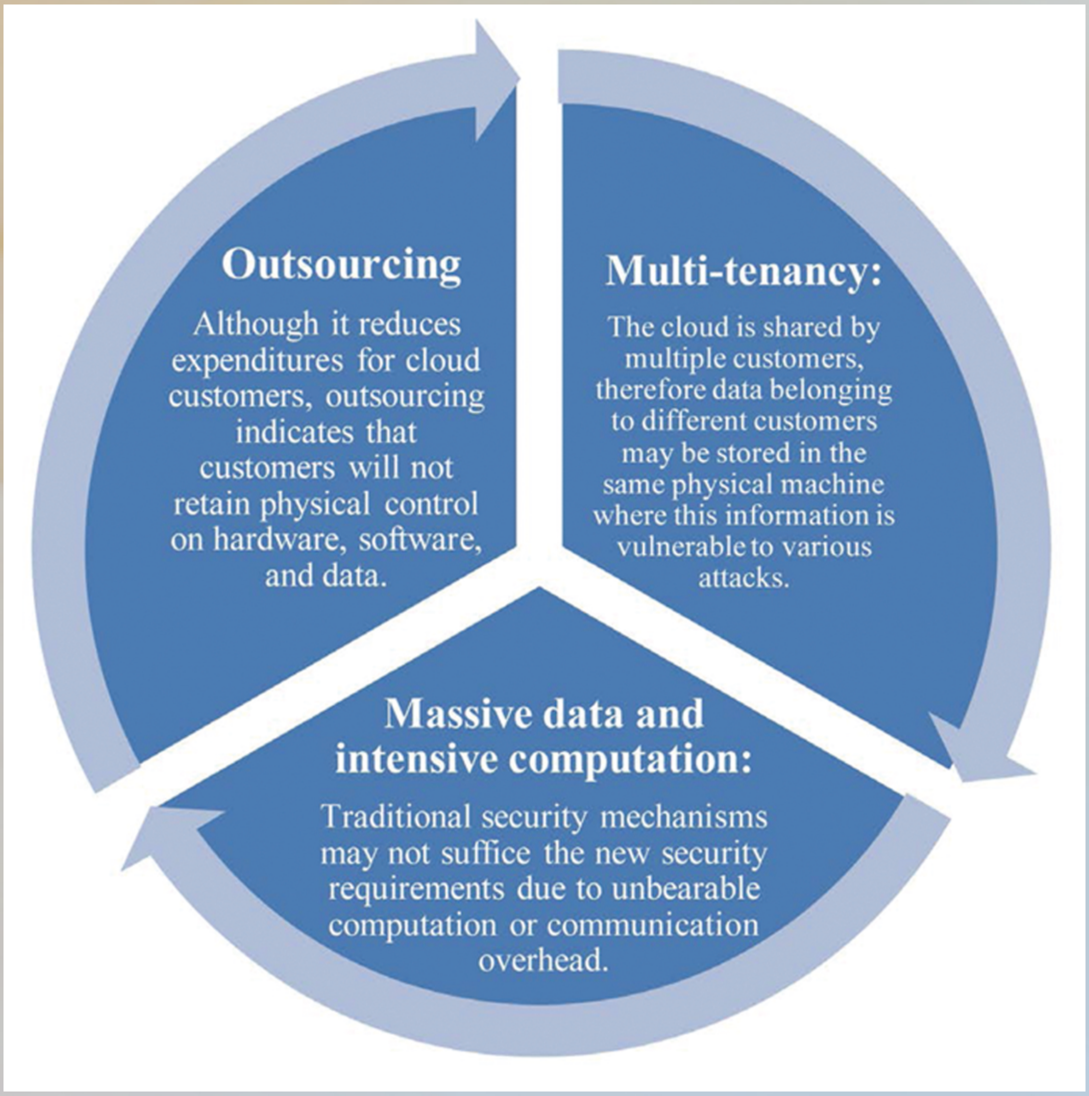
ISSN: 1913-3715 Volume 7, Number 12, December 2014

Scientific Research

International Journal of Communications, Network and System Sciences


Outsourcing
Although it reduces expenditures for cloud customers, outsourcing indicates that customers will not retain physical control on hardware, software, and data.
Multi-tenancy:
The cloud is shared by multiple customers, therefore data belonging to different customers may be stored in the same physical machine where this information is vulnerable to various attacks.
Massive data and intensive computation:
Traditional security mechanisms may not suffice the new security requirements due to unbearable computation or communication overhead.

ISSN: 1913-3715

9 771913 371006 12

www.scirp.org/journal/ijcns

JOURNAL EDITORIAL BOARD

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

Int. J. Communications, Network and System Sciences, 2014, 7, 497-535
Published Online December 2014 in SciRes. http://www.scirp.org/journal/ijcns

Table of Contents

Volume 7 Number 12 December 2014

The figure on the front cover is from the article published in Int. J. Communications, Network and System Sciences, 2014, Vol. 7, No. 12, pp. 529-535 by Patrick Mosca, *et al.*

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., USA.

Subscription rates:

Print: $89 per issue.
To subscribe, please contact Journals Subscriptions Department, E-mail: sub@scirp.org

SERVICES

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

Reprints (minimum quantity 100 copies)
Reprints Co-ordinator, Scientific Research Publishing, Inc., USA.
E-mail: sub@scirp.org

COPYRIGHT

COPYRIGHT AND REUSE RIGHTS FOR THE FRONT MATTER OF THE JOURNAL:

COPYRIGHT FOR INDIVIDUAL PAPERS OF THE JOURNAL:

REUSE RIGHTS FOR INDIVIDUAL PAPERS:

Note: At SCIRP authors can choose between CC BY and CC BY-NC. Please consult each paper for its reuse rights.

DISCLAIMER OF LIABILITY

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

PRODUCTION INFORMATION

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

Int. J. Communications, Network and System Sciences, 2014, 7, 497-507
Published Online December 2014 in SciRes. http://www.scirp.org/journal/ijcns
http://dx.doi.org/10.4236/ijcns.2014.712050

Playing against Hedge

Miltiades E. Anagnostou[1], Maria A. Lambrou[2]

[1]School of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece
[2]Department of Shipping, Trade and Transport, University of the Aegean, Chios, Greece
Email: miltos@central.ntua.gr, mlambrou@aegean.gr

Received 10 October 2014; revised 20 November 2014; accepted 1 December 2014

Abstract

Hedge has been proposed as an adaptive scheme, which guides the player's hand in a multi-armed bandit full information game. Applications of this game exist in network path selection, load distribution, and network interdiction. We perform a worst case analysis of the Hedge algorithm by using an adversary, who will consistently select penalties so as to maximize the player's loss, assuming that the adversary's penalty budget is limited. We further explore the performance of binary penalties, and we prove that the optimum binary strategy for the adversary is to make greedy decisions.

Keywords

Hedge Algorithm, Adversary, Online Algorithm, Greedy Algorithm, Periodic Performance, Binary Penalties, Path Selection, Network Interdiction

1. Introduction

The problems of adaptive network path selection and load distribution have often been considered as games that are played simultaneously and independently by agents controlling flows in a network. A possible abstraction of these and other related problems is the bandit game. In the *multi-armed bandit* game [1] a player chooses one out of N strategies (or "machines" or "options" or "arms"). A loss or penalty (or a reward, which can be modeled as a negative loss) ℓ_i is assigned to each strategy i $(i=1,2,\cdots,N)$ after each round of the game.

An agent facing repeated selections will possibly try to exploit the so far accumulated experience. A popular algorithm that can guide the agent in each selection round is the *multiplicative updates* algorithm or *Hedge*. In this paper we calculate the worst possible performance of Hedge by using the adversarial technique, *i.e.* we investigate the behavior of an intelligent adversary, who tries to maximize the player's cumulative loss. In Section 1 we describe Hedge; in Section 2 we give a rigorous formulation of the adversary's problem; in Section 3 we give a recursive solution; and in Section 4 we present sample numerical results. Finally, in Section 5 we

How to cite this paper: Anagnostou, M.E. and Lambrou, M.A. (2014) Playing against Hedge. *Int. J. Communications, Network and System Sciences*, **7**, 497-507. http://dx.doi.org/10.4236/ijcns.2014.712050

explore binary adversarial strategies. Our main result is that the greedy adversarial strategy is optimal among binary strategies.

1.1. The Bandit Game

In a generalized bandit game the player is allowed to play mixed strategies, *i.e.* to assign a fraction p_i (such that $\sum_{i=1}^{N} p_i = 1$) of the total bet to option i, thereby getting a loss equal to $L = \sum_{i=1}^{N} p_i \times \ell_i$. Alternatively, p_i can be interpreted as a probability that the player assigns the bet on option i. In the "bandit" version only the total loss L is announced to the player, while in the "full information" version the penalty vector $(\ell_1, \ell_2, \cdots, \ell_N)$ is announced.

A game consists of T rounds; a superscript t marks the t th $(t = 0, \cdots, T-1)$ round. Apparently the player will try to minimize the total cumulative loss

$$\sum_{t=0}^{T-1} L^t = \sum_{t=0}^{T-1} \sum_{i=1}^{N} p_i^t \times \ell_i^t \tag{1}$$

by controlling the bet distribution, *i.e.* by properly selecting the variables p_i^t. We use the additional assumption that the loss budget is limited in each round by setting the constraint $\sum_{i=1}^{N} \ell_i^t = 1$. Clearly a player's goal is to minimize his or her total cumulative loss. An extremely lucky player, or a player with "inside information", would select the minimum penalty option in each round and would put all his or her bet on this option, thereby achieving a total loss equal to $\sum_{t=0}^{T-1} \min_i \ell_i^t$.

1.2. The Hedge Algorithm

Quite a few algorithmic solutions, which will guide the player's hand in the full information game, have appeared in the literature. Freund and Schapire have proposed the *Hedge* algorithm [2] for the full information game. Auer, Cesa-Bianchi, Freund and Schapire have proposed the *Exp*3 algorithm in [3]. Allenberg-Neeman and Neeman proposed a Hedge variant, the *GL* (Gain-Loss) algorithm, for the full information game with gains and losses [4]. Dani, Hayes, and Kakade have proposed the *GeometricHedge* algorithm in [5], and a modification was proposed by Bartlett, Dani *et al.* in [6]. Recently Cesa-Bianchi and Lugosi have proposed the ComBand algorithm for the bandit version [7]. A comparison can be found in [8].

Hedge maintains a vector $w^t = (w_1^t, w_2^t, \cdots, w_n^t)$ of weights, such that $w_i^t \geq 0$ ($t = 0, 1, \cdots, T-1$, and $i = 1, 2, \cdots, N$). In each round t Hedge chooses the bet allocation according to the normalized weight $p_i^t = w_i^t \Big/ \sum_{i=1}^{N} w_i^t$. When the opponent reveals the loss vector of this round, the next round weight w^{t+1} is determined so as to reflect the loss results, *i.e.* $w_i^{t+1} = w_i^t \beta^{\ell_i^t}$ for some fixed β, such that $0 \leq \beta \leq 1$.

In [9] Auer, Cesa-Bianchi, Freund and Schapire have proved that the expected Hedge performance and the expected performance of the best arm differ at most by $O\left(\sqrt{TN \ln N}\right)$. Freund and Schapire [2] have given a loss upper bound, which relates the total cumulative loss with the total loss of the best arm.

1.3. Competitive Analysis

The competitive analysis of an algorithm $\mathcal{A}$, which in this paper is Hedge, involves a comparison of $\mathcal{A}$'s performance with the performance of the optimal offline algorithm. In the bandit game the optimal offline algorithm, *i.e.* the optimal player's decisions given the sequence of all penalties in advance, is trivial. In a given round the player can just bet everything on the option with the lowest penalty.

According to S. Irani and A. Karlin (in Section 13.3.1 of [10]) a technique in finding bounds is to use an "adversary" who plays against $\mathcal{A}$ and concocts an input, which forces $\mathcal{A}$ to incur a high cost. Using an adversary is just an illustrative way of saying that we try to find the worst possible performance of an online algorithm. In our analysis the adversary tries to maximize Hedge's total loss by controling the penalty vector (under a limited budget).

1.4. Interpretations and Applications

In this section we offer some interpretations from the areas of 1) communication networks and 2) transportation. The general setting of course involves a number of options or arms, which must be selected by a player without any knowledge of the future.

Bandit models have been used in quite diverse decision making situations. In [11] He, Chen, Wand and Liu have used a bandit model for the maximization of the revenue of a search engine provider, who charges for advertisements on a per-click basis. They have subsequently defined the "armed bandit problem with shared information"; arms are partitioned in groups and loss information is shared only among players using arms of the same group. In [12] Park and Lee have used a multi-armed bandit model for lane selection in automated highways and autonomous vehicles traffic control.

1.4.1. Traffic Load Distribution

This first application example can take multiple interpretations, which always involve a selection in a competitive environment, in which competition is limited. It can be seen as 1) a path selection problem in networking, 2) a transport means (mode) choice or path selection problem, 3) a computational load distribution problem, which we mention in the end of this section. Firstly, we describe the problem in the context of networking.

Consider N similar independent paths (in the simplest case just N parallel links), which join a pair of nodes $\mathcal{A}$, $\mathcal{B}$. A traffic volume equal to Q is sent from $\mathcal{A}$ to $\mathcal{B}$ in consecutive time periods or rounds by a population of agents. Q is the same in each round, but the allocation of Q to paths, *i.e.* $\left(Q_1^t, Q_2^t, \cdots, Q_N^t\right)$ such that $\sum_{i=1}^{N} Q_i = Q$, is different in each round t. An agent A produces a constant amount of traffic equal to A, such that $q \ll Q$, in T consecutive rounds, and allocates a part equal to q_i $\left(\sum_{i=1}^{t} q_i = q\right)$ to the i th path in round t. The average delay (or cost) experienced by A's traffic in the t th round is proportional to $\sum_{i=1}^{N} Q_i^t q_i^t$, if we assume a linear delay (or cost) model. Linear models are used for simplicity in network analysis [13] and can be realistic if a network resource still operates in the linear region of the delay vs. load curve, e.g. when delay is calculated in a link, which operates not very close to capacity. Agent A aims at minimizing the total delay for its own traffic and may use Hedge to determine the quantities q_i^t in round t, assuming that A knows the performance of its own traffic in each path in the past time period. Note that the maximum delay in a round occurs if A puts the whole q in a single path together with the whole traffic of the competition, *i.e.* with Q; then A's average delay in this round equals Q. On the contrary, if Q is evenly distributed in all paths, A's allocation decision does not really matter, as the average will be equal to $\sum_i \left(q_i/q\right) \times \left(Q/N\right) = Q/N$. Of course the minimum delay in a round will occur if A puts the whole q in an empty path, thereby achieving a zero delay.

The above problem can also be formulated as a more general problem of distributing workload over a collection of parallel resources (e.g. distributing jobs to parallel processors). A. Blum and C. Burch have used the following motivating scenario in [14]: A process runs on some machine in an environment with N machines in total. The process may move to a different machine at the end of a time interval. The load ℓ_i^t, which will be found on a machine i at time round t is the penalty felt by the process.

1.4.2. Interdiction

Although an adversary is usually a "technical" (fictional) concept, which serves the worst case analysis of online algorithms, in some environments a real adversary, who intentionally tries to oppose a player, does exist. An example is the interdiction problem.

We present a version of the interdiction problem in a network security context. An attacker attacks N resources (e.g. launches a distributed denial of service attack on nodes, servers, etc., see [15]) by sending streams of harmful packets to resource i at a rate w_i (where $i = 1, \cdots, N$ and $\sum_i w_i$ is constant). A defender assigns a defense mechanism of intensity ℓ_i (e.g. a filter that is able to detect and avoid harmful packets with a probability proportional to ℓ_i) to resource i. At the end of a time interval T, e.g. one day, both the attacker and the defender revise the flows and the distribution of defense mechanisms to resources respectively, based on past performance.

Similar interpretations exist in transportation network environments, as in border and custom control, including illegal immigration control. An interdiction problem formulation can be used in a maritime transport security context: pirates attack the vessels traversing a maritime route. In [16] Vanek *et al*. assign the role of the player to the pirate. The pirate operates in rounds, starting and finishing in his home port. In each round he selects a sea area (arm) to sail to and search for possible victim vessels. A patrol force distributes the available escort resources to sea areas (arms), and pirate gains are inversely proportional to the strength of the defender's forces on this area. Naval forces reallocate their own resources to sea areas.

2. Problem Formulation

In this paper we aim at finding the worst case performance of Hedge. Effectively, we try to solve the following problem:

Problem 1. *Given a number of options* N, *an initial normalized weight vector* $\boldsymbol{w}=(w_1,w_2,\cdots,w_N)$, *and a Hedge parameter* β, *find the sequence* $\boldsymbol{\ell}^0$, $\boldsymbol{\ell}^1$, $\cdots$, $\boldsymbol{\ell}^{T-1}$ *that maximizes the player's total cumulative loss*

$$L_{H(\beta)}=\sum_{t=0}^{T-1}\sum_{i=1}^{N}p_i^t\ell_i^t \tag{2}$$

where $\boldsymbol{\ell}^t=(\ell_1^t,\cdots,\ell_N^t)$ is the penalty vector in round t $(t=0,1,\cdots,T-1)$, such that $\sum_{i=1}^{N}\ell_i^t=1$, and the t th round penalty weights p_i^t are updated according to

$$w_i^t=w_i^{t-1}\beta^{\ell_i^{t-1}}=w_i\beta^{\sum_{\tau=0}^{t-1}\ell_i^\tau},\quad p_i^t=\frac{w_i^t}{\sum_{i=1}^{N}w_i^t}\quad(t\ge1) \tag{3}$$

for $t=1,\cdots,T-1$ and $p_i^0=w_i$. □

Clearly the objective function (2) is a function of a) the N initial weights w_i, and b) the $N\times T$ variables ℓ_i^t, and c) β. Due to the normalization of both weights and penalties there are $(N-1)\times(T+1)+1$ independent variables in total. In the following we use $L^{T-1}(w_1,\cdots,w_N;\ell_1^0,\cdots,\ell_N^0,\cdots,\ell_1^{T-1},\cdots,\ell_N^{T-1})$ or $L^{T-1}(\boldsymbol{w};\boldsymbol{\ell}^0,\cdots,\boldsymbol{\ell}^{T-1})$ instead of $L_{H(\beta)}$ whenever it is necessary to refer to these variables.

