IJBDCN publishes special issues on the recent trends in business data communications and networking and related issues. A short proposal containing the title of proposed special issue, tentative list of invited authors, suggested date of publication and a call-for-papers should be sent to j.gutierrez@auckland.ac.nz for review.

**Special Issue Proposal**

**Idea Group Publishing**

**http://www.idea-group.com**

**Hershey • London • Melbourne • Singapore**

Tel: 717/533-8845 • Fax: 717/533-8661 • E-mail: cust@idea-group.com

IJBDCN is listed in the following indexes: GetCited; MediaFinder; Standard Periodical Directory; Ulrich’s International Periodicals Directory
# Table of Contents

**Editorial Preface: International Journal of Business Data Communications and Networking**  
Jairo Gutierrez, Editor-in-Chief

## Reviewed Papers

### 1 Prototype Implementation of a Proxy Caching System for Streaming Media Objects
Guo Hui, Huazhong University of Science and Technology, Wuhan, China  
Zhou Jingli, Huazhong University of Science and Technology, Wuhan, China  
Wang Yunpeng, Huazhong University of Science and Technology, Wuhan, China  
Yu Shengsheng, Huazhong University of Science and Technology, Wuhan, China  
In this paper, the authors focus on the proxy caching problem specifically for multimedia streaming objects. A prototype design and implementation of a proxy caching system – HUSTProxy is proposed.

### 17 WebGuard: Web Adult Content Detection and Filtering System
Mohamed Hammami, Ecole Centrale de Lyon, France  
Liming Chen, Ecole Centrale de Lyon, France  
This paper describes a Web filtering system “WebGuard,” which aims to automatically detect and filter adult content on the Web. WebGuard uses data mining techniques to classify URLs into two classes: suspect URLs and normal URLs.

### 33 The Use of Efficient Cost Allocation Mechanisms for Congestion Pricing in Data Networks with Priority Service Models
Fernando Beltrán, University of Auckland, New Zealand  
Cesar García-Díaz, University of Groningen, The Netherlands  
This paper demonstrates the application of two efficient cost allocation mechanisms on a simple FIFO network model in order to find congestion-based prices for network access services. It is shown that it is possible to obtain congestion-dependent prices in data networks (such as the Internet), establishing simultaneously different service quality levels among the users.

### 50 Mining Parallel Patterns from Mobile Users
John Goh, Monash University, Australia  
David Taniar, Monash University, Australia  
In this paper, the author’s performance evaluation shows that the method for mobile parallel pattern mining suits for the real world problem in both the physical and logical parallel pattern mobile mining paradigm.
Mission: This journal disseminates practical and theoretical information, which will enable readers to understand, manage, use and maintain business data communication networks more effectively. For more information, visit www.idea-group.com/ijbdcn.

Coverage: The journal covers both guided and wireless communications of voice, data, images and video and their impact on the organisations in which they are used. The journal aims to address the key issues for businesses utilising data communications and the increasing importance of networking technologies in business. Contributing papers may deal with, but are not limited to the following topics: strategic use of networking technologies; outsourcing of networking and data communication services; business re-engineering issues associated with networking; cross-border network-based information systems; the effect of legislation and regulation on telecommunications; business applications of telecommunications; network pricing issues and strategies; regulatory models, issues and their impacts; design and performance evaluation of new network application and systems; management of telecommunications in organizations; mobility and m-commerce issues; frameworks for wireless security; management of business data communications projects; standards and network interoperability issues; impact of the Open Source movement on business data communications; security of interconnected systems (procedures, implementation, mechanisms, protocols and compatibility); policy-based network management systems; adoption and diffusion of networking technologies; and more (www.idea-group.com/ijbdcn).

Originality: Prospective authors should note that only original and previously unpublished manuscripts will be considered. Furthermore, simultaneous submissions are not acceptable. Submission of a manuscript is interpreted as a statement of certification that no part of the manuscript is copyrighted by any other publication nor is under review by any other formal publication. It is the primary responsibility of the author to obtain proper permission for the use of any copyrighted materials in the manuscript, prior to the submission of the manuscript.

Style: Submitted manuscripts must be written in the APA (American Psychological Association) editorial style. References should relate only to material cited within the manuscript and be listed in alphabetical order, including the author’s name, complete title of the cited work, title of the source, volume, issue, year of publication, and pages cited. Please do not include any abbreviations. See the following examples:

Example 1: Single author periodical publication.

Example 2: Multiple authors periodical publication.

Example 3: Books.

State author’s name and year of publication where you use the source in the text. See the following:

Example 1: In most organizations, information resources are considered a major resource (Brown, 1988; Smith, 2002).

Example 2: Brown (2002) states that the value of information is recognized by most organizations.

Direct quotations of another author’s work should be followed by the author’s name, date of publication, and the page(s) on which the quotation appears in the original text.