3. Recursion

Assuming that a given round starts with weights $\boldsymbol{w}=(w_1,\cdots,w_N)$ and the adversary generates penalties $\boldsymbol{\ell}=(\ell_1,\cdots,\ell_N)$, the next round will will start with weights $\boldsymbol{W}(\boldsymbol{w},\boldsymbol{\ell})=(W_1(\boldsymbol{w},\boldsymbol{\ell}),\cdots,W_N(\boldsymbol{w},\boldsymbol{\ell}))$ where

$$W_i(\boldsymbol{w},\boldsymbol{\ell})=\frac{w_i\beta^{\ell_i}}{\sum_{j=1}^{N}w_j\beta^{\ell_j}}\quad(i=1,2,\cdots,N). \tag{4}$$

Then, the total loss of a T round game, which starts with weights $\boldsymbol{w}$, can be written as the sum of the losses of a single round game, which starts with weights $\boldsymbol{w}$, and a $T-1$ round game, which starts with weights $\boldsymbol{W}(\boldsymbol{w},\boldsymbol{\ell})=(W_1(\boldsymbol{w},\boldsymbol{\ell}),\cdots,W_N(\boldsymbol{w},\boldsymbol{\ell}))$, as follows:

$$L^{T-1}(\boldsymbol{w};\boldsymbol{\ell}^0,\boldsymbol{\ell}^1,\cdots,\boldsymbol{\ell}^{T-1})=L^0(\boldsymbol{w};\boldsymbol{\ell}^0)+L^{T-2}(\boldsymbol{W}(\boldsymbol{w},\boldsymbol{\ell}^0);\boldsymbol{\ell}^1,\cdots,\boldsymbol{\ell}^{T-1}). \tag{5}$$

Note that the term L^{T-2}, which expresses the contribution of the last T rounds, depends only on the updated weights provided by the initial round. Such a Markovian property can be generalized in the following sense: A T_1+T_2 round game can be seen as consisting of a T_1 round game g_1 followed by a T_2 round game g_2, whose initial weights are the final weights of g_1, and no more details about g_1 are passed to g_2. Assuming that the solution to Problem 1 is $L_{\max}^{T-1}(\boldsymbol{w})=\max_{\boldsymbol{\ell}^0,\cdots,\boldsymbol{\ell}^{T-1}}L^{T-1}(\boldsymbol{w};\boldsymbol{\ell}^0,\cdots,\boldsymbol{\ell}^{T-1})$ the following recursive formula for $L_{\max}^{T-1}(\boldsymbol{w})$ can be derived from (5):

$$L_{\max}^{T-1}(\boldsymbol{w}) = \max_{\boldsymbol{\ell}}\left[L^0(\boldsymbol{w};\boldsymbol{\ell}) + L_{\max}^{T-2}(\boldsymbol{W}(\boldsymbol{w};\boldsymbol{\ell}))\right] \tag{6}$$

where $\boldsymbol{\ell}^0 = \boldsymbol{\ell}$ is the penalty vector chosen by the adversary in the initial round.

The optimal penalties can be computed also recursively. Let $\boldsymbol{\lambda}^{T-1;t}(\boldsymbol{w}) = \left(\lambda_1^{T-1;t}(\boldsymbol{w}),\cdots,\lambda_N^{T-1;t}(\boldsymbol{w})\right)$, where $\lambda_i^{T-1;t}(\boldsymbol{w})$ denotes the i th optimal penalty of the i th option in the t th round of a T round game (starting with weights $\boldsymbol{w}$). The optimal penalty of the initial round $(t=0)$ is apparently equal to the value of $\boldsymbol{\ell}$, which optimizes (6). Therefore

$$\boldsymbol{\lambda}^{T-1;0}(\boldsymbol{w}) = \arg\max_{\boldsymbol{\ell}}\left[L^0(\boldsymbol{w};\boldsymbol{\ell}) + L_{\max}^{T-2}(\boldsymbol{W}(\boldsymbol{w};\boldsymbol{\ell}))\right]. \tag{7}$$

In all other rounds $t = 1,2,\cdots,T-1$ the optimal penalties are such that the total loss of the rest of the game is maximized, *i.e.* such that $L_{\max}^{T-2}\left(\boldsymbol{W}\left(\boldsymbol{w},\boldsymbol{\lambda}^{T-1;0}(\boldsymbol{w})\right)\right)$ is achieved. Since the total loss $L_{\max}^{T-2}(\boldsymbol{w})$ is achieved by using penalties $\boldsymbol{\lambda}^{T-2;t}(\boldsymbol{w})$, the total loss $L_{\max}^{T-2}\left(\boldsymbol{W}\left(\boldsymbol{w},\boldsymbol{\lambda}^{T-1;0}(\boldsymbol{w})\right)\right)$ is realized by using $\boldsymbol{\lambda}^{T-2;t}\left(\boldsymbol{W}\left(\boldsymbol{w},\boldsymbol{\lambda}^{T-1;0}(\boldsymbol{w})\right)\right)$ instead. Therefore

$$\boldsymbol{\lambda}^{T-1;t+1}(\boldsymbol{w}) = \boldsymbol{\lambda}^{T-2;t}\left(\boldsymbol{W}\left(\boldsymbol{w},\boldsymbol{\lambda}^{T-1;0}(\boldsymbol{w})\right)\right) \quad (t = 0,1,\cdots,T-2). \tag{8}$$

4. Two Option Games and Numerical Results

This section we exploit the recursive methodology, which has been presented in the previous section, in order to provide some numerical results for two option games. We compare these results with available bounds in the literature. We consider $N = 2$, *i.e.* two option games. We keep only the independent penalties ℓ_1^t in the extended notation and use the more compact version $L^{T-1}\left(w_1;\ell_1^0,\ell_1^1,\cdots,\ell_1^{T-1}\right)$. As an example, the loss of a single round game is given by

$$L^0(w;\ell) = w\ell + (1-w)(1-\ell). \tag{9}$$

Also, since the initial weights are $\boldsymbol{w} = (w,1-w)$, we simplify the maximum cumulative loss $L_{\max}^{T-1}(\boldsymbol{w})$ to $L_{\max}^{T-1}(w)$. Assuming losses $\ell_1^0 = \ell$ and $\ell_2^0 = 1-\ell$, the next round will will start with weights $W(w,\ell)$ and $1-W(w,\ell)$, where

$$W(w,\ell) = \frac{w\beta^\ell}{w\beta^\ell + (1-w)\beta^{1-\ell}}. \tag{10}$$

Then (6) is simplified to

$$L_{\max}^{T-1}(w) = \max_{\ell}\left[L^0(w;\ell) + L_{\max}^{T-2}(W(w,\ell))\right] \tag{11}$$

where $\ell^0 = \ell$ is the penalty chosen by the adversary for the first option in the initial round.

The iteration starts from $L_{\max}^0(w)$, *i.e.* the loss of a single round game. In such game the adversary controls a single penalty variable, as the loss is given by (9). Apparently the adversary will choose binary values, *i.e.* $\ell = \ell_1^0 = 1$ $\left(\ell_1^0 = 0\right)$ if $w = w_1 > 1/2$ $(w_1 < 1/2)$, and the maximum total loss is $L_{\max}^0(w) = \max\{w,1-w\}$, *i.e.*

$$L_{\max}^0(w) = \begin{cases} 1-w, & \text{if } \ 0 \le w \le \dfrac{1}{2}, \\ w, & \text{if } \ \dfrac{1}{2} \le w \le 1. \end{cases} \tag{12}$$

The graph of $L_{\max}^0(w)$ appears as the lowest V-shaped "curve" in **Figure 1**. The fact that the $L_{\max}^0(w)$ is a piecewise linear function of w with a breakpoint (*i.e.* a sudden change in its slope), creates even more breakpoints in $L_{\max}^1(w)$, $L_{\max}^2(w)$ and so on. Therefore, while it is possible to use the aforementioned recursion in

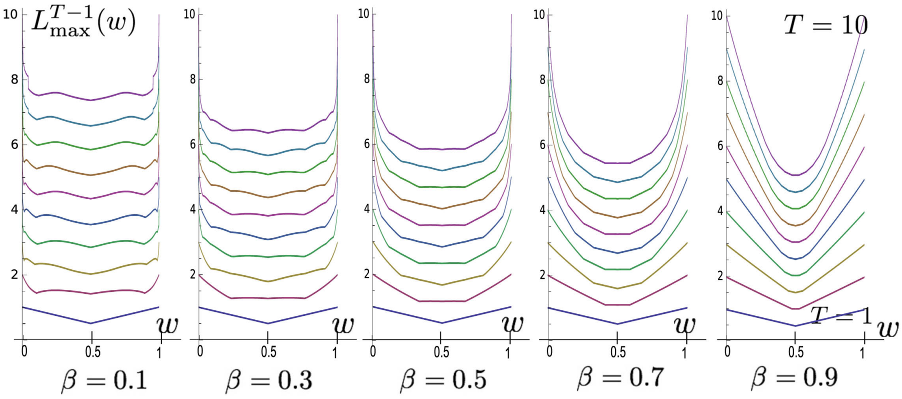


Figure 1. Plot of $L_{\max}^{T-1}(w)$ (maximum loss in a T round game) vs. w for $\beta=0.1,0.3,\cdots,0.9$ and $T=1,2,\cdots,10$.

order to find analytical expressions for the maximum total loss and the associated penalties, the analysis becomes quite complicated even for small values of the number of rounds T (*i.e.* in a $T+1$ round game). We omit this tedious analysis and present numerical results based on the recursive methodology given above.

Instead we have implemented a numerical computation based on (11). $L_{\max}^{T-1}(w)$ is approximated by $K+1$ samples in the interval $[0,1]$, *i.e.* by $L_{\max}^{T-1}(i\Delta w)$, where $i=0,1,\cdots,K$ and $\Delta w=1/K$. In the same way the functions $L^0(w;\ell)$ and $W(w,\ell)$ are represented by $(K+1)^2$ samples $L^0(m\Delta w;n\Delta\ell)$ and $W(m\Delta w,n\Delta\ell)$, where $\Delta w=\Delta\ell$. We have used $K=1000$. Initially we create $L_{\max}^0(i\Delta w)$ $(i=0,1,\cdots,K)$ by using (9). We use the result as input to (11) and create $(L_{\max}^1(i\Delta w))$. Then we use the already calculated L^0 and L^1 in (11) to calculate L^2, then L^0 and L^2 to calculate L^3, and so on. In **Figure 1** we show $L_{\max}^{T-1}(w)$ as a function of the initial weight $w_1=w$ in games with up to ten rounds $(T=1,\cdots,10)$ for different values of β. Observe that the shape of $L_{\max}^{T-1}(w)$ is more "interesting" for "unreasonably" small values of β.

The optimal penalties can be determined by using formulas (7) and (8) for $N=2$. In **Figure 2** we draw one of the curves of **Figure 1** together with the respective optimal penalties. The final round optimal penalty (*i.e.* $\lambda^{3;3}(w)$ in this example) is certain to be binary, since the adversary will assign $\ell_i^3=1$ to the option i with the greatest weight factor. However, the penalties $\lambda^{3;0}(w)$ and $\lambda^{3;1}(w)$ of the first two games are clearly non-binary.

5. Binary and Greedy Schemes

The penalty values in the first two rounds in the example of **Figure 2** prove that the adversary's optimal penalties are not necessarily binary. However, in this example β is "unnaturally" close to 0, as in practical Hedge implementations β is chosen close to 1; this choice achieves a more gradual adaptation to losses. Both experimental and analytical evidence show that the optimal penalties tend rapidly to binary values as β approaches 1. Effectively, it seems that results very close to optimum can be achieved by a "binary adversary", *i.e.* an adversary that will resort to binary values only.

On the other hand the optimal adversarial policy with binary penalties can be found exhaustively as

$$L_{\text{maxbin}}^{T-1}(\boldsymbol{w})=\max_{(\ell^0,\cdots,\ell^{T-1})\in S^T} L^{T-1}(\boldsymbol{w};\ell^0,\cdots,\ell^{T-1})$$

where S is a set of N binary vectors $(b_1,b_2,\cdots,b_N)$ such that $\sum_{i=1}^N b_i=1$, *i.e.* only one component equals 1. Apparently, the complexity of this calculation grows with N^T. However, in the following we show that the optimal binary adversary is in fact the "greedy adversary", The latter achieves binary optimality in linear time.

A "greedy adversary" is eager to punish the maximum weight option as much as possible in each round. Thus

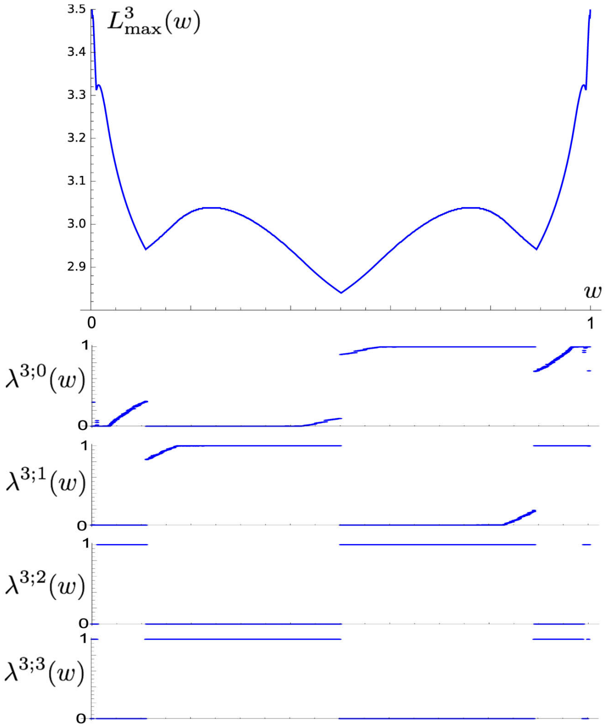


Figure 2. Plot of $L^3_{\max}(w)$ (maximum total loss of a 4 round game) vs. w for $\beta = 0.1$, together with the optimal penalties $\lambda^{3;t}$ $(t = 0,1,2,3)$.

the adversary will assign exactly one unit of penalty to the maximum current weight option, and zero penalties to all other options. Given a sufficient number of rounds (say t_0), it easy to see that the weights of an N option game are "equalized" so that any two weights p_i^t, p_j^t are such that $p_i^t / p_j^t < \beta$ for $t \ge t_0$. When equalization is achieved, a periodic phenomenon starts and the greedy penalties form a rotation scheme.

5.1. Greedy Behavior

We explore the greedy pattern in a two option game that can easily be generalized to N options. Assuming initial weights w_1, w_2 $(w_2 = 1 - w_1)$ such that $w_1 > 1/2 > w_2$, a greedy adversary will choose $\ell_1^0 = \ell_1^1 = \cdots = \ell_1^{t_0-1} = 1$, $\ell_1^{t_0} = 0$ iff $w_1\beta^{t_0-1} > w_2 > w_1\beta^{t_0}$, where $t_0 \ge 1$ (having assumed $w_1 > w_2$). At t_0 the weight of the second option becomes for the first time greater than the weight of the first option, and a loss equal to 1 is assigned to the second option. Therefore, in the next step $t_0 + 1$ the weights (before normalization) are $w_1\beta^{t_0}$ and $w_2\beta$, or equivalently $w_1\beta^{t_0-1}$ and w_2 for the second time. In the next round they become $w_1\beta^{t_0}$ and w_2 again, and in general they oscillate between these two pairs periodically. Therefore the total loss for $t \ge t_0$ in a pair of subsequent rounds is equal to

$$L_p = \frac{w_1\beta^{t_0}}{w_1\beta^{t_0} + w_2\beta} + \frac{w_2}{w_1\beta^{t_0} + w_2}. \tag{13}$$

The value of t_0 is determined by the initially assumed inequality, and since t_0 ought to be integer $t_0 = \lceil (\ln w_2 - \ln w_1)/\ln\beta \rceil$. The loss in the first t_0 steps $(t = 0,1,\cdots,t_0 - 1)$ is equal to

$$w_1 + \sum_{\tau=1}^{t_0-1} \frac{w_1\beta^\tau}{w_1\beta^\tau + w_2}.$$

Therefore, for an even positive integer $T-t_0$ the total loss in T steps is

$$L_{H(\beta)} = w_1 + \sum_{\tau=1}^{t_0-1} \frac{w_1\beta^\tau}{w_1\beta^\tau + w_2} + \frac{T-t_0}{2}\left[\frac{w_1\beta^{t_0}}{w_1\beta^{t_0} + w_2\beta} + \frac{w_2}{w_1\beta^{t_0} + w_2}\right].$$

In a game with more than two options it is straightforward to show that in the "steady" (periodic) state weights tend to become equal, *i.e.* almost equal to $1/N$, where N is the number of options. Consequently, the loss is given by $L_{H(\beta)} \approx T/N$ in a T round game.

5.2. Optimality of the Greedy Behavior

The following proposition provides a simple polynomial solution to the problem of finding the optimal binary adversary.

Proposition 1. *The greedy strategy is optimal for the adversary among all strategies with binary penalties.* □

Proof: Due to normalization of weights and penalties, in the proof we mention only option 1 weights and penalties. Assuming an initial weight ω and penalties $\ell_1^0, \ell_1^1, \cdots, \ell_1^{n-1}$ in the first n rounds, the weight, which emerges before the (*n* + 1)th round is $\omega\beta^L / \left(\omega\beta^L + 1 - \omega\right)$, where $L = \sum_{i=0}^{n-1} \ell_1^i$. Effectively, two options are available to the adversary in each step, either i) to assign a penalty equal to 1, which will produce an incremental loss equal to $\omega\beta^L / \left(\omega\beta^L + 1 - \omega\right)$, and will update the weight to $\omega\beta^{L+1} / \left(\omega\beta^{L+1} + 1 - \omega\right)$ or ii) to assign a zero penalty, which will produce a loss equal to $1 - \omega\beta^L / \left(\omega\beta^L + 1 - \omega\right)$ and an updated weight equal to $\omega\beta^{L-1} / \left(\omega\beta^{L-1} + 1 - \omega\right)$. Define $f(x) \equiv \omega\beta^x / \left(\omega\beta^x + 1 - \omega\right)$.

This looks like a new game, in which the adversary is the player. The player's status is determined by a real number x, and possible rewards are $f(x)$ and $1 - f(x)$. If the player opts for $f(x)$, this will bring him to a new status $x + \delta$. If he opts for $1 - f(x)$, this will bring him to $x - \delta$. In our case $\delta = 1$. Note also that $f(-\infty) = 1$, $f(+\infty) = 0$, and $f(0) = \omega$. Moreover, if ξ_0 is the root of $f(x) = 1/2$ (or $f(x) = 1 - f(x)$), then $f(x) \geq 1/2$ for $x \leq \xi_0$, and $f(x) \leq 1/2$ for $x \geq \xi_0$. It is easy to prove that there is an odd symmetry around $(\xi_0, 1/2)$, *i.e.* $f(\xi_0 + x) + f(\xi_0 - x) = 2f(\xi_0) = 1$, and $f(x)$ is concave in (∞, ξ_0), while it is convex in (ξ_0, ∞).

Assume that $\omega \geq 1/2$, then $f(0) = \omega \geq 1/2$, and $\xi_0 \geq 0$. If the current status of the player is x_1, and $x_1 < \xi_0$, the greedy behavior is to move $\lceil (x_1 - \xi_0)/\delta \rceil$ times to the right, which (unless T is too short) will bring the player to a point x_2 such that $x_2 \geq \xi_0$. If $x_2 > \xi_0$, then $1 - f(x_2) > \frac{1}{2} > f(x_2)$ and the greedy player must choose $1 - f(x_2)$ and move back to $x_2 - \delta < \xi_0$. Effectively, this starts an oscillation between $x_2 - \delta$ and x_2, which will last until the end of the game. In the following we prove that this behavior is optimal, in spite of the fact that profits around ξ_0 are low.

The main idea behind this sketch of proof is that a retreat (with consequent low profits $1 - f(x)$ is never a good investment for the future. Assume x_1 as the player's status, and T steps (rounds) remain until the end of the game, while $x_1 + T\delta < \xi_0$. The player executes M forward steps, *i.e.* $x_i = x_1 + i\delta$, $i = 0, 1, \cdots, M-1$, with rewards $f(x_i)$. Then, $M-1$ backward steps with gains $1 - f(x_i)$ are executed; consequently x_1 is reached again. In the rest of the game, *i.e.* until the T th step, greedy selections are made. This course of events is shown on curve (a) in **Figure 3**, where the dots mark the rewards achieved (and some dots have been vertically displaced by a small amount so as to be distinguishable from other dots at the same position). If greedy selections had been made all the way, the course of events would be as shown by curve (b).