Example 1: Brown (1989) states that “the value of information is realized by most organizations” (p. 45).

Example 2: In most organizations, “information resources are considered to be a major organization asset” (Smith, 2002, pp. 35-36) and must be carefully monitored by the senior management.

For more information please consult the APA manual.
**Review process:** To ensure the high quality of published material, IJBDCN utilizes a group of experts to review submitted manuscripts. Upon receipt of the manuscript, two reviewers are selected from the Editorial Review Board of the journal. The selection is based upon the particular area of expertise of the reviewers, matched to the subject matter of the submission. An additional ad-hoc reviewer is also selected to review the manuscript. Therefore, each submission is accordingly blind reviewed by at least three reviewers. Revised manuscripts will be reviewed again by the original review panel with the addition of one new reviewer. Return of a manuscript to the author(s) for revision does not guarantee acceptance of the manuscript for publication. The final decision will be based upon the comments of the reviewers, upon their second review of the revised manuscript.

**Copyright:** Authors are asked to sign a warranty and copyright agreement upon acceptance of their manuscript, before the manuscript can be published. All copyrights, including translation of the published material into other languages are reserved by the publisher, Idea Group Inc. Upon transfer of the copyright to the publisher, no part of the manuscript may be reproduced in any form without written permission of the publisher.

**Submission:** Authors are asked to submit their manuscripts for possible publication by e-mail as a file attachment in Microsoft Word or RTF (Rich Text Format) to j.gutierrez@auckland.ac.nz. The main body of the e-mail message should contain the title of the paper and the names and addresses of all authors. Manuscripts must be in English. The author’s name should not be included anywhere in the manuscript, except on the cover page. Manuscripts must also be accompanied by an abstract of 100-150 words, precisely summarizing the mission and object of the manuscript.

**Length:** The length of the submitted manuscript is not specifically limited, however, the length should be reasonable in light of the chosen topic. Discussion and analysis should be complete but not unnecessarily long or repetitive.

**Correspondence:** The acknowledgment letter regarding the receipt of the manuscript will be promptly sent. The review process will take approximately 8-16 weeks, and the author will be notified concerning the possibility of publication of the manuscript as soon as the review process is completed. All correspondence will be directed to the first author of multi-authored manuscripts. It is the responsibility of the first author to communicate with the other author(s). Authors of accepted manuscript will be asked to provide a final copy of their manuscript in either Word or RTF text format stored on a 3 1/2” disk, zip disk or CD-ROM, accompanied by a hardcopy of the manuscript and the original signed copy of the Warranty and Copyright Agreement. The accepted manuscript will be edited by the Journal copyeditor for format and style.

**Book Reviews:** IJBDCN invites prospective book reviewers to submit their review of either textbooks or professional books for possible inclusion in the journal. Reviewers should focus on the following guidelines when developing the book review:
- Book reviews must not exceed 1,500 words
- Reviews should summarize the book and indicate the highlights, strengths, and weaknesses of the book.
- Reviews should evaluate the organizational and managerial applications of the material discussed in the book relevant to information resources and technology management.
- Reviews should critique and constructively evaluate the author’s work and not merely list the chapters' contents.
- The writing style, accuracy, relevance, and the need for such a work in the discipline should be analyzed.
- The review must include the title of the book, author, publishing company, publication date, number of pages, cost (if listed), and ISBN number.
- Each submission must be accompanied by a short biography of the reviewer.

**Case Studies:** The IJBDCN encourages submissions of case studies based on actual scenarios related to practice-based issues and practical applications of business data communications and networking. Case studies must not exceed 2000 words and must provide adequate information regarding the educational environment upon which the study is based, presentation of the issues involved, coverage of any experiments or techniques involved, and elaborations of the lessons learned or conclusions drawn from the study.

---

**All submissions and questions should be forwarded to:**

Jairo Gutierrez, Editor-in-Chief, IJBDCN
University of Auckland, New Zealand
j.gutierrez@auckland.ac.nz
Editorial Preface:
International Journal of Business Data Communications and Networking

Jairo Gutierrez, Editor-in-Chief

INCLUDED IN THIS JOURNAL

When we announced the mission of this new journal we stated that “Data Communications and Networks are no longer peripheral issues best left to the IT team, they are integral components of any modern information system and in many cases are essential to the success or failure of the organisation.” This has been the main motivation for the launching of the International Journal of Business Data Communications and Networking. We felt there was a need to disseminate research focused on the range of technologies, processes and ideas that enable us to enjoy the many benefits of modern data communications and computer networks. Furthermore, we wanted to assist our readers in understanding, managing, and using business data communication networks more effectively.

This inaugural issue contains a selection of interesting contributions with which we start tackling some important contemporary networking challenges. In the first paper, Prototype Implementation of a Proxy Caching System for Streaming Media Objects, Guo Hui, Zhou Jingli, Zeng Dong and Yu Shengsheng introduce a cache system used to facilitate the streaming of real-time multimedia traffic, which is an increasingly popular requirement for many modern networked applications. In the second paper, WebGuard: Web Adult Content Detection and Filtering System, Mohamed Hammami and Liming Chen deal with the challenges of detecting and filtering adult content when using the Web, a very topical issue given the widespread use of these technologies by children everywhere. The third paper, The Use of Efficient Cost Allocation Mechanisms for Congestion Pricing in Data Networks with Priority Service Models, by Fernando Beltrán and Cesar García, unveils a mechanism which allows users to send data over networks that can guarantee different levels of quality of service at different price levels. A pricing technique using efficient cost allocation methods is presented. This paper is of special interest to Internet Service Providers attempting to offer premium services to subscribers and to businesses requiring different levels of Internet service rather than the current best-effort (or send-and-pray) model. Finally, in the last paper, Mining Parallel Patterns from Mobile Users, John Goh and David Taniar discuss the fairly new field of mobile data mining and they propose methods that can be used to discover useful patterns from mobile users. These techniques can aid businesses planning mobile commerce offerings.
It is important to note here that you will not be reading this inaugural issue without the tremendous help our editorial team received from Idea Group Inc. They had the patience and experience to guide this process to fruition. Thanks!

Best regards,

Jairo Gutierrez, Editor-in-Chief

Jairo Gutierrez has expertise in networking and data communications. He has worked in industry as a research and development manager, systems integration consultant, and information systems manager. He regularly conducts seminars on networking technologies and their use. His current research topics are in network management systems, viable business models for mobile commerce, programmable networks, and Quality of Service issues associated with Internet protocols. Jairo is a senior lecturer in IS at the University of Auckland and coordinator of its Cisco Networking Academy Program. He teaches data communications and computer networking. He has supervised more than 30 post-graduate students during the last seven years with research projects covering a wide range of networking technologies issues. He received a Systems and Computer Engineering degree from The University of The Andes (Colombia, 1983), a master’s degree in computer science from Texas A&M University (1985), and a PhD (1997) in information systems from The University of Auckland (New Zealand).
Prototype Implementation of a Proxy Caching System for Streaming Media Objects

Guo Hui, Huazhong University of Science and Technology, Wuhan, China
Zhou Jingli, Huazhong University of Science and Technology, Wuhan, China
Wang Yunpeng, Huazhong University of Science and Technology, Wuhan, China
Yu Shengsheng, Huazhong University of Science and Technology, Wuhan, China

ABSTRACT

Existing techniques for caching Web objects are not appropriate for the multimedia streaming service. In this paper, the authors focus on the proxy caching problem specifically for multimedia streaming objects. A prototype design and implementation of a proxy caching system – HUSTProxy is proposed. The main contribution of HUSTProxy is its ability of partial video caching, sending rate control and providing a more rational prefix caching size determination. These techniques are implemented in our design and are discussed in detail in this paper.

Keywords: multimedia transfer; partial video caching; prefix caching; proxy caching; startup latency; streaming media

INTRODUCTION

With the fact that streaming applications consume network bandwidth along the client-server path for the entire session, the traditional client-server architecture for streaming continuous media objects could not scale to a large number of clients. To achieve scalability and deliver high quality streams, multimedia content should be maintained close to interested clients. Proxy caching is a client-oriented solution for large-scale delivery of high quality streams over the Internet. A proxy cache stores recently accessed resources in the hope of satisfying future client requests without contacting the server, which in turn reduces the load on the network and server, and also accommodates the scalability. Furthermore, since a proxy is located close to its clients, caching of popular streams at a proxy can effectively avoid network bottleneck and then substantially reduces
service response time. Figure 1 shows the connection diagram with server-proxy-client architecture over the Internet. Most recent research on proxy caching focus on handles generic Web objects, such as Harvest (Chankhunthod, 1996) and Squid. However, the existing techniques for caching Web objects are not appropriate for the continuous streaming media services because video files are usually much larger than other Web documents. Compared to Web caching techniques, evaluation of proxy caches for multimedia streaming objects is still immature; design and evaluation of multimedia proxy caching mechanisms clearly require substantially more investigation.