If y_i describes the status of the adversary on the greedy curve (b) at the i th step and x_i the status on curve (a), then $f(x_i) = f(y_i)$ for $i = 0, \cdots, M-1$. Furthermore, $f(x_{3M+i}) = f(y_{M+i})$. Therefore the difference between the cumulative reward on curve (b) and curve (a) is

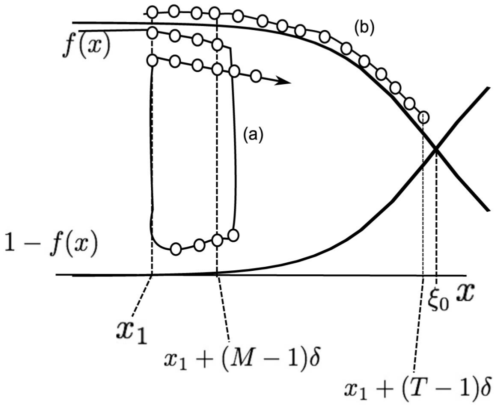


Figure 3. Sample paths of player behavior, which are used in the proof of Proposition 1.

$$\Delta R = \sum_{i=1}^{T}\left[f(y_i) - f(x_i)\right] = \sum_{i=T-2M+1}^{T} f(x_1+i\delta) - \left(\sum_{i=0}^{M-1} f(x_1+i\delta) + \sum_{i=1}^{M}\left[1-f(x_1+i\delta)\right]\right)$$

$$= \sum_{i=T-2M+1}^{T} f(x_1+i\delta) - \left[M + f(x_1) - f(x_1+M\delta)\right].$$

Effectively we need to show that $\Delta R \geq 0$. First, let us make some observations and explore other variations of $\Delta R \geq 0$. Note that ΔR, as given by (14), is positive if the cumulative reward from the back and forth movement (in the first $2M$ steps) is less than the reward in the last $2M$ steps. However, as T increases, the position of the last step approaches ξ_0 and it can be shown that the cumulative reward of the last $2M$ steps decreases. This property will be proved later, and it is due to the convexity and monotonicity properties of f. When T further increases, some of the very last $2M$ steps of the greedy behavior enter the phase of oscillation around ξ_0, and for T sufficiently large, all $2M$ belong to the oscillation phase. Note, however, that the oscillation phase rewards are those closer to 1/2, which is the lower limit of all greedy steps. If the greedy algorithm is to be optimal, even the $2M$ oscillatory steps should bring a cumulative reward greater than the original back and forth movement. On the other hand, if we prove this last inequality, this will also prove (14), whose last $2M$ steps bring more reward than the $2M$ oscillatory steps.

Let (ψ_1, ψ_2) be the pair of oscillation points around ξ_0, *i.e.* $\psi_1 = x_1 + \left\lfloor (\xi_0 - x_1)/\delta \right\rfloor \delta$ and $\psi_2 = \delta + \psi_1$. In the worst case, which has just been mentioned,

$$\Delta R = M\left(\frac{\omega\beta^{\psi_1}}{\omega\beta^{\psi_1}+1-\omega} + 1 - \frac{\omega\beta^{\psi_2}}{\omega\beta^{\psi_2}+1-\omega}\right) - \left[M + f(x_1) - f(x_1+M\delta)\right]$$

$$= M\left(\frac{\omega\beta^{\psi_1}}{\omega\beta^{\psi_1}+1-\omega} - \frac{\omega\beta^{\psi_2}}{\omega\beta^{\psi_2}+1-\omega}\right) - \left[f(x_1) - f(x_1+M\delta)\right].$$

However, $f(x_1) - f(x_1+M\delta)$ can be seen as the sum of M terms $f(x_1+i\delta) - f(x_1+(i+1)\delta)$, for $i = 0$, $M-1$. We shall further prove that each of these terms is smaller than the difference inside the big parentheses, *i.e.*

$$f(x_1+i\delta) - f(x_1+(i+1)\delta) \leq \frac{\omega\beta^{\psi_1}}{\omega\beta^{\psi_1}+1-\omega} - \frac{\omega\beta^{\psi_2}}{\omega\beta^{\psi_2}+1-\omega}. \tag{14}$$

This is a consequence of the following lemma:

Lemma 1. *For any concave function* $f(x)$ *the following inequality is true*:

$$f(x) - f(x+\Delta x) \leq f(x+\Delta x) - f(x+2\Delta x). \tag{15}$$

Inequality (15) holds because

$$\frac{f(x)-f(x+\Delta x)}{\Delta x} \ge f'(x+\Delta x) \ge \frac{f(x+\Delta x)-f(x+2\Delta x)}{\Delta x} \tag{16}$$

which is a consequence of the mean value theorem stating that there is a point ϕ_1 in $(x, x+\Delta x)$ such that $f'(\phi_1)=\left[f(x+\Delta x)-f(x)\right]/\Delta x$. Also, there is a point ϕ_2 in $(x+\Delta x, x+2\Delta x)$ such that $f'(\phi_2)=\left[f(x+2\Delta x)-f(x+\Delta x)\right]/\Delta x$. However, f is a concave function, and its derivative is non-increasing, therefore $\phi_1 \le x+\Delta x \le \phi_2$ implies $f'(\phi_1) \ge f'(x+\Delta x) \ge f'(\phi_2)$, which proves (16). In fact (15) can be easily generalized to any same length intervals, even overlapping ones, *i.e.* if $x_1 \le x_2$, then

$$f(x_1)-f(x_1+\Delta x) \le f(x_2)-f(x_2+\Delta x). \tag{17}$$

Due to (15) each successive equal length (*i.e.* Δx) interval produces an incremental reward $f(x)-f(x+\Delta x)$, which is smaller than the incremental reward of the next interval, and of all succeeding intervals, as long as f remains concave. Effectively, Lemma 1 proves that the incremental reward of the rightmost interval, which does not contain ξ_0, *i.e.* the interval $(\psi_1-\delta, \psi_1)$, is the highest among the rewards of all intervals of the same length, which begin to the left of $\psi_1-\delta$. Unfortunately, our aim was to prove (14), which would be secured if f remained concave in ψ_1, ψ_2, e.g. if $\psi_1=\xi_0-\delta$ and $\psi_2=\xi_0$. However this is not true, since at ξ_0 f turns from concave to convex. Fortunately, the term $f(\psi_1)-f(\psi_2)$, which covers the interval (ψ_1,ψ_2) can be seen as the sum of rewards related with the concave f in (ψ_1,ξ_0) and the concave $1-f$ in (ξ_0,ψ_2). Due to the odd symmetry around ξ_0, $f(\xi_0+(\psi_2-\xi_0))+f(\xi_0-(\psi_2-\xi_0))=2f(\xi_0)$, therefore $f(\psi_2)=2f(\xi_0)-f(2\xi_0-\psi_2)$, and $f(\psi_1)-f(\psi_2)=f(\psi_1)-\left[2f(\xi_0)-f(2\xi_0-\psi_2)\right]=\left[f(\psi_1)-f(\xi_0)\right]+\left[f(2\xi_0-\psi_2)-f(\xi_0)\right]$.

However, due to the concavity of f, $f(\psi_1)-f(\xi_0) \ge f(\psi_1-\delta)-f(\xi_0-\delta)$, and $f(2\xi_0-\psi_2)-f(\xi_0) \ge f(2\xi_0-\psi_2-(\xi_0-\psi_1))-f(\xi_0-(\xi_0-\psi_1))=f(\xi_0-\delta)-f(\psi_1)$. Therefore $f(\psi_1)-f(\psi_2) \ge \left[f(\psi_1-\delta)-f(\xi_0-\delta)\right]+\left[f(\xi_0-\delta)-f(\psi_1)\right]=f(\psi_1-\delta)-f(\psi_1)$.

This result states that the interval $(\psi_1,\psi_1+\delta)$, which contains ξ_0, provides higher Δf than the previous interval $(\psi_1-\delta,\psi_1)$, which in turn is higher than the Δf of any previous interval of the same length.

Therefore we have seen so far that a sequence of penalties, which begins at some $x<\xi_0$ and involves one fold, can be reduced to a sequence without any folds, and with improved total reward, as shown in **Figure 4**. In **Figure 4** a sequence of steps with a single fold, which starts at x_1 and ends at x_2, is shown together with the respective greedy sequence, which starts at x_1 and ends at $x_3=2M\delta+x_2$. If the sequence must extend after ξ_0, the additional steps are oscillation steps around ξ_0. The rest of this proof is just an application of this result, so that a sequence with an arbitrary number of folds can be reduced to an improved reward foldless sequence.

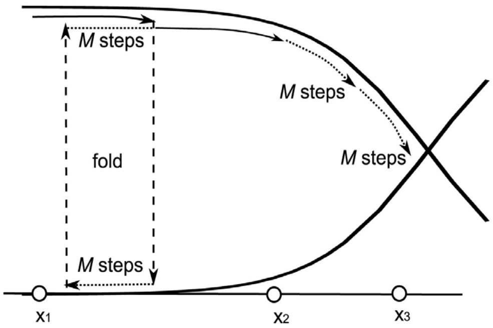


Figure 4. Reduction of a sequence of penalties, which contains a fold, to a sequence without folds and with improved total reward.

Suppose that the initial position of the game is at x_1, and that $x_1 \le \xi_0$ (otherwise reverse the initial probabilities ω, $1-\omega$). Suppose also that the initial sequence does not extend beyond ψ_2, *i.e.* it does not reach ξ_0 or it involves a number of oscillations around ξ_0. Then take the last fold and reduce it as mentioned, *i.e.* by replacing it with an equal number of greedy steps at the end of the current sequence. If these steps reach ξ_0, they are oscillation steps. Repeat the same step, until all folds have disappeared (oscillations do not count as folds). If the original sequence does extend beyond ξ_0, the approach is the same, but the reader should note that the application of this algorithm will finally reduce the part, which extends beyond ψ_2, to oscillations between ψ_1 and ψ_2.

6. Conclusion

We summarize the main results of this paper: An worst performance (adversarial) analysis of the Hedge algorithm has been presented, under the assumption of limited penalties per round. A recursive expression has been given for the evaluation of the maximum total cumulative loss; this expression can be exploited both numerically and analytically. However, binary penalty schemes provide an excellent approximation to the optimal scheme, and, remarkably, the greedy binary strategy has been proved optimal among binary schemes for the adversary.

References

[1] Robbins, H. (1952) Some Aspects of the Sequential Design of Experiments. *Bulletin of the American Mathematical Society*, **58**, 527-535. http://dx.doi.org/10.1090/S0002-9904-1952-09620-8

[2] Freund, Y. and Schapire, R.E. (1997) A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, **55**, 119-139. http://dx.doi.org/10.1006/jcss.1997.1504

[3] Auer, P., Cesa-Bianchi, N., Freund, Y. and Schapire, R.E. (2002) The Non-Stochastic Multi-Armed Bandit Problem. *SIAM Journal on Computing*, **32**, 48-77. http://dx.doi.org/10.1137/S0097539701398375

[4] Allenberg-Neeman, C. and Neeman, B. (2004) Full Information Game with Gains and Losses. *Algorithmic Learning Theory*: 15*th International Conference*, **3244**, 264-278.

[5] Dani, V., Hayes, T.P. and Kakade, S.M. (2008) The Price of Bandit Information for Online Optimization. In: Platt, J.C., Koller, D., Singer, Y. and Roweis, S., Eds., *Advances in Neural Information Processing Systems*, MIT Press, Cambridge, 345-352.

[6] Bartlett, P., Dani, V., Hayes, T., Kakade, S., Rakhlin, A. and Tewari, A. (2008) High-Probability Regret Bounds for Bandit Online Linear Optimization. *Proceedings of* 22*nd Annual Conference on Learning Theory* (COLT), Helsinki.

[7] Cesa-Bianchi, N. and Lugosi, G. (2012) Combinatorial Bandits. *Journal of Computer and System Sciences*, **78**, 1404-1422. http://dx.doi.org/10.1016/j.jcss.2012.01.001

[8] Uchiya, T., Nakamura, A. and Kudo, M. (2010) Algorithms for Adversarial Bandit Problems with Multiple Plays. In: Hutter, M., Stephan, F., Vovk, V. and Zeugmann, T., Eds., *Algorithmic Learning Theory*, Lecture Notes in Artificial Intelligence No. 6331, Springer, 375-389.

[9] Auer, P., Cesa-Bianchi, N., Freund, Y. and Schapire, R.E. (1995) Gambling in a Rigged Casino: The Adversarial Multi-Armed Bandit Problem. *Proceedings of* 36*th Annual Symposium on Foundations of Computer Science*, Milwaukee, 322-331.

[10] Hochbaum, D.S. (1995) Approximation Algorithms for NP-Hard Problems. PWS Publishing Company, Boston.

[11] He, D., Chen, W., Wang, L. and Liu, T.-Y. (2013) Online Learning for Auction Mechanism in Bandit Setting. *Decision Support Systems*, **56**, 379-386. http://dx.doi.org/10.1016/j.dss.2013.07.004

[12] Park, C. and Lee, J. (2012) Intelligent Traffic Control Based on Multi-Armed Bandit and Wireless Scheduling Techniques. *International Conference on Advances in Vehicular System, Technologies and Applications*, Venice, 23-27.

[13] Bertsekas, D.P. (1998) Network Optimization. Athena Scientific, Belmont.

[14] Blum, A. and Burch, C. (2000) On-Line Learning and the Metrical Task System Problem. *Machine Learning*, **39**, 35-88. http://dx.doi.org/10.1023/A:1007621832648

[15] Cole, S.J. and Lim, C. (2008) Algorithms for Network Interdiction and Fortification Games. *Springer Optimization and Its Applications*, **17**, 609-644. http://dx.doi.org/10.1007/978-0-387-77247-9_24

[16] Vaněk, O., Jakob, M. and Pěchouček, M. (2011) Using Agents to Improve International Maritime Transport Security. *IEEE Intelligent Systems*, **26**, 90-95. http://dx.doi.org/10.1109/MIS.2011.23

Int. J. Communications, Network and System Sciences, 2014, 7, 508-518
Published Online December 2014 in SciRes. http://www.scirp.org/journal/ijcns
http://dx.doi.org/10.4236/ijcns.2014.712051

Complexity Reduced MIMO Interleaved SC-FDMA Receiver with Iterative Detection

Masaki Tsukamoto, Yasunori Iwanami

Department of Computer Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology, Nagoya, Japan
Email: 25417574@stn.nitech.ac.jp, iwanami@nitech.ac.jp

Received 11 October 2014; revised 21 November 2014; accepted 1 December 2014

Abstract

In this paper, we propose the receiver structure for Multiple Input Multiple Output (MIMO) Interleaved Single Carrier-Frequency Division Multiple Access (SC-FDMA) where the Frequency Domain Equalization (FDE) is firstly done for obtaining the tentative decision results and secondly using them the Inter-Symbol Interference (ISI) is cancelled by ISI canceller and then the Maximum Likelihood Detection (MLD) is used for separating the spatially multiplexed signals. Furthermore the output from MLD is fed back to ISI canceller repeatedly. In order to reduce the complexity, we replace the MLD by QR Decomposition with M-Algorithm (QRD-M) or Sphere Decoding (SD). Moreover, we add the soft output function to SD using Repeated Tree Search (RTS) algorithm to generate soft replica for ISI cancellation. We also refer to the Single Tree Search (STS) algorithm to further reduce the complexity of RTS. By examining the BER characteristics and the complexity reduction through computer simulations, we have verified the effectiveness of proposed receiver structure.

Keywords

Interleaved SC-FDMA, MLD, QRD-M, Sphere Decoding, RTS, STS, Iterative Detection, Complexity Reduction

1. Introduction

Recently MIMO transmission techniques with multiple transmit and receive antennas are widely used to achieve the spatially multiplexed transmission and to increase the transmission rate in wireless communications. For MIMO spatially multiplexed transmission, MLD is known as the optimum signal separation method at the receiver side, which attains the minimum BER. However, when the number of transmit antenna and the modulation levels are increased, MLD needs very high computational complexity and the reduction of complexity be-

How to cite this paper: Tsukamoto, M. and Iwanami, Y. (2014) Complexity Reduced MIMO Interleaved SC-FDMA Receiver with Iterative Detection. *Int. J. Communications, Network and System Sciences*, **7**, 508-518.
http://dx.doi.org/10.4236/ijcns.2014.712051

comes a problem [1]-[3]. The SC-FDMA is used as the uplink wireless scheme in LTE (Long Term Evolution). The feature of its low Peak to Average Power Ratio (PAPR) characteristics decreases the burden of amplifier linearity in User Equipment (UE) and the SC-DFMA is more suitable to uplink transmission than Orthogonal Frequency Division Multiplexing (OFDM) [4]. Moreover by employing the interleaved SC-FDMA where the subcarriers for each UE are deployed like a comb tooth, the PAPR of SC-FDMA is further reduced and the frequency diversity effect becomes large. We have already proposed the MIMO SC-FDE and MIMO Interleaved SC-FDMA receivers with iterative detection where the receive signal is firstly detected by FDE, *i.e.*, MMSE nulling, to obtain the tentative decision results and secondly the ISI cancellation from the receive signal using the tentative decision results is done followed by the MLD for separating spatially multiplexed signals [5]. However, as the complexity of MLD increases as the power of modulation levels to the number of transmit antenna, the complexity reduction of MLD becomes an important issue. For reducing the complexity of MLD, QRD-M is proposed [6], but it is a quasi-Maximum Likelihood (ML) method and could not obtain the ML solution although the complexity is greatly reduced. On the other hand, SD [1] [2] can obtain the ML solution like MLD with reduced complexity. In this paper, we propose the novel receiver structure in which the MLD is replaced by QRD-M or SD to reduce the complexity of MLD [5]. In addition, by using RTS algorithm [7], we add the bit LLR output function on SD, which enables the proposed SD receiver to cancel the ISI with the soft replica resulting in more accurate ISI cancellation. Moreover we have replaced the RTS by the STS [8] algorithm to further reduce the complexity of RTS. Through computer simulations, we have examined the BER characteristics and the complexity reduction effect of proposed MIMO interleaved SC-FDMA iterative receiver with ISI canceller and MLD, QRD-M, SD with RTS or STS. Consequently we verify that the receiver structure using STS mostly improves the BER and the complexity.