This paper presents the design and implementation of a multimedia proxy caching system, which contributes to reduce server load, network load and service start-up latency, named as HUSTProxy. The name HUST is come from the abbreviation of our university. Although the design is fairly flexible to accommodate control and data transfer protocols used by different vendors, we have implemented our proxy system using RTSP (Real Time Streaming Protocol) (Schuluhrinne, 1998) as the control setup protocol and RTP (Realtime Transport Protocol) (Schulzrinne, 1996) as the data transport protocol. Different from earlier works, we emphasize partial caching and partial replacement techniques. Because the video and audio objects are too larger compared to other Internet objects, storing the entire contents of several long streams would exhaust the capacity of a conventional proxy cache. The instead way is caching just a portion of each requested stream. In particular, we propose that proxy caches only a fix set of frames at the beginning of each popular stream, and the replacement scheme is from the end portions of an unpopular stream. Combining with prefetch techniques implemented in our design, the proxy caching disk space can be dramatically decreased. In addition, due to the unpopular steam has not been replaced entirely, which would also enhance the hit ratio for user request. Another contribution of server-proxy-client architecture is its ability to adjust the sending rate to client based on the available bandwidth between the proxy and interested clients’ path. Our study complements this effort to perform rate control; the effectiveness of this proposal is validated through experimental results from prototype implementation. As expected, by increasing the data sending rate to requested clients, the service start-up latency can be improved accordingly. Additionally, we propose a new definition of popularity of a cached stream compared with earlier work in this paper.

The rest of this paper is organized as follows. The next section gives a briefly reviews of related work. The third section discusses the main issues in the design of multimedia streaming proxy, and presents our solutions. The fourth section describes the design and architecture of our proxy caching system. The fifth section validates our implementation through various experiments, and presents the preliminary results. In the last section, we present the conclusions and directions for future work.

RELATED WORK

Caching techniques have been widely used for traditional Web content such as HTML pages and image files (Chankhunthod, 1996; Abrams, 1995). With the emergence of streaming media applications on the Internet, interest in effective streaming media delivery techniques has increased dramatically.
Different with Web caching schemes, storing the entire media object in the proxy may be inefficient if mostly small portions of very large media objects are accessed. The first intuition is to cache portions of the media object. Some early work on the storage for media objects can be found in David (1992) and Tobagi (1993). Two typical types of partial caching have been investigated. Sen (1999) presents a prefix caching which stores only the first part of the popular media object. It’s similar to our approach. But the challenge of determination of the prefix size is not clearly presented. We propose a solution by considering caching efficiency in calculating the appropriate prefix length. Alternatively, media objects can be cached in segments. Wu et al. (Wu et al., 2001) use segments with exponentially incremental size to model the fact that clients usually start viewing a streaming media object from the beginning and are more and more likely to terminate the session toward the end of the object. A combination of fixed length and exponential length segment-based caching method is considered in the RCache and Silo project (Chae, 2002). Rejaie (2000) considered layered-encoded multimedia streams. The proxy attempts to replay a quality-variable cached stream while performing quality adaptation dynamically. This mechanism is complementary to our approach and can be easily accommodated in our system. Work in Hoffmann (1999) presents a caching architecture for multimedia streams, called SOCCER. SOCCER consists of a self-organizing and cooperative group of proxies. Acharya (1999) also proposed cooperative caching techniques named MiddleMan, which utilize the aggregate storage of client machines. Video streams are stored across multiple proxies where they can be replaced at a granularity of a block. However, this work has not implemented fast-forward/rewind play for clients. We have supported VCR functionality in our HUSTProxy prototype. Several studies are focused on streaming media workload characterization (Almeida, 2001; Chesire, 2001) as well as buffer caching techniques (Dan, 1993; Chen, 2003).

To the best of our knowledge, none of these works has mentioned how to determine the prefix size, and usually treat access frequency as stream...
popularity. It’s not appropriate for popularity definition. Since users are likely to just view the opening of a stream and then close it, maybe the stream is not an interesting object to the client. Some of the works have mentioned sending rate control, but they do not give a detailed discussion and experimental results.

**PROBLEM DISCUSSION**

**Proxy Prefix Size**

Because the size of a streaming object is usually very large, given finite disk capacity, only a limited number of movies can be stored in the cache. This decreases hit ratio and efficiency of the caching system. Storing an entire movie that will probably not be accessed in the future is an exhausted way for video caching. It would be natural to divide movies into segments and turn to partial storage caching techniques to retrench the disk space. As discovered in Acharya (1999), users are far more likely to view the opening of a movie than play it back until its end. Based on this observation, we propose that proxy caches should store a fixed set of beginning frames of each stream, which is named prefix caching. Similar to traditional Web caching, storing a duration of beginning frames also enables the proxy to reduce client delay without sacrificing quality.

Another question is put forward, that of how much of the prefix size is the most appropriate. To resolve this problem, we propose the solution that the prefix size varies dynamically according to popularity of the stream. That is, if the amount of data requested by clients increases, proxy caches more data for that stream. On the other hand, if it decreases, proxy caches less data. The objective of this scheme is to maintain the cache space of each stream to be proportional to its popularity. With this caching scheme it achieves a more efficient utilization of cache disk space, which in turn achieves higher hit ratio. For realizing this scheme, a term of popularity is defined as the total amount of data of a particular stream played back by clients during an interval. In order to mark the popularity of each cached stream, we establish a linked list in memory for each stream, and each element of the linked list we called a CacheItem. When the variation of popularity of a CacheItem has occurred, the CacheItem would change its position to the according location in the linked list. For the linked list, we define that the head of the list is the most popular stream and the tail of list records the most unpopular stream. Figure 2 shows the stream caching pattern with the linked list.