2. MIMO Interleaved SC-FDMA Receiver

2.1. Proposed Transmitter and Receiver Structure

In **Figure 1**, the block diagram of transmitter and receiver for the uplink is shown. At the transmitter of each UE, the Quadrature Amplitude Modulation (QAM) modulated signal is Fast Fourier Transform (FFT) transformed with *N*-points and converted to the frequency domain. The FFT points are then mapped to the interleaved frequency points like a comb tooth. After that, the frequency points are Inverse FFT (IFFT) transformed with *M* points ($M = 4N$ in case of 4 UE's for example) to obtain the time domain signal. The Cyclic Prefix (CP) is inserted and the signal is transmitted to the channel. At the receiver in Base Station (BS), after removing the CP, the FDE, *i.e.*, MMSE nulling, is firstly done. The receive signal is then FFT converted with M points and the frequency domain signal is obtained. The frequency points are de-mapped to each user subcarrier arrangement and the subcarriers are multiplied by the MMSE weight $\boldsymbol{G}_u(n)$ at frequency point *n*, *i.e.*

$$\boldsymbol{G}_u(n) = \boldsymbol{H}_u(n)^H \left\{ \boldsymbol{H}_u(n) \boldsymbol{H}_u(n)^H + n_T \sigma^2 \boldsymbol{I}_{n_T} \right\}^{-1} \tag{1}$$

where $\boldsymbol{H}_u(n)$ denotes the MIMO channel matrix at the frequency point n assigned to the user u $(u = 1 \sim U)$, σ^2 the variance of noise at each frequency point, $\boldsymbol{I}_{n_T}$ the identity matrix with the size n_T which is the number of transmit antenna. After that, the frequency points are IFFT transformed with N points and the time domain signal to be detected is obtained as $\hat{\boldsymbol{x}}_u$.

$$\hat{\boldsymbol{x}}_u = \left[\hat{\boldsymbol{x}}_{u1}, \cdots, \hat{\boldsymbol{x}}_{uk}, \cdots, \hat{\boldsymbol{x}}_{uN} \right] \tag{2}$$

We call $\hat{\boldsymbol{x}}_u$ as the tentative decision through FDE. Using $\hat{\boldsymbol{x}}_u$, the receive signal replica due to ISI caused by the transmit signals other than the signal at time k to be detected, is generated and is subtracted from the receive signal. Using tentative decision result of (2) and by letting the transmit signal of user u at time k being 0, the tentative decision result for ISI cancellation is obtained as (3).

$$\hat{\boldsymbol{x}}_{u\backslash k} = \left[\hat{\boldsymbol{x}}_{u1}, \cdots, \hat{\boldsymbol{x}}_{u(k-1)}, 0, \hat{\boldsymbol{x}}_{u(k+1)}, \cdots, \hat{\boldsymbol{x}}_{uN} \right] = \begin{bmatrix} \hat{x}_{u1,1} & \cdots & \hat{x}_{u(k-1),1} & 0 & \hat{x}_{u(k+1),1} & \cdots & \hat{x}_{uN,1} \\ \hat{x}_{u1,2} & \cdots & \hat{x}_{u(k-1),2} & 0 & \hat{x}_{u(k+1),2} & \cdots & \hat{x}_{uN,2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hat{x}_{u1,n_T} & \cdots & \hat{x}_{u(k-1),n_T} & 0 & \hat{x}_{u(k+1),n_T} & \cdots & \hat{x}_{uN,n_T} \end{bmatrix}, \tag{3}$$

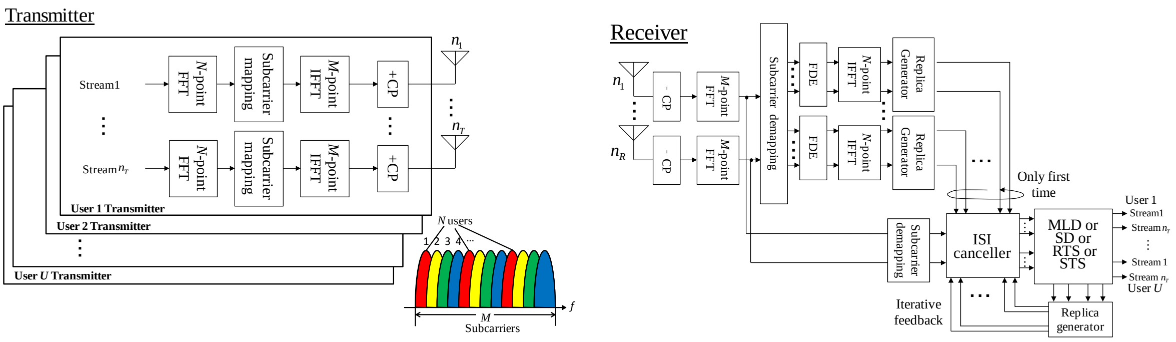


Figure 1. Block diagram of MIMO SC-FDMA transmitter and the proposed receiver structure with iterative feedback.

Next, $\hat{\boldsymbol{x}}_{u\backslash k}$ in (3) is transformed to frequency domain signal of $\hat{\boldsymbol{X}}_{u\backslash k}$ using FFT with N points. The frequency domain ISI replica is made through multiplying $\hat{\boldsymbol{X}}_{u\backslash k}(n)$ by the channel matrix $\boldsymbol{H}_u(n)$ and the ISI replica $\boldsymbol{H}_u(n)\hat{\boldsymbol{X}}_{u\backslash k}(n)$ is subtracted from the receive signal in frequency domain. Accordingly, the ISI components due to the transmit signals other than time k are removed. If the ISI cancellation is perfect, then the condition where only the transmit signals at time k from n_T transmit antennas are transmitted to the receiver is achieved. The ISI cancelled receive signal $\boldsymbol{Z}_{u/k}(n)$ in frequency domain is expressed as

$$\begin{aligned}\boldsymbol{Z}_{u/k}(n) &= \boldsymbol{Y}_u(n) - \boldsymbol{H}_u(n)\hat{\boldsymbol{X}}_{u\backslash k}(n)\\ &= \begin{bmatrix} Y_{u1}(n)\\ Y_{u2}(n)\\ \vdots\\ Y_{un_R}(n)\end{bmatrix} - \begin{bmatrix} H_{u,11}(n) & H_{u,12}(n) & \cdots & H_{u,1n_T}(n)\\ H_{u,21}(n) & H_{u,12}(n) & \cdots & H_{u,2n_T}(n)\\ \vdots & \vdots & \ddots & \vdots\\ H_{u,n_R1}(n) & H_{u,n_R2}(n) & \cdots & H_{u,n_Rn_T}(n)\end{bmatrix}\begin{bmatrix}\hat{X}_{u\backslash k,1}(n)\\ \hat{X}_{u\backslash k,2}(n)\\ \vdots\\ \hat{X}_{u\backslash k,n_T}(n)\end{bmatrix},\end{aligned} \tag{4}$$

where $\boldsymbol{Y}_u(n)$ is the receive signal of user u in frequency domain. Next, for $\boldsymbol{Z}_{u/k}(n)$ in (4), the signal separation of spatially multiplexed transmission is done using MLD. The total number of candidates of receive replica for MLD is K^{n_T}, where K is the modulation levels. The candidate signal for MLD in time domain $\hat{\boldsymbol{x}}_{u/k}$ is obtained by letting the transmit signals all 0 except for at time k to be detected.

$$\hat{\boldsymbol{x}}_{u/k} = \begin{bmatrix}0 & \cdots & 0 & \hat{\boldsymbol{x}}_{uk} & 0 & \cdots & 0\end{bmatrix} = \begin{bmatrix} 0 & \cdots & 0 & \hat{x}_{uk,1} & 0 & \cdots & 0\\ 0 & \cdots & 0 & \hat{x}_{uk,2} & 0 & \cdots & 0\\ & & & \vdots & & \cdots & \\ 0 & \cdots & 0 & \hat{x}_{uk,n_T} & 0 & \cdots & 0\end{bmatrix} \tag{5}$$

where the matrix size is $n_T \times N$. $\hat{\boldsymbol{x}}_{u/k}$ in (5) is then FFT transformed with N points resulting in $\hat{\boldsymbol{X}}_{u/k}(n)$. $\hat{\boldsymbol{X}}_{u/k}(n)$ is then multiplied by the channel matrix of $\boldsymbol{H}_u(n)$ which is assigned for user *u* at frequency point *n* and the candidate receive replica for MLD is obtained as $\boldsymbol{H}_u(n)\hat{\boldsymbol{X}}_{u/k}(n)$. Then the squared distance between the ISI cancelled receive signal and the candidate MLD replica in frequency domain is calculated as

$$\sum_{n=1}^{N}\left\|\boldsymbol{Z}_{u/k}(n) - \boldsymbol{H}_u(n)\hat{\boldsymbol{X}}_{u/k}(n)\right\|^2 \tag{6}$$

where $\|*\|$ denotes the Euclidian norm. (6) is minimized over the total K^{n_T} MLD candidates and the MLD output of $\hat{\boldsymbol{x}}_{uk}$ in (5) which minimizes (6) is obtained. The tentative decision result $\hat{\boldsymbol{x}}_{uk}$ in (2) is then replaced by the obtained $\hat{\boldsymbol{x}}_{uk}$ and the procedure proceeds from time k to time $k+1$ where the initial value of k is 1. This ISI canceller with MLD procedure is sequentially done from time 1 to N. Accordingly the residual ISI components in tentative FDE decision results are more precisely removed and the spatially multiplexed signals are more accurately separated. After the processing for one FFT block is done, the obtained decision results for one block are regarded as the evolved tentative decision results. Then the MLD outputs are fed back to the ISI canceller at each FFT block and this feedback is iteratively done to lower the final BER.

2.2. Complexity Reduction of MLD by QRD-M or SD

The number of candidate replicas in MLD increases exponentially as K^{n_T}. As the complexity reduction method of MLD, we illustrate the method utilizing the tree search of MLD with QR decomposition [6]. The receive signal vector is written as

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n} \tag{7}$$

where $\boldsymbol{y}$ is the receive signal vector with $n_R \times 1$, $\boldsymbol{H}$ the frequency flat channel matrix with $n_R \times n_T$, $\boldsymbol{x}$ the transmit signal vector with $n_T \times 1$ and $\boldsymbol{n}$ the receive noise vector with $n_R \times 1$. Using the QR decomposition, the channel matrix is decomposed into $\boldsymbol{H} = \boldsymbol{Q}\boldsymbol{R}$, where $\boldsymbol{Q}$ is the unitary matrix and $\boldsymbol{R}$ is the upper triangular matrix. By multiplying the Hermitian transpose $\boldsymbol{Q}^H$ by $\boldsymbol{y}$ from the left hand side, we obtain

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{Q}\boldsymbol{R}\boldsymbol{x} + \boldsymbol{n} \\ \boldsymbol{Q}^H\boldsymbol{y} &= \boldsymbol{R}\boldsymbol{x} + \boldsymbol{Q}^H\boldsymbol{n} \\ \tilde{\boldsymbol{y}} &= \boldsymbol{R}\boldsymbol{x} + \tilde{\boldsymbol{n}} \end{aligned} \tag{8}$$

where $\tilde{\boldsymbol{y}} = \boldsymbol{Q}^H\boldsymbol{y}$ and $\tilde{\boldsymbol{n}} = \boldsymbol{Q}^H\boldsymbol{n}$. The MLD criterion is then expressed as

$$\hat{\boldsymbol{x}} = \underset{\hat{\boldsymbol{x}}}{\operatorname{argmin}} \left\| \boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}} \right\|^2 = \underset{\hat{\boldsymbol{x}}}{\operatorname{argmin}} \left\| \boldsymbol{y} - \boldsymbol{Q}\boldsymbol{R}\hat{\boldsymbol{x}} \right\|^2 = \underset{\hat{\boldsymbol{x}}}{\operatorname{argmin}} \left\| \boldsymbol{Q}^H\boldsymbol{y} - \boldsymbol{Q}^H\boldsymbol{Q}\boldsymbol{R}\hat{\boldsymbol{x}} \right\|^2 = \underset{\hat{\boldsymbol{x}}}{\operatorname{argmin}} \left\| \tilde{\boldsymbol{y}} - \boldsymbol{R}\hat{\boldsymbol{x}} \right\|^2 \tag{9}$$

As $\boldsymbol{R}$ is the upper triangular matrix, the detection of transmit signal is considered as the tree search problem from $\hat{x}_{n_T}$ where $\hat{x}_{n_T}$ denotes the transmit signal candidate from antenna n_T. The tree structure is shown in **Figure 2** when $K = 2$ (BPSK) and $n_T = 4$, where the diverging number at each node and the depth of tree become $K = 2$ and $n_T = 4$ respectively. Equation (9) is also expressed in elements as

$$\begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \\ \hat{x}_4 \end{bmatrix} = \underset{\hat{x}_1,\hat{x}_2,\hat{x}_3,\hat{x}_4}{\operatorname{argmin}} \left\| \begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \tilde{y}_3 \\ \tilde{y}_4 \end{bmatrix} - \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ 0 & r_{22} & r_{23} & r_{24} \\ 0 & 0 & r_{33} & r_{34} \\ 0 & 0 & 0 & r_{44} \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \\ \hat{x}_4 \end{bmatrix} \right\|^2 \tag{10}$$

As the tree search method for **Figure 2** toward the width direction, M algorithm is widely known. At each step, the squared distance norm for every branch is calculated, and arbitral m survival paths with the least cumulative squared distance metric are retained. The complexity of M algorithm is constant when the value of m is determined and the QRD-M algorithm reduces the complexity of MLD very much, especially when $m = 1$. But it could not obtain the ML solution, *i.e.*, quasi-ML. On the other hand, the SD algorithm searches the tree of **Figure 2** toward the depth direction. The SD first determines the initial sphere radius $C = \left\| \tilde{\boldsymbol{y}} - \boldsymbol{R}\hat{\boldsymbol{x}} \right\|$ for some transmit candidate of $\hat{\boldsymbol{x}}$. Next SD searches the transmit signal vector which falls within the radius C toward the depth direction. When the cumulative distance metric exceeds the initial radius, then the subsequent

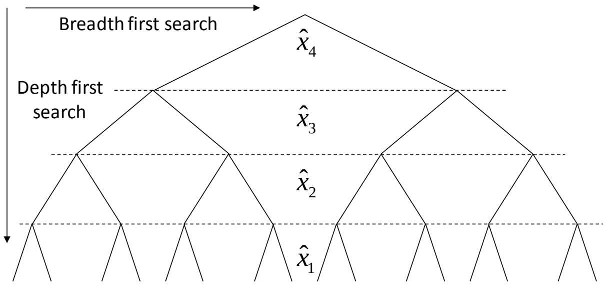


Figure 2. Tree structure of MLD when using QR decomposition.

search along the path is no more needed, thus the amount of calculation is saved. Therefore, when the initial sphere radius is small, the complexity reduction becomes more effective. In other words, the higher the E_b/N_0 and the smaller the initial sphere radius is, more effectively the complexity reduction is done. If the cumulative distance metric does not exceeds the initial radius till the bottom of tree, then the initial radius is replaced by the cumulative metric and the new radius is set. In the same manner the tree search is done for every path in the tree, thus the SD can obtain the ML solution.

2.3. Receiver Structure When Using QRD-M Algorithm

By using QRD-M instead of MLD in the receiver structure in **Figure 1**, we reduce the complexity of MLD. The same signal processing procedure mentioned in 2 is done to cancel the residual ISI and to satisfy the condition as if only the transmit signal at time k is transmitted. After the ISI cancellation, the QRD-M is applied instead of MLD. The number of transmit signal candidates $\hat{\boldsymbol{x}}_{u/k}$ equals K^{n_T}. Like in (5), the time domain transmit signal vector with N points in which the candidate transmit signal is located at time k and the transmit signals at other time instants are all set to 0 is generated. Then the time domain signal vector is transformed to the frequency domain signal vector $\hat{\boldsymbol{X}}_{u/k}(n)$ using FFT with N points.

Then the channel matrix $\boldsymbol{H}_u(n)$ assigned to user u at subcarrier number n is QR decomposed.

$$\boldsymbol{H}_u(n) = \boldsymbol{Q}_u(n)\boldsymbol{R}_u(n) \tag{11}$$

The Hermitian transpose $\boldsymbol{Q}_u^H(n)$ is multiplied by the output $\boldsymbol{Z}_{u/k}(n)$ of the ISI canceller from the left hand side.

$$\tilde{\boldsymbol{Z}}_{u/k}(n) = \boldsymbol{Q}_u^H(n)\boldsymbol{Z}_{u/k}(n), \quad n = 1,\cdots,N \tag{12}$$

The squared metric for minimization using $\tilde{\boldsymbol{Z}}_{u/k}(n)$ in (12) is given by

$$\sum_{n=1}^{N}\left\| \begin{pmatrix} \tilde{Z}_{u/k,1}(n) \\ \tilde{Z}_{u/k,2}(n) \\ \vdots \\ \tilde{Z}_{u/k,n_R}(n) \end{pmatrix} - \begin{pmatrix} R_{u,11}(n) & R_{u,12}(n) & \cdots & R_{u,1n_T}(n) \\ 0 & R_{u,22}(n) & \ddots & R_{u,2n_T}(n) \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & R_{u,n_Rn_T}(n) \end{pmatrix} \begin{pmatrix} \hat{X}_{u/k,1}(n) \\ \hat{X}_{u/k,2}(n) \\ \vdots \\ \hat{X}_{u/k,n_T}(n) \end{pmatrix} \right\|^2 \tag{13}$$

Using M algorithm, (13) is step by step calculated from the bottom to the top. The m survival paths with the least cumulative metrics are retained at each step from the bottom. The path which minimizes (13) is finally selected from the m survival paths which reach the top. The path obtained by ORD-M determines the output $\hat{\boldsymbol{x}}_{uk}$. The signal processing afterward is the same as MLD.

2.4. Receiver Structure When Using SD Algorithm

By using SD instead of MLD in the receiver structure in **Figure 1**, we reduce the complexity of MLD. The same signal processing procedure mentioned in 2 is done to cancel the residual ISI and to satisfy the condition as if only the transmit signal at time k is transmitted. In the proposed SD, the initial radius is set using QRD-M. (13) is used for the search of initial radius using QRD-M. The cumulative metric with small radius is firstly searched in the tree using QRD-M and we set this cumulative metric as the initial radius. Next the transmit signal candidate which satisfies the initial radius is searched toward the depth direction in the tree. When the cumulative distance metric exceeds the initial radius, then the subsequent search along the path is no more needed, thus the amount of calculation is saved. If the cumulative distance metric does not exceeds the initial radius till the bottom of tree, then the initial radius is replaced by the cumulative metric and the new radius is set. In the same manner, the tree search is done for every path in the tree, thus the SD can obtain the ML solution. In (13) the search procedure is done toward the upward direction with exhaustive search to obtain the ML solution of $\hat{\boldsymbol{x}}_{uk}$. If the QRD-M can find a smaller initial radius, then more effectively the tree search is done.

2.5. Realization of Soft Output in SD

We aimed to obtain the soft output from the SD in **Figure 1**. In case of QPSK, the bit LLR's for the 1st bit and the 2nd bit of the transmit signal from antenna $i\ (=1,\cdots,n_T)$ are given by (14) and (15) respectively.

$$\mathrm{LLR}_{1,i} \cong \left(-\min\left[\sum_{n=1}^{N}\left\|\tilde{\boldsymbol{Z}}_{u/k}(n)-\boldsymbol{R}_u(n)\hat{\boldsymbol{X}}_{u/k,i,1(0)}(n)\right\|^2\right]+\min\left[\sum_{n=1}^{N}\left\|\tilde{\boldsymbol{Z}}_{u/k}(n)-\boldsymbol{R}_u(n)\hat{\boldsymbol{X}}_{u/k,i,1(1)}(n)\right\|^2\right]\right)\Big/2\sigma^2 \tag{14}$$

$$\mathrm{LLR}_{2,i} \cong \left(-\min\left[\sum_{n=1}^{N}\left\|\tilde{\boldsymbol{Z}}_{u/k}(n)-\boldsymbol{R}_u(n)\hat{\boldsymbol{X}}_{u/k,i,2(0)}(n)\right\|^2\right]+\min\left[\sum_{n=1}^{N}\left\|\tilde{\boldsymbol{Z}}_{u/k}(n)-\boldsymbol{R}_u(n)\hat{\boldsymbol{X}}_{u/k,i,2(1)}(n)\right\|^2\right]\right)\Big/2\sigma^2 \tag{15}$$

$\hat{\boldsymbol{X}}_{u/k,i,1(0)}(n)$ in (14) denotes the transmit signal in frequency domain of u-th user at time k and frequency point n from transmit antenna i with the 1st bit being 0. $\hat{\boldsymbol{X}}_{u/k,i,1(1)}(n)$ in (14) has the same notation but with the 1st bit being 1. $\hat{\boldsymbol{X}}_{u/k,i,2(0)}(n)$ and $\hat{\boldsymbol{X}}_{u/k,i,2(1)}(n)$ in (15) represent the same notation but with the 2nd bit being 0 and 1 respectively. In SD, there exist some paths for which searches are not made in the tree. In order to calculate the bit LLR, the path for bit "0" and the path for bit "1", both of which have minimum path metrics, have to be evaluated. For this evaluation, we have used the RTS [7] and STS [8] algorithms.

2.6. RTS Algorithm

In RTS, using (13) and the M algorithm, the path with minimum path metric is obtained firstly and is regarded as the initial radius of SD. Then, the path metric which is not yet searched is calculated through SE algorithm [1] [2]. In RTS, to evaluate the bit LLR, the SE algorithm is repeatedly applied to calculate the path metric which is not searched in SD. In **Figures 3(a)-(c)**, we show the tree structure for BPSK when the number of transmit antenna is 3, for example. In **Figure 3(a)**, the red line shows the minimum path metric [101] obtained from the M algorithm with $m=1$. In this case, in order to obtain the bit LLR of $\hat{x}_3$, we have to find the minimum path metric for which $\hat{x}_3=0$, which is illustrated in **Figure 3(b)**. Likewise, in order to obtain the bit LLR's of $\hat{x}_2$ and $\hat{x}_1$, we have to find the minimum path metrics for which $\hat{x}_2=1$ and $\hat{x}_1=0$, those are illustrated in **Figure 3(c)** and **Figure 3(d)** respectively. To find the minimum path metrics having the counter bits, we repeatedly use the SE algorithm.

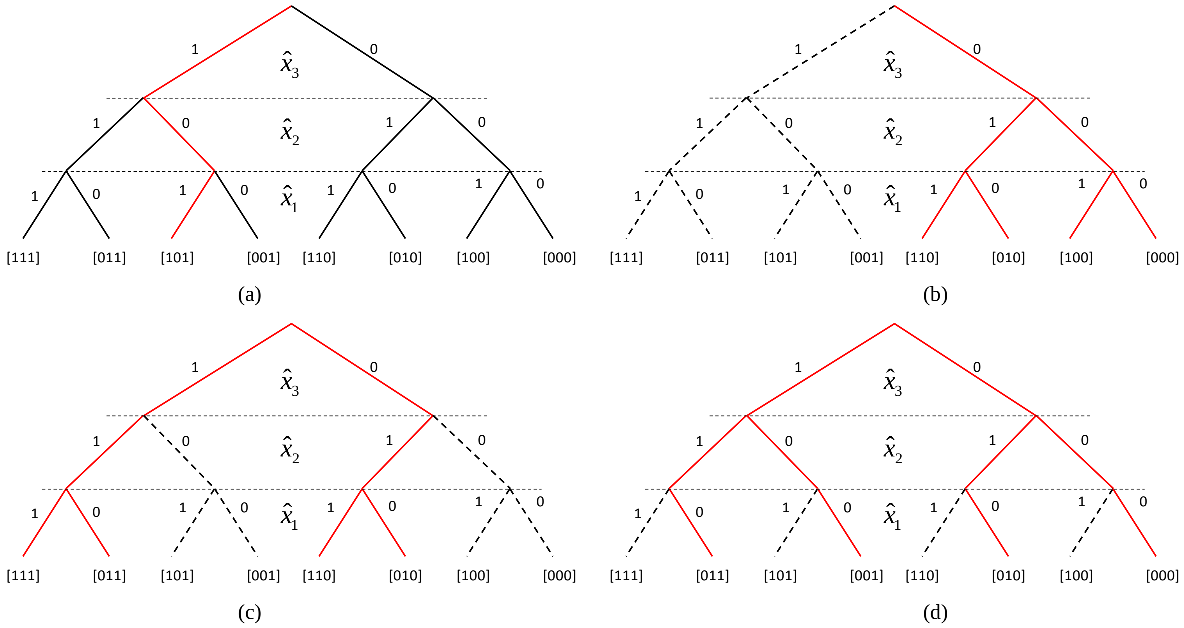
2.7. STS Algorithm

In STS, the path metrics for calculating the bit LLR's are evaluated using the single search of the tree. The basic idea of STS follows that every path metric and its search depth are stored in the list and monitored. When the evolution of all the path metrics in the list does not occur during the tree search, the search of specific branch is saved and this results in complexity reduction. At the initial stage, the values in the list are all set to infinity. In **Figure 4**, we show the STS algorithm where the number of transmit antenna is 4, the number of modulation level K, $\lambda_{s,b}$ the accumulated norm with the search depth s and the symbol number b, λ_{ML} the accumulated norm of ML sequence. Also, the list as an example is illustrated in **Figure 5** where the number of transmit antenna is 4 and the QPSK modulation is used.

In STS algorithm, the list in **Figure 5** is filled up with the algorithm in **Figure 4**. When $\lambda_{3,2}$ is calculated for example, its value is compared with all $\lambda_{1\sim2,1\sim4}$ already stored in the list. At this stage, when $\lambda_{3.2}<\lambda_{1\sim2,1\sim4}$, we find that the further search of this branch does not lead to the evolution of the path metric. Accordingly we stop the search and move to the calculation of $\lambda_{3.3}$. When λ_{ML} is finally obtained, the needed norms are read from the list and the bit LLR's are calculated using (14) and (15).

3. Computer Simulation Results

Computer simulations are made for the system in **Figure 1**. The simulation conditions are listed in **Table 1**. **Figure 6** shows the BER characteristics when the hard decision replica is used to cancel the ISI through MLD, QRD-M or SD for spatial de-multiplexing. In **Figure 6**, #4, for example, denotes the number of MLD, QRD-M or SD iterated. **Figure 7** shows the BER characteristics when the soft decision replica is used to cancel the ISI through RTS or STS for spatial de-multiplexing. In **Figure 8**, we compared the BER characteristics between hard replica cancellation and soft replica cancellation with iteration being used. Also in **Figure 6-8**, we showed



(a)

(b)

(c)

(d)

Figure 3. Tree search algorithm in RTS. (a) Path with minimum path metric; (b) Paths for calculating the minimum counter path metrics for antenna 3; (c) Paths for calculating the minimum counter path metrics for antenna 2; (d) Paths for calculating the minimum counter path metrics for antenna 1.

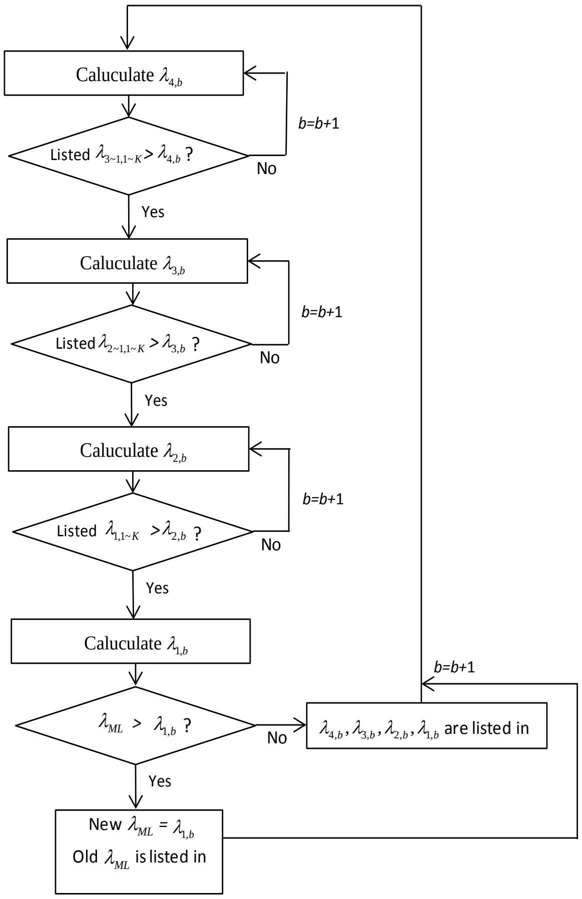
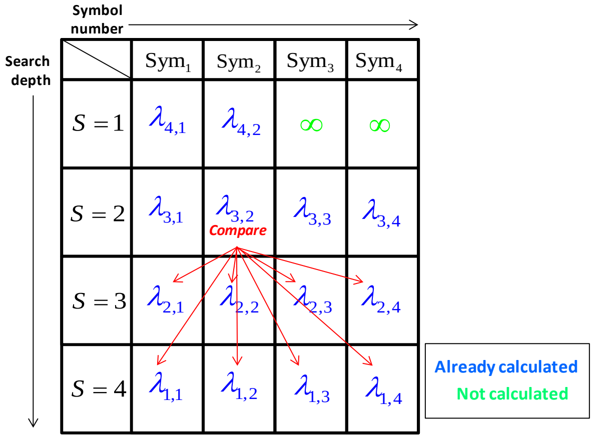

Caluculate $\lambda_{4,b}$
Listed $\lambda_{3\sim1,1\sim K} > \lambda_{4,b}$?
No
b=b+1
Yes
Caluculate $\lambda_{3,b}$
Listed $\lambda_{2\sim1,1\sim K} > \lambda_{3,b}$?
No
b=b+1
Yes
Caluculate $\lambda_{2,b}$
Listed $\lambda_{1,1\sim K} > \lambda_{2,b}$?
No
b=b+1
Yes
Caluculate $\lambda_{1,b}$
$\lambda_{ML} > \lambda_{1,b}$?
No
$\lambda_{4,b}, \lambda_{3,b}, \lambda_{2,b}, \lambda_{1,b}$ are listed in
b=b+1
Yes
New $\lambda_{ML} = \lambda_{1,b}$
Old λ_{ML} is listed in

Figure 4. STS algorithm.



	Sym_1	Sym_2	Sym_3	Sym_4
$S=1$	$\lambda_{4,1}$	$\lambda_{4,2}$	∞	∞
$S=2$	$\lambda_{3,1}$	$\lambda_{3,2}$ Compare	$\lambda_{3,3}$	$\lambda_{3,4}$
$S=3$	$\lambda_{2,1}$	$\lambda_{2,2}$	$\lambda_{2,3}$	$\lambda_{2,4}$
$S=4$	$\lambda_{1,1}$	$\lambda_{1,2}$	$\lambda_{1,3}$	$\lambda_{1,4}$

Figure 5. Example of list with 4 transmit antennas and the modulation level *K*.

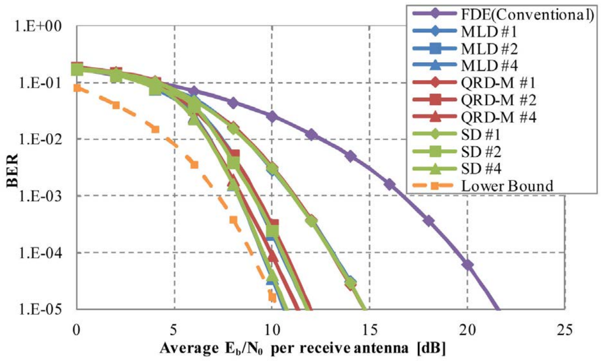


Figure 6. Comparison of BER characteristics of MIMO interleaved SC-FDMA receiver with hard replica cancellation of ISI.

Table 1. Simulation condition.

Number of UE	4
Number of transmit antennas in each UE	4
Number of receive antennas at BS	4
Modulation formats	QPSK
Number of total subcarriers	$M = 256$
Number of subcarriers assigned to each user	$N = 64$
Symbol length of QPSK	T
Cyclic prefix length	$4T/$
Channel model between each transmit and receive antenna	Equal power 16 paths quasi-static Rayleigh fading channel
Interval of delay time	$T/4$
Subcarrier assignment	IFDMA
Channel estimation	Perfect at BS
FDE	Nulling (MMSE)
Initial radius setting for SD (SE algorithm)	QRD-M ($m = 1$)
Number of iterative feedbacks in the receiver	0,1,3 (= #−1) # denotes the repetition number of MLD, QRD-M, SD, RTS or STS

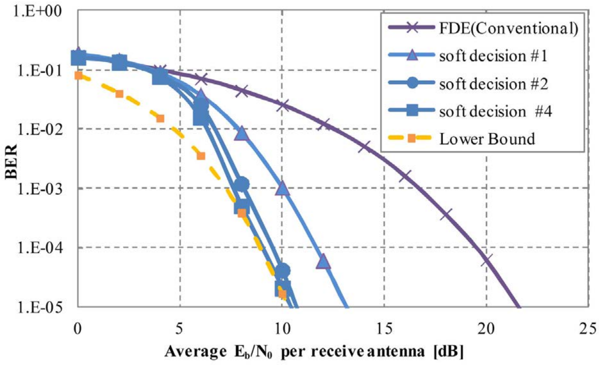


Figure 7. Comparison of BER characteristics of MIMO interleaved SC-FDMA receiver with soft replica cancellation of ISI.

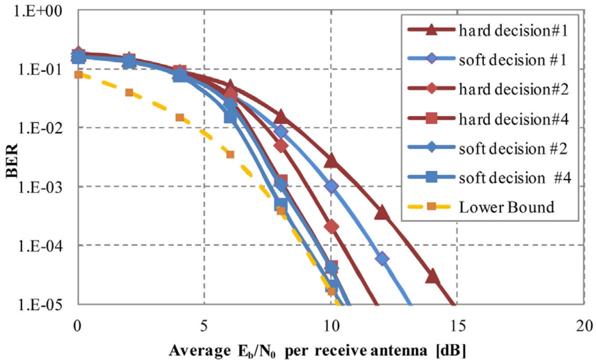


Figure 8. Comparison of BER characteristics of MIMO interleaved SC-FDMA receiver between hard decision and soft decision with iterative feedback.

the lower bound of BER where the ISI cancellation is perfect, which means the demodulated bits for ISI cancellation are error-free. In **Figure 9**, we show the comparison of complexity of MLD, QRD-M, SD, RTS and STS on 4×4 flat fading channel. This complexity is measured using "tic" and "toc" function in MATLAB and the computation time needed for MLD is normalized to unity.

From **Figure 6**, compared with the conventional FDE receiver, the proposed receiver using MLD with no iterative feedback improves the BER by about 7 dB at $\text{BER} = 10^{-5}$. By increasing the number of iterative feedbacks, the proposed receiver further improves the BER and obtains the BER improvement of more than 10 dB which is close to the lower bound of BER. This is because the MLD outputs with high reliability are used as the improved decision results for making the accurate ISI replicas. Accordingly more exact ISI cancellation becomes possible followed by improved MLD performance. We observe the BER performance of QRD-M is inferior to MLD, but the BER of SD coincides with the MLD, thus the SD can obtain the ML solution.

From **Figure 7**, we see that the BER performance with soft ISI cancellation behaves basically the same as the SD with hard ISI cancellation in **Figure 6**, but the BER approaches more rapidly to the lower bound than the hard ISI cancellation. We find that at average $E_b/N_0 = 8\ (\text{dB})$ the BER coincides with the lower bound and this means the perfect ISI cancellation is possible at this receive SNR value.

From **Figure 8**, we see that the soft ISI cancellation with iterative feedback performs better than the hard replica cancellation. This is because more accurate ISI replica for cancellation can be generated for the soft decision than the hard decision.

From **Figure 9**, QRD-M, SD, RTS and STS can reduce the computation time compared with MLD. The QRD-M is the most effective in reducing the computation time. The computation time is almost 1/100 of MLD and is constant over the average E_b/N_0 value. However, QRD-M is sub-optimal and ML solution is not obtained. The SD can obtain the ML solution, but for low average E_b/N_0 region less than 10 dB the computation

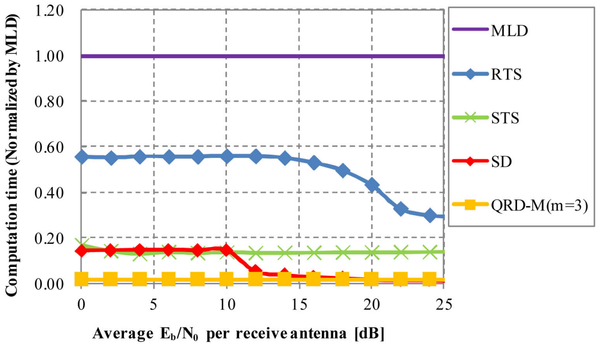


Figure 9. Comparison of computation time among MLD, QRD-M, SD, RTS and STS on flat fading channel.

time is about 15 times higher than QRD-M. However, above $E_b/N_0 = 10\ (\text{dB})$, the computation time approaches to QRD-M. This is because for high average E_b/N_0 region, the initial radius can be set to a very small value. Although the RTS can produce the soft output, the RTS needs a lot of path metric calculation leading relatively high computation time. The STS which is the improved version of RTS shows the computation time almost the same as the SD in low E_b/N_0 region. Therefore it can reduce the computation time about 1/5 compared with the RTS. However, the computation time of STS is almost constant over entire E_b/N_0 region.

4. Conclusion

In this paper, we have proposed the low BER receiver structure for the interleaved SC-FDMA on the uplink MIMO frequency selective fading channels. In the proposed receiver, using the tentative decision results obtained from the MMSE nulling (FDE), the ISI components are cancelled and the MLD is then used for separating the spatially multiplexed signal streams. The reliable output from MLD is again fed back to the ISI canceller to reduce the residual ISI. Furthermore we improve the complexity of MLD by replacing it with QRD-M or SD. We have verified the BER characteristics of the proposed receiver with MLD, QRD-M or SD through computer simulations. The receiver with SD achieves the same BER as the one with MLD, *i.e.*, ML solution, whereas the QRD-M has the inferior BER because of its quasi-ML solution. We have also verified that the complexity of SD is very much improved compared with MLD especially in high E_b/N_0 region. In order to cancel the ISI more effectively using soft replica, we have further replaced the SD by RTS or STS algorithm in which the soft out from SD is available. As a result, the BER characteristic approaches more rapidly to the lower bound. The complexity of STS is lower than the RTS and almost coincides with the SD in low E_b/N_0 region. The proposed receiver structure will be useful to extend the coverage of uplink.

Acknowledgements

This study has been supported by the Scientific Research Grant-in-aid of Japan No. 24560454 and Sharp cooperation.

References

[1] Guo, Z. and Nilsson, P. (2004) Reduced Complexity Schnorr-Euchner Decoding Algorithms for MIMO Systems. *IEEE Communications Letters*, **8**, 286-288. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1300579 http://dx.doi.org/10.1109/LCOMM.2004.827376

[2] Shim, B. and Kang, I. (2008) Sphere Decoding with a Probabilistic Tree Pruning. *IEEE Transactions on Signal Processing*, **56**, 4867-4878. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4626106 http://dx.doi.org/10.1109/TSP.2008.923808

[3] Vikalo, H. and Hassibi, B. (2002) Maximum-Likelihood Sequence Detection of Multiple Antenna Systems over Dispersive Channels via Sphere Decoding. *EURASIP Journal on Applied Signal Processing*, **2002**, Article ID: 156743. http://dx.doi.org/10.1155/S1110865702204011

[4] Myung, H.G., Lim, J. and Goodman, D.J. (2006) Single Carrier FDMA for Uplink Wireless Transmission. *IEEE Vehicular Technology Magazine*, **1**, 30-38. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4099344 http://dx.doi.org/10.1109/MVT.2006.307304

[5] Moriyama, M. and Iwanami, Y. (2012) Complexity Reduction Using QRD-M or SD in MIMO Interleaved SC-FDMA Receiver with Iterative Detection. *International Symposium on Information Theory and Its Applications*, Honolulu, 28-31 October 2012, 145-149. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6400904

[6] K.J., Kim, Jiang, Y., Iltis, R.A. and Gibson, J.D. (2005) A QRD-M/Kalman Filter-Based Detection and Channel Estimation Algorithm for MIMO-OFDM Systems. *IEEE Transactions on Wireless Communications*, **4**, 710-721. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1413237 http://dx.doi.org/10.1109/TWC.2004.842951

[7] Studer, C. and Burg, A. (2008) Soft-Output Sphere Decoding: Algorithms and VLSI Implementation. *IEEE Journal on Selected Areas in Communications*, **26**, 290-300. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4444760 http://dx.doi.org/10.1109/JSAC.2008.080206

[8] Studer, C. and Bölcske, H. (2010) Soft-Input Soft-Output Single Tree-Search Sphere Decoding. *IEEE Transactions on Information Theory*, **56**, 4827-4842. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5571884 http://dx.doi.org/10.1109/TIT.2010.2059730

Int. J. Communications, Network and System Sciences, 2014, 7, 519-528
Published Online December 2014 in SciRes. http://www.scirp.org/journal/ijcns
http://dx.doi.org/10.4236/ijcns.2014.712052

Proxy Server Experiment and Network Security with Changing Nature of the Web

Olatunde Abiona[1], Adeniran Oluwaranti[2], Ayodeji Oluwatope[2], Surura Bello[2], Clement Onime[3], Mistura Sanni[2], Lawrence Kehinde[4]

[1]Department of Computer Information Systems, Indiana University Northwest, Garry, USA
[2]Department of Computer Science and Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria
[3]Information and Communication Technology Section, Abdus Salam International Centre for Theoretical Physics, Trieste, Italy
[4]Department of Electrical and Electronic Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria
Email: oabiona@iun.edu, aranti@oauife.edu.ng, aoluwato@oauife.edu.ng, apinkebello@yahoo.com, onime@ictp.it, misturasanni@gmail.com, lokehinde@oauife.edu.ng

Received 12 October 2014; revised 22 November 2014; accepted 2 December 2014

Abstract

The total reliance on internet connectivity and World Wide Web (WWW) based services is forcing many organizations to look for alternative solutions for providing adequate access and response time to the demand of their ever increasing users. A typical solution is to increase the bandwidth; this can be achieved with additional cost, but this solution does not scale nor decrease users perceived response time. Another concern is the security of their network. An alternative scalable solution is to deploy a proxy server to provide adequate access and improve response time as well as provide some level of security for clients using the network. While some studies have reported performance increase due to the use of proxy servers, one study has reported performance decrease due to proxy server. We then conducted a six-month proxy server experiment. During this period, we collected access logs from three different proxy servers and analyzed these logs with Webalizer a web server log file analysis program. After a few years, in September 2010, we collected log files from another proxy server, analyzed the logs using Webalizer and compared our results. The result of the analysis showed that the hit rate of the proxy servers ranged between 21% - 39% and over 70% of web pages were dynamic. Furthermore clients accessing the internet through a proxy server are more secured. We then conclude that although the nature of the web is changing, the proxy server is still capable of improving performance by decreasing response time perceived by web clients and improved network security.