Storing and evicting a stream is a segment-based fashion. A segment consists of a group of contiguous packets and each segment has approximately equal size. The HUSTProxy treats a segment as an atomic unit, that is, all packets of a segment are cached or replaced together. Consider a stream s is divided into equal-size segments, seg0, seg1, seg2, .... Currently seg0~seg2 are cached. If the popularity of stream s increases, segments after seg2 will be cached from seg3. On the contrary, if the popularity of stream s decreases, segments will be evicted from the end of the stream, that is, s will be evicted for seg2 to seg0 gradually. The evicted size based on the extent of the popularity variation. The detailed replacement policy is discussed in the third section.
Packets Recombination

When the user requests a video in the cache, it is served by sending to them the portion of the video locally present in cache, while obtaining the remainder frames from the streaming server and transparently passing it on to the client. In order to achieve prefix caching, video servers and streaming protocols must support random access, and the proxy caching system must incorporate prefetch technique. For implementing prefetching function, we could use the RTSP PLAY message with a modified Range header forward to server, which based on how much data is available from the local disk, for example, Range: 20-End. The number 20 means that the last cached time of prefix caching.

Considering that the cache might not have the entire media object in its local disk, the proxy have to compose a coherent stream by fetching the prefix data from the local disk and fetching the remainder of stream from the server. Since the RTP packets obtained from a particular source begin with a random base timestamp and a random base sequence number, and a uniform stream should have a unique SSRC (synchronizing source identifier) field, another request such as prefetch the remainder of data from server will have another random start timestamp and sequence number, and has a new different SSRC. RTP headers of streamed from server need to be modified by the proxy to consistent with the prefix cached RTP data. This is a challenge in the implementation of prefetching. Figure 3 shows an example of how to compose RTP streams from two different sources into a consistent one. Here, we consider a media clip which is 15 seconds long: the initial 5 seconds are cached in proxy while the rest of the 10 seconds data is obtained from a remote server; assuming the stream clockrate is 90000, which can be obtained from SDP (Handley, 1998) feedback from the server. Through the timestamp of the last RTP packet, base timestamp and the stream clockrate, we can obtain the last cached time with the metric of seconds by EQ(1):

\[
\text{CachedTime(seconds)} = \frac{\text{LastTimestamp} - \text{BaseTimestamp}}{\text{Clockrate}}
\]  

As shown in Figure 3, 180000 and 630000 are the timestamps of the first and last RTP packets obtained from the disk; \(\omega\) is the timestamp of the next packet that has not been cached on the disk, shown as a shaded box. And we have known that the timestamp gap between two successive RTP packets is 6000. Assume -376000 and -16000 are the timestamps of the first and last RTP packets obtained from the network, and we also assume that the data

---

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.
in packet with timestamp \( \omega \) is the same as the data in packet with timestamp \(-376000\). A special translation should be done when proxy obtains RTP packets switches from a segment from disk cache to a segment from the network. Before forwarding the first packet obtained from the network, the HUSTProxy needs to determine the outgoing sequence number and timestamp for this packet. The translations are explained in Figure 3. If the RTP packets of segment 1 are obtained from playing of the beginning of the stream, the random base timestamp of this stream should be 180000. We can calculate that the cached time position in the disk is \((630000 - 180000) / 90000 = 5\) (seconds). And the value of \( \omega \) should be \((630000-180000)+6000\), which is 456000. Whereas the actual value we obtained from the network is \(-376000\). So the packets gap between the two sessions is \(G = 456000 - (-376000) = 832000\). By the session gap \(G\), we can obtain that the outgoing timestamp of the first packet from the network is \(-376000 + G = -376000 + 832000 = 456000\), and timestamp of the last packet from the network is: \(-16000 + G = -16000 + 832000 = 816000\). The SSRC field in the RTP headers fetched from network can be set the same as value of RTP headers which stored in cache disks.

**DESIGN AND IMPLEMENTATION**

In the implementation we use FreeBSD5.0 (McKusick, 1996) as our prototype operating system and the whole proxy system is programmed by C++ code. Besides some fundamental functions of a multimedia proxy have been implemented, our prototype has realized techniques such as prefix caching, stream prefetching, data transfer-rate control, which can contribute to improve cache efficiency and reduce service start-up latency. The whole system is composed by request manager module, cache manager module, prefetch manage module, request module and server module.
System Architecture

As the HUSTProxy is located in the network, it appears as a client for a server and as a server for a client, so there are a request module and a server module in the proxy. Figure 4 depicts the internal architecture of HUSTProxy as a combination of a client and a server. As shown in Figure 4, the video streaming was controlled through RTSP/TCP sessions. There were two sets of sessions for the client. The first was established between the originating video server and proxy to retrieve uncached blocks. The other was between the proxy and client. Each of video and audio was transferred over a dedicated RTP/UDP session. The quality of service was monitored over RTCP (RTP Control Protocol) (Schulzrinne, 1996)/UDP sessions. We will explain the functionality of individual components by describing Request Manage Module within the HUSTProxy in the next section.

Request Management

Request Manager module handles all the RTSP signaling between HUSTProxy and client or server. Figure 5 shows a completed process of streaming proxy deal with a client RTSP request. First, a client begins by establishing connections for audio/video streams with the proxy server using a series of RTSP OPTIONS, DESCRIBE, and SETUP messages. An OPTIONS message is used to request communication options. A DESCRIBE message is used for media initialization and a SETUP message is used for transport parameter initialization. These RTSP messages are received by the Request Manager through an RTSP Server module (Figure 4). The proxy server relays RTSP OPTIONS, DESCRIBE, and SETUP messages to the video server. Thus, connections between the video server and the proxy server are also established at this stage. Then, the client requests delivery of the video stream by sending an RTSP PLAY message. A proxy maintains information about cached blocks.
in the Cache Table. Each entry in the table contains a block identifier, the size of the cached block, and the flag. The size is set at zero when the block is not cached. The flag is used to indicate that the block is being transmitted. On receiving a PLAY request for a video stream from a client, the Request Manager begins providing the client with data blocks. It first examines the cache table. As long as a copy of requested stream is available in the cache, the Request Manager sequentially reads them out and sends them to the client through the RTP Sender. Simultaneously, The Request Manager forwards the PLAY request with a modified Range header based on how much data have been stored within the local cache. In this way, the remainder of the requested stream is fetched in advance for later streaming response, as shown in Figure 5(b).

If the requested stream $s$ is missed [Figure 5(a)], the Request Manager simply relays the RTSP PLAY message to the video server with no modification. On receiving a stream from the video server through the RTP Receiver, the Request Manager sets its flag with on to indicate that the streaming blocks is being transmitted, and it relays the blocks to the RTP Sender. When reception is completed, the flag is cancelled and the Request Manager deposits the blocks in its local cache disk. If there is not enough room to store the newly retrieved block, the Cache Manager replaces one or more less important segments in the cache with the new stream segment. When a proxy server receives an RTSP TEARDOWN message from a client, the proxy server relays the message to the video server, and closes the sessions.

Figure 5. RTSP message exchange across client, proxy and server

(a) RTSP signaling and data deliver on cache miss

(b) RTSP signaling and data deliver on cache hit
Cache Replacement

HUSTProxy periodically invokes replacement routines and if the amount of total cached data is higher than a configured high-water mark, a sufficient number of existing segments have to be evicted from cache. For achieving popularity-proportioned replacement (discussed in the previous section), we use the term cache efficiency as the measure of cache replacement. Cache efficiency of a stream is defined as the ratio of its popularity to the cache space for the stream allocated. It represents the correlation between the caching benefit and the cache cost. To perform this replacement, proxy server calculates caching efficiency value of all cached streams and selects the victim stream with the minimum value. In addition, the victim stream should be eliminated from last cached segment to make room for a new popular one.

\[
\text{CacheEfficiency} = \frac{\text{StreamPopularity}}{\text{StreamCachedSpace}}
\]

(2)

- **StreamCachedSpace**: Size of storage space allocated for the stream, that is, the size of cached portion of the stream. In this paper, we use \( P_i \) denote the allocated disk space of cached portion of stream \( i \).
- **StreamPopularity**: Popularity of a stream as the total amount of data of each stream played back by clients during an interval.

Since a continuous media object has a large size and wide range of playback duration from several seconds to 1 or 2 hours, users are likely to just view the opening of a stream and then close it, which is because it is not a popular media for clients. Thus, it is not proper to use the access frequency as the popularity of a stream. Only the total amount of data

![Figure 6. Streams playing back status during the time \( [r_1, r_2] \)](image-url)
played back by all clients is the real popularity judgment. Figure 6 shows the way to measure the total amount of data for a particular stream that is played back by all clients during an temporal distance of $\Delta r_1, r_2$. Stream $s_1, s_2, ..., s_n$ denote active streams that have been playback within time $[r_1, r_2]$. The solid lines indicate the extension of the streams that are played back within $[r_1, r_2]$ period. For a particular stream, the total amount of data played back by all clients within this temporal distance is used to calculate the popularity of the stream. Caching efficiency of a steam depends on the accomplished data access amounts. For example, from Figure 6 we can see that the stream $S_1$ are playing back by client 1, client 6 and client 7. We calculate the total amount of data of $S_1$ played back by these three clients during the time $[r_1, r_2]$, and we consider the result as the popularity of $S_1$ during the time interval of $\Delta r_1, r_2$.

For presenting stream popularity and cache efficiency expressions, we define related variables in Table 1.