Keywords

Proxy Server, Network Security, Hit Ratio, Webalizer, Proxy Log Analysis

How to cite this paper: Abiona, O., Oluwaranti, A., Oluwatope, A., Bello, S., Onime, C., Sanni, M. and Kehinde, L. (2014) Proxy Server Experiment and Network Security with Changing Nature of the Web. *Int. J. Communications, Network and System Sciences*, **7**, 519-528. http://dx.doi.org/10.4236/ijcns.2014.712052

1. Introduction

Many organizations today rely heavily on the use of the internet and the WWW; this has open doors for network administrators to acquire skills to manage the ever growing demand for access and good response time. A typical solution to providing access and good response time is to increase the bandwidth; this is not a scalable option. An alternative solution is to deploy proxy servers to service the ever increasing request of users.

A proxy server is a server that sits between a client application, such as a web browser, and a real server. It intercepts all requests to the real server to see if it can fulfill the requests itself. If not, it forwards the request to the real server. A proxy server can improve network performance by functioning as a caching server. Most Internet Service Provider (ISP) and organizations have been installing proxy caches to reduce bandwidth and decrease the latency to their users [1]-[5]. The performance increase due to proxy servers has been widely reported; however, a study reports that proxy servers actually decrease performance [6]. A pertinent question that comes to our mind is that since the web is evolving from static to dynamic information repository, is there a future for the caching proxy server?

In order to further understand the nature of proxy server and how it can be used to provide improved access and response time to a large number of users requesting same object from the cache, we conducted a proxy server experiment. A non-intrusive network traffic monitoring system was setup in [7] to collect access logs from three proxy servers, for a period of five months to three years. These access logs were analyzed using Webalizer. The three proxy servers are institutional web proxy cache. Two of the proxy servers are on the academic network, the Obafemi Awolowo University, Ile-Ife, Nigeria (OAU), the Indiana University Northwest Computer Networking Lab in HH226, Gary Indiana (IUN). The third proxy is on the Wide Area Network of the International Centre for Theoretical Physics, Trieste, Italy (ICTP).

The rest of the paper is organized as follows. In Section 2, we review related work, followed by data collection in Section 3. In Section 4, we perform access log analysis on raw data and reduced data. In Section 5, we present the results of our analysis for each caching proxy server and finally, in Section 6, we conclude the paper.

2. Related Work

Caching can be applied at several locations, namely at the web client, web server and within the network (proxy servers) [8]-[10]. Caching proxy server has gained popularity on the Internet, due to their ability to keep local copies of documents requested by web clients and using them to satisfy future request for same document. This can save bandwidth and reduce delays perceived by web users.

Several studies have reported performance increase due to proxy servers. One of the major functions of a caching proxy server is to decrease access time. The result of a study in [11] showed that the average response time of a hit may be five times smaller than a miss. A 20% to 25% improvement in user perceived response time was reported in [12] [13]. Research on the effectiveness of proxy caching is very active. A study at Virginia Tech has shown that hit rates of 30% to 50% can be achieved by a caching proxy [14]. Other studies gave a range from between 20% to 60% hit rate [9] [11] [15] [16] and [17] reported hit rate of between 10% to 40% for a three level caching hierarchy, and about 35% to 40% for a university-level web proxy cache.

However, a study conducted in [6] reported a hit rate of 4%, which shows a decrease in performance. The reason for this decrease in performance was traced to the changing nature of the web, *i.e.* the web is evolving from static nature to dynamic repository. Furthermore, research into the ability of proxy servers to cache video was reported in [6]. In the last few years there have been research efforts to improve multi-level proxy cache configuration [18]-[21]. Other factors that may improve proxy cache performance are the replacement polices used by the cache and the workload characteristics. The results of [17] showed that combining different replacement polices at different levels of the cache can improve the performance of a caching hierarchy. Finally the results of [22] showed that the cache replacement polices are sensitive to Zipf slope, temporal locality, and correlation between file size and popularity but relatively insensitive to one-timers, and heavy tail index.

3. Raw Data Collection

We collected access logs from three proxy servers located at three different locations. Two of the proxy servers are located at the Obafemi Awolowo University, Ile-Ife, Nigeria and Indiana University Northwest, Gary, Indiana computer networking lab. The third proxy server is located at the International Center for theoretical Physics, Trieste, Italy; we refer to the proxies as follows:

- ASOJU used by the OAU academic network;
- IUN used by only the students in computer networking lab HH226;
- ICTP used by the ICTP network.

ASOJU continuously recorded access log on a daily basis for six months, details can be found in [7], The IUN records proxy logs during the academic year (August-December and January-May) for a period of three years, while ICTP proxy server had only one month of access log. Two of the proxies are institutional-level proxy servers while the third is only used by students in the networking lab HH226.

4. Access Log Analysis

4.1. Raw Data Analysis

Webalizer [23] is capable of generating reports on a monthly basis and also a summary report for the entire period. We have five months summary, from September 30, 2006 to February 28, 2007 for the first OAU proxy server which is referred to as ASOJU access log. The five-month ASOJU access log recorded a total of 153,125,959 requests in 107 days of activity. The access logs for 45 days were not available due to down time and power outages. Similarly, we have three years of access log from September 2010 to October 2013 for the IUN proxy server which is referred to as IUN access log. The Three years IUN access log recorded a total of 62,675,342 requests in 210 days of activity. The access log was only collected when students are using the lab during the semester. Hence the need to collect log files for a longer period since the lab is only in use three months in a year. The eight days ICTP access log referred to as ICTP recorded a total of 5,458,868 requests in 8 days.

Table 1 provides a summary of the access logs for the three proxy servers studied. ASOJU has the highest activity in terms of number of request per day and also the highest average volume of bytes transferred per day.

In this study we are interested in requests for the transfer of web documents. Hence we study the response code in the access logs for all web requests. The breakdown of the HTTP reply code as a percentage of the total request is shown in **Table 2**. Web proxy server can provide many possible responses to web client [24]. Here are some response code and their corresponding meaning: The 200 series response code means a valid document was made available to the client, 300 series means redirection, 400 series means client error and 500 series means server error.

4.2. Raw Data Reduction Analysis

The access log recorded the amount of data transferred regardless of the source (*i.e.* from proxy cache, another cache or origin server). To know the actual workload of a proxy server, we consider all requests resulting in the

Table 1. Summary of proxy access logs (raw data).

Item	ASOJU	IUN	ICTP
Access log duration	107 days	210 days	8 days
Start date	Sept 30 2006	Sept 2010	Feb 11 2007
End date	Feb 28 2007	Oct 2013	Feb 18 2007
Total request	153,129,674	62,675,324	5,458,868
Avg request/day	1,431,118	298,453	682,359
Total bytes transferred (GB)	1637.4	495.9	86.3
Avg bytes/day (GB)	15.30	2.36	10.79

Table 2. Breakdown of HTTP response code.

Response code	ASOJU (%)	IUN (%)	ICTP (%)
200 series	38.34	67.00	45.20
300 series	12.30	20.00	28.00
400 series	46.00	9.00	2.70
500 series	2.16	1.00	0.90
Undefined	1.20	3.00	23.20
Total	100	100	100

documents being accessed from the origin server without an intermediate proxy. The objective is to evaluate the effectiveness of proxy caching.

Suppose a client using a proxy makes requests $r_1, r_2, \cdots, r_n$ to pages, if a page has F objects out of which C can be obtained from the cache and W from the origin server. Total request R will be:

$$R = \sum_{i=1}^{n} r_i .$$

But not all requests will bring back data. Hence, all requests that will result in data transfer will be,

$$F = \sum_{i=1}^{n} W_i + \sum_{i=1}^{n} C_i .$$

So we can compute the document hit ratio (DHR) and byte hit ratio (BHR) as,

$$\text{DHR} = \frac{\sum_{i=1}^{n} C_i}{\sum_{i=1}^{n} W_i + \sum_{i=1}^{n} C_i}, \quad \text{and} \quad \text{BHR} = \frac{\text{Cache byte}}{\text{Total byte}}$$

Cache byte = the no of bytes transferred from the cache;

Total byte = the total no of bytes transferred.

For DHR we only considered 200 and 300 series of response, in order to consider only successful transfer of documents to requesting clients. For BHR we did not consider the 400 series (client error). **Table 3** summarizes the reduced access logs for the three proxies. Based on the average number of request seen by each proxy server per day, ASOJU has the highest activity while IUN and ICTP have about the same activity. The successful transfer accounted for 45% to 87% while the total bytes transferred accounted for 64% to 89% similar to the observation in [19]. Other values on the table were calculated. The two performance metrics used in this study to evaluate the performance of the proxy servers are DHR and BHR.

5. Results

We observe that the total requests in the reduced data for ASOJU is smaller, this is expected since about 46% of the total request are error due to client authentication see **Table 2**. This is possible because ASOJU runs proxy authentication. Again about 60% to 78% of the requests are for dynamic pages that cannot be serviced by the proxy server. These observations support the fact that the web is fast changing from static nature to dynamic information repository [6]. However, the DHR range from 21% to over 38% for the three proxy servers analyzed in the study, these results are similar to the results obtained in [9] [11] [14]-[16]. Similarly, the BHR range from 21% to 29% for the three proxy servers. This result is also comparable with [11]. Since The ICTP data was only collected for only 8 days in the month, we can only plot the graphs of the hit ratios for ASOJU and IUN using the reduced data for a six-month period.

Figure 1 and **Figure 2** show that both hit ratios for ASOJU and IUN are not affected by the volume of the workloads across the months. We further study this observation on monthly hourly raw data. We are unable to generate the hourly reduced data, since the breakdown of the HTTP response used for generating the reduced data can only be obtained for monthly data. We study the monthly variations of the mean hit ratios across the hours of the day for the three proxy servers. The *y* error bars on the graph shows the variability of the hit ratios across the hours. We observe that our hit ratios in the following monthly graphs are relatively lower, varying in the 2% to 8% range. This is expected since the raw data contains client errors that were not removed. We also plot the mean monthly requests for the three proxy servers, in order to identify the peak periods of the day for the servers, since it varies.

Figure 3 shows the mean monthly hourly requests for ASOJU, the high usage periods (peak periods) are 9 hrs to 17 hrs and the low usage periods are 18 hrs to 8 hrs. This graph shows a typical work or social pattern in the environment. The traffic volume rises steadily with some deeps indicating break periods and fall steadily during the close of work for the day. It gives a representation of the user's access pattern. The graph shows that monthly hourly requests follow a normal distribution.

Figure 4 and **Figure 5** show the variations in the monthly average hit ratios for ASOJU. Both hit ratios

Table 3. Summary of proxy access logs (reduced data).

Item	ASOJU	IUN	ICTP
Total request	77,580,469	88,922,534	3,998,971
Avg request/day	725,051	499,565	499,871
Total bytes transferred (GB)	884.2	878.0	84.0
Avg bytes/day (GB)	8.26	4.93	10.5
Total internet requests	58,707,185	69,945,197	2,469,348
Total cache requests	18,873,284	18,977,337	1,529,623
Total cache byte (GB)	108.7	184.4	23.5
DHR (%)	24.3	21.7	38.3
Cache miss rate (%)	75.7	78.3	61.7
BHR (%)	22.8	21.4	28.8

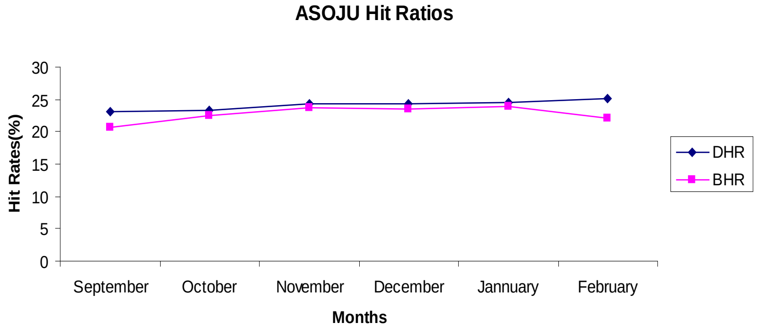


Figure 1. ASOJU hit ratios for the reduced data.

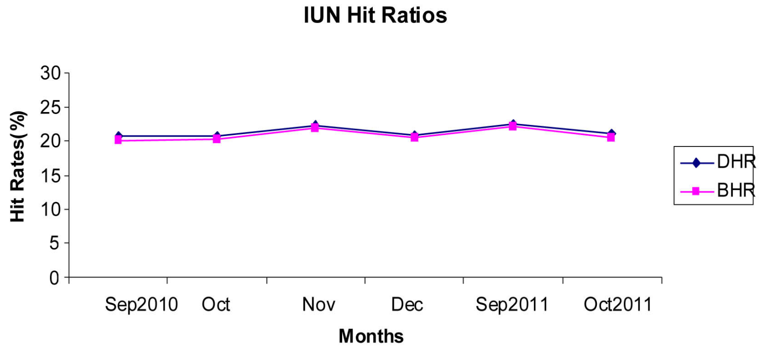


Figure 2. IUN hit ratios for the reduced data.

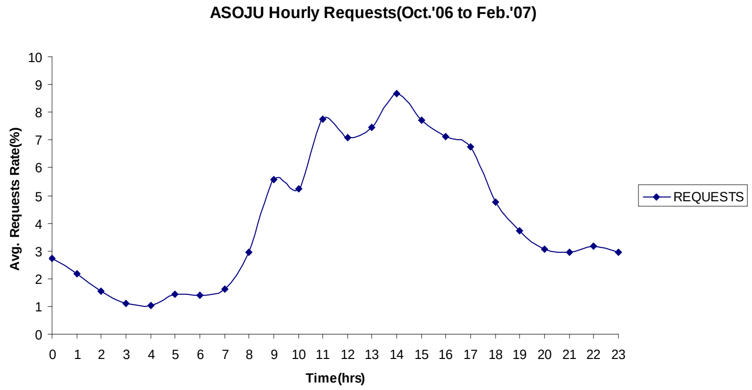


Figure 3. Mean hourly requests for ASOJU.

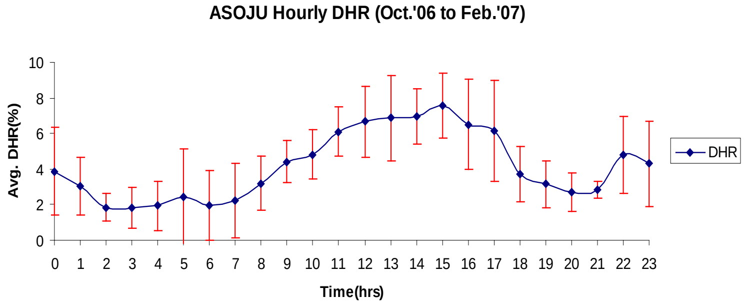


Figure 4. Variation in DHR for ASOJU.

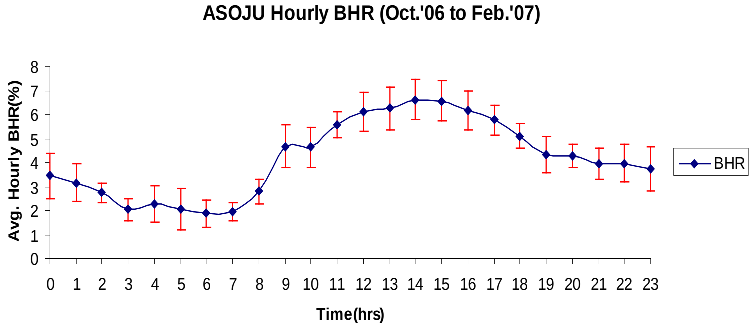


Figure 5. Variation in BHR for ASOJU.

follow a similar trend, the standard deviation shown by the *y* error bars have a high dispersion for both ratios during the peak periods. This is expected since the traffic intensity increases during the peak periods. The variation of the DHR is higher; this is a reflection of the replacement algorithm and size of objects cached by the proxy server. This particular proxy is configured to cache small objects. Hence higher values of DHR, this will result in faster response time for the users.

Figure 6 shows the coefficient of variation (COV) for ASOJU hit ratios. The hit ratios show low variations during the peak periods (9 hrs to 15 hrs). This shows that neither ratio depend on traffic intensity.

Figure 7 shows the mean monthly hourly requests for IUN, the high usage periods (peak periods) are 0 hrs to 17 hrs and the low usage periods are 18 hrs to 20 hrs. This graph shows a typical access pattern for a student lab, the traffic volume is high for most time of the day with a small deep and then rise again. This pattern is however different from the access pattern of an academic network which has a high traffic during office hours (8 am - 5 pm).

Figure 8 and **Figure 9** show the variations in the monthly average hit ratios for IUN. Again, both hit ratios follow a similar trend, the standard deviation shown by the *y* error bars have a high dispersion for both ratios during the peak periods. This is expected since the traffic intensity increases during the peak periods. The variation of the BHR is higher; this is a reflection of the replacement algorithm and size of objects cached by the proxy server. This particular proxy is configured to cache large objects. Hence higher values of BHR, this will result in more bandwidth savings for the network.

Figure 10 shows the coefficient of variation for IUN hit ratios. Similarly, the hit ratios show low variations during the peak periods (0 hrs to 17 hrs). Again, this implies that neither ratio depend on traffic intensity.

Figure 11 shows the mean hourly requests for ICTP, the high usage periods (peak periods) are 8 hrs to 23 hrs and the low usage periods are 0 hrs to 7 hrs. The graph shows a typical social or work pattern. The traffic volume rises steadily with some deeps indicating break periods and fall slightly and remain high for the duration of the peak period. The graph shows the users access pattern.

Figure 12 shows the effect of traffic intensity on the ICTP hit ratios. Similarly, the hit ratios show low variations during the peak periods (8 hrs to 23 hrs). Again, this implies that neither ratio depend on traffic intensity. One may expect that when the number of client requests increases (peak periods) so does the number of hits. However, this is not the case since the peak hours user population has access patterns different from light load hours.