The total amount of data played back by all clients for stream $i$ can be deduced as EQ(3):

$$D_i = \sum_{j=1}^{m} D_{ij}$$  

And the caching efficiency of a particular stream $CE_i$ can be given by EQ(4):

$$CE_i = \frac{D_i}{P_i} = \frac{\sum_{j=1}^{m} D_{ij}}{P_i}$$  

From EQ(3), we deduce that the caching efficiency can be increased either by enhancement of stream popularity, that is, the total amount of data played back by all clients, or by a decrease of cached space of this stream. Fortunately, when the most unpopular stream $s$ is sacrificed by eliminating the last segment of the stream, this just promotes the stream cache.

Table 1. Parameters definition

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_{ij}$</td>
<td>Amount of data played back of stream $i$ by client $j$ during $[r_1, r_2]$</td>
</tr>
<tr>
<td>$D_i$</td>
<td>Total amount data of stream $i$ played back by all clients during $[r_1, r_2]$</td>
</tr>
<tr>
<td>$m$</td>
<td>Total amount of clients have accessed proxy during $[r_1, r_2]$</td>
</tr>
<tr>
<td>$P_i$</td>
<td>Allocated disk space of cached portion for stream $i$</td>
</tr>
<tr>
<td>$CE_i$</td>
<td>Caching Efficiency of stream $i$</td>
</tr>
</tbody>
</table>

Figure 7. Connection of server-proxy-client architecture
efficiency by cutting its allocated disk space. Going on evicting the subsequent last segment of the stream s until its cache efficiency is promoted enough that its value is larger than the second most unpopular one, then remove its CacheItem from the tail of linked list and insert it to an appropriate position. So the tail of the list is always the most unpopular stream, as shown in Figure 2. These operations continue until the total size of all cached streams is less than the maximum allowable cache space.

PERFORMANCE EXPERIMENTS

Network Load Reduction

The goal of a proxy is to intercept client requests transparently and reduce the number of requests sent to the server, which leads to a reduction of server load and network load. To demonstrate the decrease of network load by our HUSTProxy system, we use traffic variation ratio (TVR) as the criterion of evaluation.

\[
TVR = \frac{\xi_{sp}(t)}{\xi_{pc}(t)}
\]  

- \(\xi_{sp}(t)\): The total amount of data transferred from the server to proxy at time \(t\).
- \(\xi_{pc}(t)\): The total amount of data transferred from the proxy to client at time \(t\).

A larger TVR value indicates increase in relative network load. On the contrary, a smaller TVR value means a decrease in network load. The maximum value of TVR is 1. In our experiments, we use a server-proxy-client setup, where the client simulates multiple media players by our RTSP request generation program. This program disregards the requested data, only periodically generating requests for media objects following a Zipf-like (Breslau, 1999) distribution. Simultaneously, we detect the network traffic both of the server-proxy path and proxy-client path with the time \(t\) increases. The connection of the experiments is shown in Figure 7.
Figure 8 shows that as time increases, the traffic variation ratio variation with different prefix caching size. The client generates RTSP requests at the interval of 2 seconds and the URLs are selected following a Zipf-like distribution from a set of 1000 MPEG media objects which located in the server. The requested objects have the same playback length of 60 seconds. In each time, we specify a fixed prefix caching size, which is 20%, 40%, 60% and 80% of the total cached streams long respectively. The total cache space is always large enough that the prefix cache segments would not be replaced, so we need not consider hit ratio variation caused by the disk space limitation. From Figure 8, we see that the TVR changes more rapidly when the prefix caching size increases. Larger variations of TVR indicate a greater drop in network load. We notice that the traffic variation ratio is up to 25% when the prefix size is 80% of the total stream length, which means that the network load of the path between server to proxy is only 25% of the path from proxy to client.

### Start-Up Latency Reduction

Another major function of a proxy system is to reduce the client request start-up latency. When a client receives stream data from a server or the proxy, it does not start playing the object until its playout buffer is filled to absorb network jitter. So there is a start-up latency at the beginning of playback every time. Now we consider two scenarios: one involves a server-client architecture and the other is server-proxy-client architecture. For presenting the client start-up latency expression, we define the related parameters as shown in Table 2.

We assume the delay between the server and the proxy is $t_1$, and the proxy to client path delay is $t_2$. Then the delay between the server and the client is $t_1 + t_2$, and the round-trip delay is $2t_1 + 2t_2$.