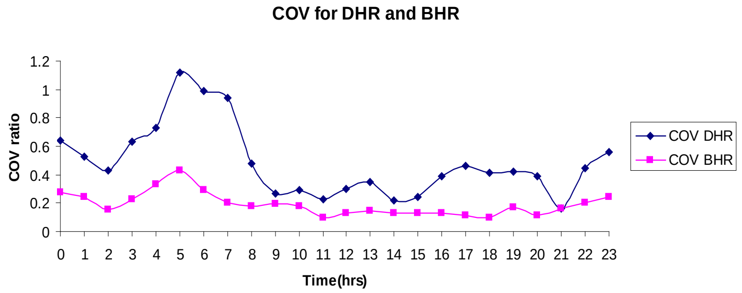


Figure 6. COV for ASOJU hit ratios.

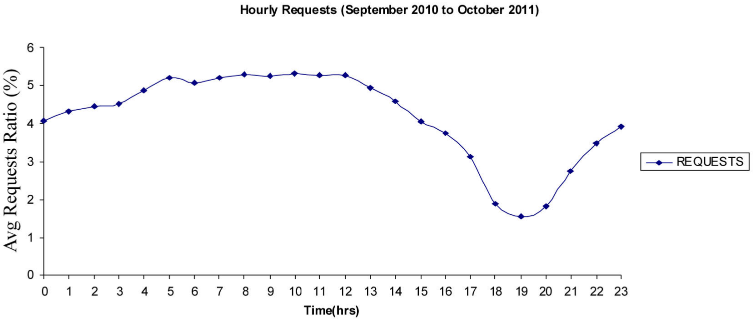


Figure 7. Mean hourly requests for IUN.

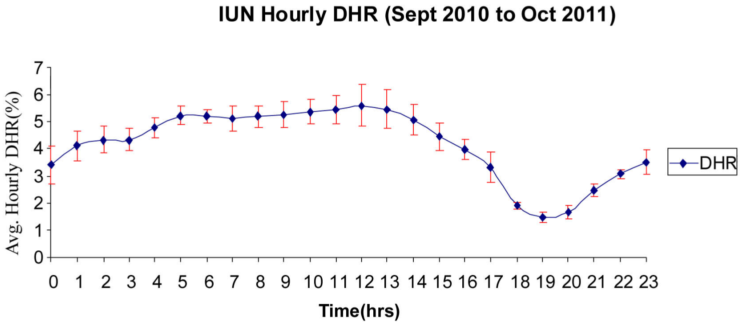


Figure 8. Variation in DHR for IUN.

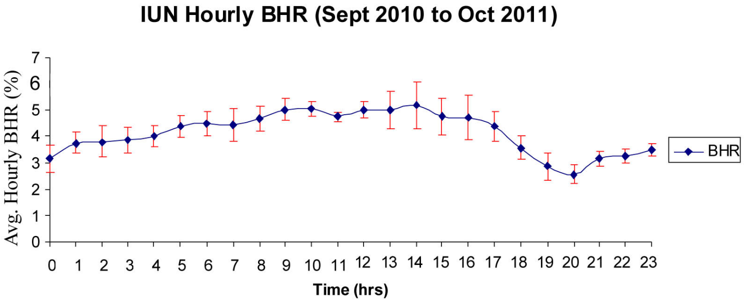


Figure 9. Variation in BHR for IUN.

The study also shows that the proxy server provides better security for clients accessing the internet using a proxy server. Most proxy server run dynamic host configuration protocol (DHCP), a service that provide clients with Internet protocol (IP) addresses that are private and using masquerading or network address translation, the clients can access the internet. From the outside, only the proxy server is visible. All clients using the proxy server are protected from attack, since they are not visible from outside the network. Only the proxy server

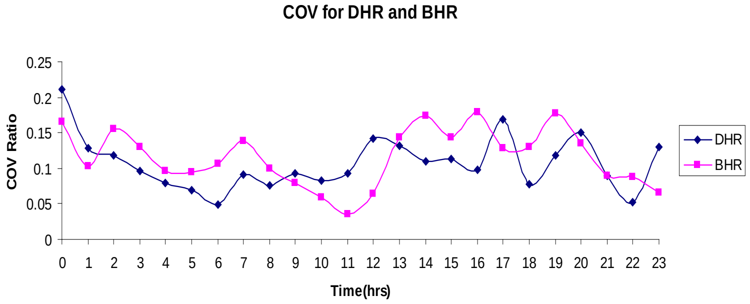


Figure 10. COV for IUN hit ratios.

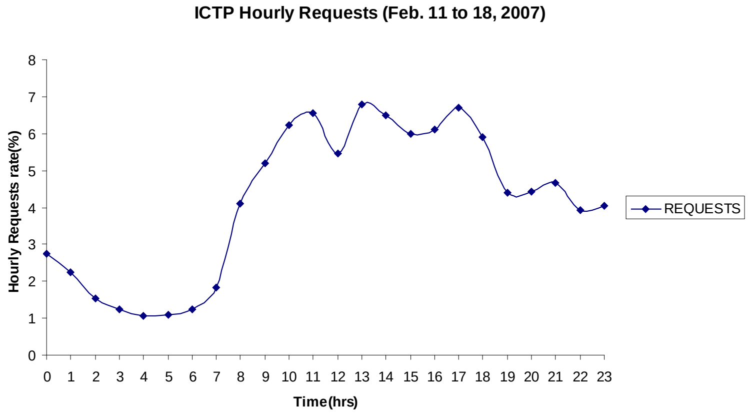


Figure 11. Mean hourly requests for ICTP.

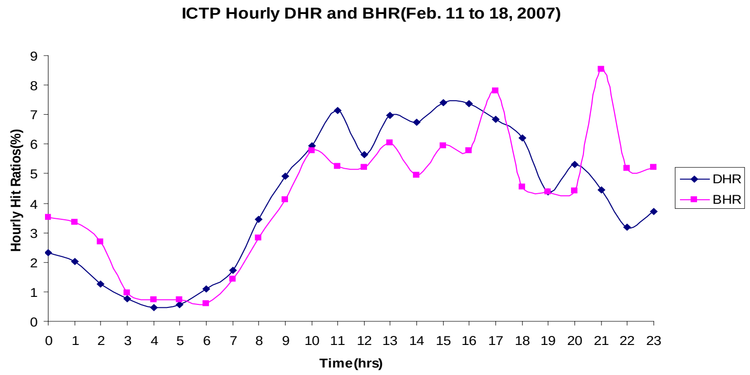


Figure 12. Effect of traffic intensity on the hit ratios.

has a public IP address. We tried to attack clients behind the proxy server with no success. This technique shows that the proxy server provides a layer of security for clients accessing the internet using a proxy server.

6. Conclusion

This paper presents an experiment to determine the effectiveness of proxy servers and security provided by using proxy servers. We are also interested to know how the changing nature of the web has affected the performance of proxy servers and level of security provided by proxy server. We conducted a six-month proxy server experiment to know the performance of proxy servers. Access logs of varying durations were collected, from the three different proxy servers to see if it would have any effect on our results. We analyzed the logs using Webalizer. Two performance parameters—DHR and BHR—were used to evaluate the performance of proxy servers. We compute DHR and BHR for the duration of the study, and we also compute DHR and BHR for monthly and hourly traffic to study the effect of traffic intensity on proxy server performance. The result shows a hit rate of about 21% to 38% and a byte rate of 21% to 28%, and the *y* error bar graphs show a high variation during the peak periods, while the COV graph shows a low or constant variation during the peak periods indicating that neither hit ratios depend on traffic load. The result shows that good performance can be achieved using proxy servers. Although the web is changing from the static nature to dynamic information repository, proxy servers actually improve performance and provide better security despite the changing nature of the web. In the future we hope to look into further enhancing security using honey pots and honey nets. We plan to investigate the cyclic multicast engine and proxy server as a possible technique to improve proxy server performance.

References

[1] Baentsch, M., Barum, L., Molter, G., Rothkugel, S. and Sturm, P. (1997) Enhancing the Web's Infrastructure: From Caching to Replication. *IEEE Internet Computing*, **1**, 18-27. http://dx.doi.org/10.1109/4236.601083

[2] Bestavros, A., Carter, R., Crovella, M., Cunha, C., Heddaya, A. and Mirdad, S. (1995) Application-Level Document Caching in the Internet. *Proceedings of the* 2nd *International Workshop on Services in Distributed and Networked Environments* (*SDNE*'95), Whistler, 166-173.

[3] Cohen, E., Krishnamurthy, B. and Rexford, J. (1998) Improving End-to-End Performance of the Web Using Server Volumes and Proxy Filters. *Proceedings of ACM SIGCOMM*'98 *Conference*, Vancouver, 241-253.

[4] Baentsch, M., Baum, L., Molter, G., Rothkugel, S. and Sturm, P. (1997) World Wide Web Caching: The Application-Level View of the Internet. *IEEE Communications Magazine*, **35**, 170-178. http://dx.doi.org/10.1109/35.587725

[5] Zhang, L., Floyd, S. and Jacobson, V. (1997) Adaptive Web Caching. *Proceedings of the NLANR Web Caching Workshop*, Boulder.

[6] Howard, R. and Jansen, B.J. (1998) A Proxy Server Experiment: An Indication of the Changing Nature of the Web. *Proceedings of the* 7th *International Conference on computer Communications and Networks* (*ICCCN*'98), Washington DC, 646-649.

[7] Abiona, O.O., Onime, C.E., Oluwaranti, A.I., Adagunodo, E.R., Kehinde, L.O. and Radicella, S.M. (2006) Development of a Non Intrusive Network Traffic Monitoring and Analysis System. *African Journal of Science and Technology*, **7**, 1-17.

[8] Rousskov, A. and Soloviev, V. (1998) On Performance of Caching Proxies. *Proceedings of the ACM SIGMETRICS Conference*.

[9] Caceres, R., Douglis, F., Feldmann, A., Glass, G. and Rabinovich, M. (1998) Web Proxy Caching: The Devil Is in the Details. *ACM Performance Evaluation Review*, **26**, 11-15. http://dx.doi.org/10.1145/306225.306230

[10] Abdulla, G., Fox, E., Abrams, M. and Williams, S. (1997) WWW Proxy Traffic Characterization with Application to Caching. Technical Report TR-97-03, Computer Science Department, Virginia Tech.

[11] Rousskov, A. and Soloviev, V. (1998) On Performance of Caching Proxies. *Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems*, **26**, 272-273.

[12] DiDio, L. (1997) Proxy Servers Gain User Appeal. *Computerworld*, **31**, 16.

[13] Machlis, S. (1997) Planning Blunts Web Traffic Spikes. *Computerworld*, **31**, 6.

[14] Williams, S., Abrams, M., Standridge, C.R., Abdulla, G. and Fox, E.A. (1997) Removal Policies in Network Caches for World-Wide Web Documents. *Proceedings of the ACM SIGCOMM Computer Communication Review*, **26**, 293-305.

[15] Abrams, M., Stanbridge, C., Abdulla, G., Williams, S. and Fox, E. (1995) Caching Proxies: Limitations and Potentials. Boston. http://ei.cs.vt.edu/~succeed/WWW4/WWW4.html

[16] Glassman, S. (1994) A Caching Relay for the World-Wide Web. 1*st International World-Wide Web Conference*, Geneva, 25-27 May 1994, 69-76.

[17] Busari, M. and Williamson, C.L. (2001) Simulation Evaluation of a Heterogeneous Web Proxy Caching Hierarchy. *Proceedings of the IEEE MASCOTS*, Cincinnati, 15-18 August 2001, 379-388.

[18] Fan, L., Cao, P., Almeida, J. and Broder, A. (1998) Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol. *Proceedings of ACM SIGCOMM*'98 *Conference*, Vancouver, September 1998, 254-265.

[19] Mahanti, A., Williamson, C. and Eager, D. (2000) Traffic Analysis of a Web Proxy Caching Hierarchy. *IEEE Network*, **14**, 16-23. http://dx.doi.org/10.1109/65.844496

[20] Yu, P. and MacNair, E. (1998) Performance Study of a Collaborative Method for Hierarchical Caching in Proxy Servers. *Proceedings of World-Wide Web Conference*, Brisbane, 14-18 April 1998, 215-224.

[21] Tewari, R., Dahlin, M., Vin, H. and Kay, J. (1999) Beyond Hierarchies: Design Considerations for Distributed Caching on the Internet. *Proceedings of the* 19*th International Conference on Distributed Computing Systems*, Austin.

[22] Busari, M. and Williamson, C.L. (2001) On the Sensitivity of Web Proxy Cache Performance to Workload Characteristics. *Proceedings of the IEEE INFOCOM*, Anchorage, 22-26 April 2001, 1225-1234.

[23] The Webalizer Home Page. http://www.mrunix.net/webalizer/

[24] Vass, J., Harwell, J., Bharadvaj, H. and Joshi, A. (1998) The World Wide Web: Everything You (N)ever Wanted to Know about Its Servers. *IEEE Potentials*, **17**, 33-37. http://dx.doi.org/10.1109/45.721730

Int. J. Communications, Network and System Sciences, 2014, 7, 529-535
Published Online December 2014 in SciRes. http://www.scirp.org/journal/ijcns
http://dx.doi.org/10.4236/ijcns.2014.712053

Cloud Security: Services, Risks, and a Case Study on Amazon Cloud Services

Patrick Mosca[1], Yanping Zhang[1], Zhifeng Xiao[2], Yun Wang[3]

[1]Department of Computer Science, Gonzaga University, Spokane, USA
[2]Department of Computer Science & Software Engineering, Penn State Erie, Erie, USA
[3]Department of Computer Science and Information Systems, Bradley University, Peoria, USA
Email: zhangy@gonzaga.edu

Received 15 October 2014; revised 26 November 2014; accepted 10 December 2014

Abstract

Recent advances have witnessed the success and popularity of cloud computing, which represents a new business model and computing paradigm. The feature of on-demand provisioning of computational, storage, and bandwidth resources has driven modern businesses into cloud services. The cloud is considered cutting edge technology and it is solely relied on by many large technology, business, and media companies such as Netflix or Salesforce.com. However, in addition to the benefit at hand, security issues have been a long-term concern for cloud computing and are the main barriers of the widespread use of cloud computing. In this paper, we briefly describe some basic security concerns that are of particular interest to cloud technology. We investigate some of the basic cloud concepts and discuss cloud security issues. Amazon Web Services is used as a case study for discussing common cloud terminology. Data security, as well as some cloud specific attacks is introduced. The current state and the future progression of cloud computing is discussed.

Keywords

Could Computing, Security, Amazon, Cloud Storage

1. Introduction

Recent advances have witnessed the success and popularity of cloud computing, which represents a new business model and computing paradigm [1]. The feature of on-demand provisioning of computational, storage, and bandwidth resources has driven modern businesses into cloud services [2]. The cloud is considered cutting edge technology and it is solely relied on by many large technology, business, and media companies such as Netflix or Salesforce.com. However, in addition to the benefit at hand, security issues have been a long-term concern and are the main barriers of the widespread use of cloud computing [1]. There are three main challenges [1] for

How to cite this paper: Mosca, P., Zhang, Y.P., Xiao, Z.F. and Wang, Y. (2014) Cloud Security: Services, Risks, and a Case Study on Amazon Cloud Services. *Int. J. Communications, Network and System Sciences*, **7**, 529-535.
http://dx.doi.org/10.4236/ijcns.2014.712053

building a secure and trustworthy cloud:

- *Outsourcing* reduces both capital expenditure and operational expenditure for cloud customers [1]. However, outsourcing also indicates that cloud customers no longer retain the physical control on hardware, software, and data. To address this challenge, a trustworthy cloud is expected, meaning that cloud customers are enabled to verify the data and computation in terms of confidentiality, integrity, and other security services [1].
- *Multi-tenancy* means that a cloud is shared by multiple customers [1]. Virtualization is heavily used by cloud vendors to optimize resource allocation and management [1]. A common but risky situation is that data belonging to different customers may be stored in the same physical machine. Adversaries can exploit this vulnerability to launch various attacks such as data/computation breach, flooding attack, etc. [1].
- *Massive data and intensive computation* are two other features of cloud computing. Therefore, traditional security mechanisms may not suffice the new security requirements due to unbearable computation or communication overhead [1].

This paper investigates various aspects on cloud security [2]-[4], including data security [5], cloud risks [8] and API concerns [9] [10], cloud services and account hijacking [2]-[14]. The goal of this paper is twofold: first, we focus on the valuable and unique security aspects of the cloud that are different from security issues that widely exist in other computing platforms, since there are certain risks and vulnerabilities only presenting themselves on the cloud environment; second, our intention is to provide an overview of cloud security from the practitioners' point of view. Therefore, we start from Amazon's cloud service [12], and then proceed to discuss the security concerns and the applicable criteria that follow (**Figure 1**).

The rest of this paper is organized as follows: Section 2 presents the background knowledge of Amazon's cloud storage; Section 3 discusses the aspect of data security in cloud; Section 4 investigates other cloud risks and API concerns; Section 5 reviews cloud services and the risk of account hijacking; Section 6 sheds some light on the future of cloud security; Section 7 concludes the paper.

2. Amazon's Cloud Storage

In this section, we will discuss basic technical terms and concepts associated with Amazon's cloud platform. There are different types of storage on Amazon's cloud: AMI (Amazon Machine Image) [15], EBS (Elastic Block Store) [16], snapshots [17], and volumes [16]-[19].

- A volume consists of stored data and possibly empty space. Also, a volume can exist virtually or can consume a full physical hard drive [18].

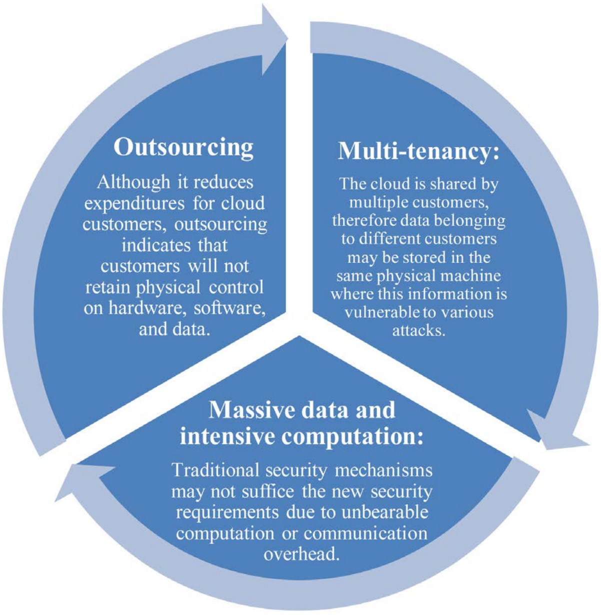

Outsourcing
Although it reduces expenditures for cloud customers, outsourcing indicates that customers will not retain physical control on hardware, software, and data.
Multi-tenancy:
The cloud is shared by multiple customers, therefore data belonging to different customers may be stored in the same physical machine where this information is vulnerable to various attacks.
Massive data and intensive computation:
Traditional security mechanisms may not suffice the new security requirements due to unbearable computation or communication overhead.

Figure 1. Three main challenges in cloud secruity [1].

- A snapshot is simply a backup or copy of an instance's volume data. A snapshot can be used to restore the data on an instance, similar to restoring from a backup. A snapshot is typically not a bootable form of storage [17].
- EBS is a new form of data storage. An EBS is virtual data storage that acts identically to a volume, but the data can be spread across many physical hard drives and can be moved quickly and easily [16]. The motivation behind EBS is to increase storage efficiency in the cloud. Cloud providers can then sell leftover storage to more customers. Additionally, an EBS can consist of multiple volumes, similar to partitions on a drive [16].
- An AMI is an advanced image of a virtual machine that can be used to create one or more instances of that AMI [15]. These images are similar to bootable snapshots that carry additional information about the virtual machine. An AMI is loaded onto an EBS when an instance is created [15]. For example, when a user obtains an instance and sets it up to host his or her website, all he or she needs to do is save the instance as an AMI, copy it to clouds across the world, and then produce duplicate instances of that AMI. All of his or her instances are live, working clones of the original image that are spread throughout regions.

3. Data Security

Cloud customers may store sensitive information in cloud instances. From a security perspective, cloud companies need to ensure the confidentiality of the service [2]. For example, this data could be the backend database for a financial service. A client of any cloud service is supposed to know the risks associated with data security, e.g., data loss and data theft [8]. When storing sensitive information, encryption is always a powerful scheme. Naturally, it would make sense to encrypt sensitive information such as credit card numbers that are stored in the cloud. A potential weakness to encryption in the cloud is the security of the keys. In the hacker world, it is commonly known that physical access to a machine always results in game over. This is because an attacker has control over the machine [2] [5]. Simple passwords on the operating system will not prevent an attacker from stealing data. A break-in is unavoidable unless the full disk is encrypted [8]. Full disk encryption means that the entire volume is encrypted, including the operating system [20]. While full disk encryption is possible in the cloud-computing world, many clients do not encrypt their data for performance and financial reasons. Disk encryption adds additional overhead to the total data stored. Even though data rates vary from region to region, when clients pay by the terabyte, less data is best (see **Table 1**) [3]. Additionally, many large data stores require quick access. For example, a video streaming service needs to read data quickly [3]. Disk encryption will slow this process down significantly and increase business costs. To this end, many cloud customers do not encrypt their volumes.

When cloud customers do not encrypt their volumes, a security risk is presented. A rogue employee of the provider has the power to snoop around without the customer's knowledge. Since the employee has physical access to the costumer's cloud instance, there is nothing to stop the employee from grabbing vital information and any other private keys [2] [8]. This employee can do this simply by cloning the victim's virtual machine, and then running the clone on a second offline hypervisor [5]. The employee can monitor the behavior of the virtual machine and take their time looking for valuable data. The rogue employee can then proceed to steal the data or use the keys to break into more cloud instances. When storing data in the cloud, trust is a very important part of data privacy. "The threat of a malicious insider is well-known to most organizations. This threat is amplified for consumers of cloud services by the convergence of IT services and customers under a single management domain, combined with a general lack of transparency into provider process and procedure" [2]. Therefore, a trustworthy cloud is an essential step toward the success of cloud computing.

A key concern when encrypting data is determining whether or not the encryption software is open source.

Table 1. Amazon storage pricing [3].

	Standard storage	Glacier storage
First 1 TB/month	$0.105 per GB	$0.011 per GB
Next 49 TB/month	$0.090 per GB	$0.011 per GB
Next 450 TB/month	$0.075 per GB	$0.011 per GB
Next 500 TB/month	$0.070 per GB	$0.011 per GB
Next 4000 TB/month	$0.065 per GB	$0.011 per GB
Next 5000 TB/month	$0.060 per GB	$0.011 per GB

Opening encryption software is key to ensuring that no back doors or additional keys are created [1]. This has become a major problem for many services such as text messaging, videoconferencing, and email. For example, Apple has a service called, “iMessage” that handles text messages in the cloud. All messages are encrypted end-to-end, ensuring that no middleman can read your conversations [4]. What Apple does not tell you is that they are legally required to keep a copy of the key. Again customers are putting trust in the provider, Apple.

4. Cloud Risks and API Concerns

4.1. General Server Risks

Of all the risks being reported by the news and blogs on the Internet, many of them are not risks inherent to cloud services, meaning they would apply to all servers. Although, the cloud does increase the risk of some of them (**Figure 2**):

- Denial of Service (DoS) [5] being of the latter is obviously always an issue for servers. The added risk to using the cloud is that attacks on other users of the cloud would affect your portion. If an attack on the cloud unrelated to you brought it down it would also bring your server down or at least slow it down [5]. So while your server may not be the target of attacks, consideration needs to be added which include the notion that you may be working on the same hardware with anyone.
- Data breaches have greater potential of disaster on the cloud. A single flaw in a cloud service could cause one data breach to extend to a breach of the entire system [2]. Methods more simple than side channeling could extract keys or gather unencrypted data. While some individuals think that it is a considerable risk of cloud computing, it is in fact more realistically less of a risk than it would be to create one’s own server and service it [8]. In the latter case there are many precautions to be taken, which have already been implemented by cloud services.
- Data loss is an issue not unique to the cloud. Power loss is a potential scenario everywhere on Earth and sometimes unavoidable [8]. Articles have defamed cloud services for losing data when in reality those servers probably have better surge and outage protection than you could afford [14].
- The risk of giving other access to your server’s internals and secrets is once again almost unavoidable [2] [5]. Unless you were to buy, setup, and implement your own server in your home you will probably have to trust someone else to help you, thereby risking the data’s integrity. It would be unwise to attempt to secure grand amounts of money on the cloud for this reason; even on your own server the temptation would exist for the valuables to be stolen [2]. Perhaps an employee would risk their job and reputation for a chance at this money or perhaps the cloud service has taken precautions against employees gaining too much valuable information. This much is unclear and unreported by cloud service businesses. Nonetheless if looking toward using a cloud one should remember that risks surround every server and the most important question is: would you do the extra work for the extra security?

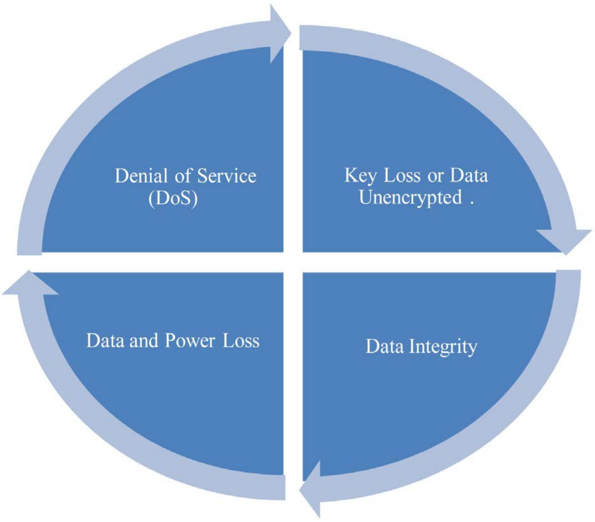


Figure 2. Four general server risks in the cloud [5] [8].

4.2. API Keys

Application Programming Interface (API) Keys [19] on the cloud were first used solely as the identifier for client programs running on a cloud. This allowed for the management of client programs and users to be monitored so as to backtrack events and log usage. While initially this had no security issues involved, later progress on cloud infrastructure has expanded the use of these keys [2]. In some cases it has been reported that these keys are used for authorization. Thus having this key gives one the power to alter delete or transfer an account's data or to use the servers for any other purpose, which would then be traced back and billed to the account holder [2]. After these keys became security risks the major problem was that they were not treated like them. Developers would email them around and store them in their hard drives, where snooping and sniffing could find them.

Years ago Google and Yahoo were making this mistake, but it was not long until the risks were found. They have since bulked their authorization security using Security Assertion Markup Language [21], and hashed-based authentication codes [22]. Yet the issue remains a threat as developers fail to follow best practices and continue to use API Keys for security purposes [2]. The older, more experienced businesses like Yahoo, Google, and Amazon have all either fallen into this trap before or are aware of the faults present. These companies can be trusted to build better software and control data flow than startups. If API Keys are going to secure information, they need to be handled with greater care.

4.3. APIs

Application Programming Interfaces or APIs, give what is almost a roadmap into how an application works [9] [10]. They are usually treated securely but not often enough. The University of Texas at Austin and Stanford University examined several commonly used web services [10]. Payment services at several of them were found to have vulnerabilities in the Secure Sockets Layer (SSL) protocol when accessed through APIs not meant for a browser [3]. Taking advantage of this flaw led to getting access to a user's files. Applications like Chase Mobile Banking and Instagram failed to implement SSL with complete security [10].

5. Service and Account Hijacking

At this point in its development, the cloud is seriously at risk for service and account hijacking [2]. This entails the unauthorized access to and use of the accounts and services of clients who utilize the cloud. This hijacking can happen any number of ways—since the cloud is simply a network run on many different servers, it is vulnerable to all the same attacks as both networks and servers [2].

Once an attacker has hijacked a service or account, he or she may be able to eavesdrop on the activities of the authorized users, impersonate authorized users, tamper with the network data, or utilize the service or account to propagate malware, e.g. by redirecting clients to malicious websites—all the threats typical for non-cloud networks and servers [2]. Unique to the cloud, however, the attacker may use the hijacked service or account as a base of operations to perform further attacks on other machines in the cloud [2] [5].

5.1. Recent Examples

In recent years, one of the companies on the vanguard of cloud technology—Amazon.com, Inc.—fell prey to such an attack. In 2010 hijackers performed a cross-site scripting (XSS) attack on some site to gain its credentials, and were successful [23]. The attackers then infiltrated the Amazon Relational Database Service (RDS) [7] such that, even if they lost their original access, they would still have a backend into the Amazon system. From that point on, they could capture the login information of anyone who clicked the login button on the Amazon homepage.

The attackers used their servers to infect new machines with the Zeus Trojan horse [23] and control machines already infected with it (Zeus is a piece of malware designed for Windows most often used for stealing bank information through form-grabbing and password-logging via a man-in-the-browser attack [24] [25]). Computers infected with the malware began to report to Amazon's EC2 for updates and instructions [23].

One of the most interesting facts about this case was that it was not, strictly speaking, Amazon's fault. The attackers gained access through some other, more vulnerable domain [23]. This reveals one truth about the cloud: on it, even one vulnerable system may lead to the compromising of the whole network. Furthermore, Amazon was only one of several sites to suffer this type of attack in the period of just a few months, and it was not in bad company: Twitter, Google's app engine, and Facebook all experienced similar threats [23].

5.2. Possible Defenses

To prevent this type of breach, the Cloud Security Alliance (CSA) admonishes organizations to disallow users and services from sharing account credentials between themselves, and in addition to employ multi-factor authentication requirements when feasible [2]. However, both these changes may make systems more difficult to use, more expensive, and slower. Multi-factor authentication [26] is authentication demanding at least two of the following: knowledge, or something one knows; possession, or something one has; and inference, or something one is. Thus, multi-factor authentication places much more of a burden on users and services than single-factor authentication. And if users and services are disallowed from directly sharing credentials, cloud service providers may have to construct secure channels (an expensive undertaking) or hire a third party for communication between users and services (likewise expensive) [26].

6. The Future of Cloud Security

6.1. PRISM Scandal

In June 2013 Edward Snowden revealed that the National Security Agency (NSA) has been collecting enormous amounts of communication and search data from internet companies such as Microsoft, Yahoo, Google, and many more, including data about the activities of American citizens [27]. Snowden also explained that even low-level NSA employees have the ability to access this data without warrants. Such surveillance has taken place since January 2007. It may not be immediately clear why this information is particularly relevant to the cloud. The government can force cloud service providers to install backdoors in their hypervisors, but it can do the same for operating systems and even individual machines [11]. However, targeting the machine of one individual is much less likely, since at that point the government has singled out that user specifically. Instead, the cloud provides the NSA with a brimming ocean of network activity, in which it can cast its net and hope to catch something of use—much more efficient than targeting individual machines. As one writer for Porticor said: "Scanning all the data from a cloud provider is relatively easy, because massive amounts of data from multiple owners is all available" [11]. Porticor recommends that users encrypt their own data to combat such invasions of privacy, but it is doubtful that such a solution will ever prove widely acceptable, seeing as it places undue responsibility on users and requires a degree of expertise. The example of PRISM [27] touches on many issues within the future of cloud security: maintenance of privacy, government policy, and data theft (since attackers may capture user data using NSA techniques, or even the NSA channels themselves). These issues are not often considered by users of cloud services, and are not being discussed on a large scale.

6.2. A Better Cloud

There are organizations working towards a more secure cloud, such as the CSA [2]. Another is Silver Sky, an expert provider of cloud security and provider of "the industry's only advanced Security-as-a-Service platform from the cloud" [13] [28]. The CTO of Silver Sky, Andrew Jaquith, explains that many CIOs are moving their services to the cloud in order to save money, but that security remains a key concern and these moves may be insecure or at least hasty. But on the other hand, he also explains that many cloud service providers are becoming clearer, more transparent, and more assured than ever before that they could protect customer data [13].

Thus, the move to the cloud, while it may in some ways be insecure, does not herald anyone's doom. And, with its ever-increasing popularity, even hesitant companies may not soon have a choice.

7. Conclusion

In this paper, we provide an overview of cloud security in various aspects. We first review the data storage scheme for Amazon's cloud. The unique forms of products and services offered through cloud services show the incentive for modern business use. Using Amazon Web Services [12] as a case study, we are able to implore some of the basic terms and concepts of cloud computing. We then proceed to discuss data security, API concerns, account hijacking, and other security concerns. These general concerns are shown to be of particular interest to cloud security. The main differences between traditional services and cloud services are compared from a security perspective. Service and account hijacking is covered, as well as possible defenses. We investigate differences between security issues in cloud services and in traditional services. From the practitioners' view, we briefly overview the security in cloud. The study in this paper provides a guideline of research on cloud services and security issues. Finally, we give some ideas on how to build a more secure cloud. Our future work will focus on the security concerns in cloud services. It will include the privacy protection of data information stored in cloud, data integrity with multiple backups for services purpose, etc.

References

[1] Xiao, Z. and Xiao, Y. (2013) Security and Privacy in Cloud Computing. *IEEE Communications Surveys & Tutorials*, **15**, 843-859.

[2] Cloud Security Alliance (2010) Top Threat to Cloud Computing. https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf

[3] Amazon: Amazon Glacier. http://aws.amazon.com/glacier/

[4] Quarks Lab (2013) iMessage Privacy. http://blog.quarkslab.com/imessage-privacy.html

[5] Mutch, J. (2010) How to Steal Data from the Cloud. http://www.cloudbook.net/resources/stories/how-to-steal-data-from-the-cloud

[6] Yorozu, Y., Hirano, M., Oka, K. and Tagawa, Y. (1982) Electron Spectroscopy Studies on Magneto-Optical Media and Plastic Substrate Interface. *IEEE Translation Journal on Magnetics in Japan*, **2**, 740-741.

[7] Amazon: Service Level Agreement. http://aws.amazon.com/ec2-sla/

[8] Kirchgaessner, S. (2013) Cloud Storage Carries Potent Security Risk. http://www.ft.com/cms/s/0/4729ed7c-3722-11e3-9603-00144feab7de.html

[9] Lemos, R. (2012) Insecure API Implementations Threaten Cloud. http://www.darkreading.com/cloud/insecure-api-implementations-threaten-cl/232900809

[10] Lemos, R. (2013) Vulnerable APIs Continue to Pose Threat to Cloud. http://www.darkreading.com/services/vulnerable-apis-continue-to-pose-threat/240146453

[11] Porticor Cloud Security (2013) Did Snowden Compromise the Future of Cloud Security? http://www.porticor.com/2013/07/cloud-security-snowden/

[12] Amazon: Amazon Web Services. http://aws.amazon.com

[13] SilverSky (2013) The Future of Cloud Computing and the Latest Security Threats. https://www.silversky.com/blog/the-future-of-cloud-computing-and-the-latest-security-threats

[14] Columbia University (2012) Fog Computing: Mitigating Insider Data Theft Attacks in the Cloud. http://www.cs.columbia.edu/~angelos/Papers/2012/Fog_Computing_Position_Paper_WRIT_2012.pdf

[15] Amazon: Amazon Machine Image (AMI). http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html

[16] Amazon: Amazon EBS. http://aws.amazon.com/ebs/

[17] Amazon: Amazon EBS Product Details. http://aws.amazon.com/ebs/details/#snapshots

[18] Amazon: Amazon EC2 Instance Store. http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html

[19] MailChimp (2014) About API Keys. http://kb.mailchimp.com/accounts/management/about-api-keys

[20] Janssen, C. Full-Disk Encryption (FDE). http://www.techopedia.com/definition/13623/full-disk-encryption-fde

[21] Cover, R. (2010) Security Assertion Markup Language (SAML). http://xml.coverpages.org/saml.html

[22] United Sates Department of Veterans Affairs (2014) Keyed-Hash Message Authentication Code (HMAC). http://www.va.gov/trm/StandardPage.asp?tid=5296

[23] Goodin, D. (2009) Zeus Bot Found Using Amazon's EC2 as C&C Server. http://www.theregister.co.uk/2009/12/09/amazon_ec2_bot_control_channel/

[24] Nahorney, B. and Nicolas, F. (2010) Trojan.Zbot. http://www.symantec.com/security_response/writeup.jsp?docid=2010-011016-3514-99

[25] Acunetix: Cross Site Scripting Attack. https://www.acunetix.com/websitesecurity/cross-site-scripting/

[26] Amazon: Multi-Factor Authentication. http://aws.amazon.com/iam/details/mfa/

[27] The Guardian: The NSA Files. http://www.theguardian.com/world/the-nsa-files

[28] SilverSky (2013) About Us. https://www.silversky.com/about-us

Call for Papers

ISSN: 1913-3715 Volume 7, Number 12, December 2014
Scientific Research
International Journal of
Communications, Network and System Sciences
Outsourcing
Although it reduces expenditures for cloud customers, outsourcing indicates that customers will not retain physical control on hardware, software, and data.
Multi-tenancy:
The cloud is shared by multiple customers, therefore data belonging to different customers may be stored in the same physical machine where this information is vulnerable to various attacks.
Massive data and intensive computation:
Traditional security mechanisms may not suffice the new security requirements due to unbearable computation or communication overhead.
ISSN: 1913-3715
www.scirp.org/journal/ijcns

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

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

IJCNS is an international refereed journal dedicated to the latest advancement of communications and network technologies. The goal of this journal is to keep a record of the state-of-the-art research and promote the research work in these fast moving areas.

Editor-in-Chief

Prof. Boris S. Verkhovsky New Jersey Institute of Technology, USA

Subject Coverage

This journal invites original research and review papers that address the following issues in wireless communications and networks. Topics of interest include, but are not limited to:

- Ad Hoc and Mesh Networks
- Coding, Detection and Modulation
- Cognitive Radio
- Communication Networks Architecture Design
- Communication Reliability and Privacy
- Communication Security and Information Assurance
- Cooperative Communications
- Embedded Distributed Systems
- Global Networks
- Heterogeneous Networking
- Microprocessor
- MIMO and OFDM Technologies
- Network Protocol, QoS and Congestion Control
- Network Survivability
- Next Generation Network Architectures
- Reconfigurable Networks
- Resource Management
- Satellite Communication
- Sensor Networks
- Simulation and Optimization Tools
- UWB Technologies
- Wave Propagation and Antenna Design
- Wireless Personal Communications

We are also interested in short papers (letters) that clearly address a specific problem, and short survey or position papers that sketch the results or problems on a specific topic. Authors of selected short papers would be invited to write a regular paper on the same topic for future issues of the IJCNS.

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 **E-Mail: ijcns@scirp.org**

Advances in Alzheimer's Disease
Advances in Chemical Engineering and Science
Advances in Aging Research
Atmospheric and Climate Sciences
Journal of Biomaterials and Nanobiotechnology
Journal of Electromagnetic Analysis and Applications
Journal of Electronics Cooling and Thermal Control
Journal of Modern Physics
Journal of Service Science and Management
Open Journal of Mass Spectrometry
Journal of Environmental Protection
Journal of Software Engineering and Applications
Advances in Applied Sociology
Art and Design Review
Advances in Reproductive Sciences
Creative Education
Health
Low Carbon Economy
Psychology
JBiSE
Surgical Science
Natural Science
Spectral Analysis Reviews
Open Journal of Business and Management
Advances in Bioscience and Biotechnology
Advances in Entomology
Applied Mathematics
Engineering
Intelligent Control and Automation
Modern Economy
Open Journal of Chromatography
Advances in Biological Chemistry
Advances in Historical Studies
American Journal of Plant Sciences
Energy and Power Engineering
Modeling and Numerical Simulation of Material Science
Open Journal of Accounting
Soft