Considering server-client architecture, the media server sends out data packets according to its playback rate $\alpha$ bytes/second, and assume each client keeps a playout buffer of $S$ seconds, the initial $S$ seconds of data are related to client’s start-up latency. The start-up latency of this scenario $T_0$ can be given by

$$T_0 = 2t_1 + 2t_2 + S$$  

(6)

$$T_1 = t_2 + \max(S, \alpha / \beta, 2t_1) + t_2 + (S - S_1)\alpha / \min(\alpha, \beta)$$  

(7)

### Table 2. Parameters and definition

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>The delay between the server and the proxy</td>
</tr>
<tr>
<td>$t_2$</td>
<td>The delay between the proxy and the client</td>
</tr>
<tr>
<td>$S$</td>
<td>Start playback buffer length (the scale is seconds)</td>
</tr>
<tr>
<td>$S_1$</td>
<td>Proxy has cached the first $S_1$ seconds of the requested stream</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Data transfer rate from server to client</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Data transfer rate from proxy to client</td>
</tr>
</tbody>
</table>
Next, we consider the server-proxy-client architecture. If \( 0 \leq S_1 \leq S \), the proxy starts two processes concurrently. One is to download the existing \( S_1 \) seconds of data to the client at \( \beta \) transfer rate, which takes \( (S_1 \alpha)/\beta \) seconds to transfer \( S_1 \) seconds of data. The other process is to request the rest of \( S - S_1 \) seconds of data from stream server. It takes \( 2t_1 \) seconds for the first byte of the data to arrive proxy. So the time for both processes to finish is \( \max(S_1 \alpha)/\beta, 2t_1 \). The \( S - S_1 \) seconds of data is transferred firstly from server to proxy, then from proxy to client. Thus the transfer rate is \( \min(\alpha, \beta) = \alpha \). Therefore the transfer time of this part of data is \( t_2 + (S - S_1) \alpha/\alpha = t_2 + S - S_1 \). We can present the start-up latency of this scenario by Eq(7).

If \( S_1 = 0 \), all the initial \( S \) seconds of data are transferred from server, and with the transferring rate of \( \alpha \). In this case, \( \max(S, \alpha/\beta, 2t_1) = 2t_1 \), hence the start-up latency \( T_1 = 2t_1 + S + 2t_2 \) which is the same as \( T_0 \).

If \( S_1 \geq S \geq 0 \), all the initial \( S \) seconds of data are transferred from proxy, and with the transferring rate of \( \beta \). In this case the client start-up latency is:

\[
T_2 = 2t_2 + (S\alpha)/\beta
\] (8)

Figure 9 shows the client start-up latency with the variation of cached size (seconds). The solid lines represent our experimental results, and the dotted lines represent calculation results from the Eq(7). The parameter setting in the formula is based on \( t_1 = 15 \) ms, \( t_2 = 1 \) ms. In these experiments, the value of \( \beta/\alpha \) is changed from 1 to 10 by varying the proxy’s data transfer rate \( \beta \), and the start playback buffer length \( S \) is 5 seconds. We set different value of \( S_1 \) as 1, 3, 5 seconds respectively for each experiment. When \( S_1 = 0 \), the latency is the same as getting data directly from the server. When \( S_1 > 0 \), the \( S_1 \) seconds of data are transferred with rate \( \beta \), and the \( S-S_1 \) seconds of data are transferred with rate \( \alpha \). From Figure 9, we note that the calculation results and the experimental results appear closely, and we can also see that with a fixed \( S_1 \), the start-up latency decreases as the increase of the value of \( \beta/\alpha \), with a fixed \( \beta/\alpha \) value, the start-up latency decreases as the increase of \( S_1 \).
SUMMARY AND FUTURE WORK

This work describes the advantage and feasibility of implementing a proxy caching system for multimedia streaming service over the Internet. Some issues and challenges in the design of proxy system are detailed and discussed and our solutions are presented. The prototype implementation—HUSTProxy—is used in validating the performance claims. From experimental results, by caching videos relatively close to the clients, HUSTProxy can reduce overall client start-up latency and the possibility of adverse Internet conditions disrupting video playback. From the point of view of the server, HUSTProxy dramatically reduces network load by intercepting a large number of server accesses, which ensures a large number of video requests and also accommodates the scalability.

We plan to continue this work in a couple of directions. Firstly, implementation of a proxy cluster system is our next target. In this way, all the proxies have joined the cluster are cooperative fashion. On receiving a request, the local proxy serves the request at first. If checks cache missed, the proxy forwards the request to the most appropriate proxy to check whether the requested data exist in other proxies. Issues are related to how these proxies cooperate, how to find the most appropriate proxy. And this scheme also needs a global cache management and cache replacement policy to guarantee cache consistency. Secondly, we will incorporate quality adaptive function in HUSTProxy in the near future. It is achieved by delivering different number of layer for layer-encoded streaming according to the proxy-to-client bandwidth. Implementation of this function is relatively easier than the first one.

REFERENCES


Guo Hui received a BS in computer science and technology from Jianghan Petroleum University, Hubei, China (July 1999), and an MS in computer science from Huazhong University of Science and Technology (HUST) (August 2001). He is currently a PhD candidate in computer science in HUST and a research assistant at the National Storage System Lab, HUST. His research interests include network storage, streaming media, multimedia network and distributed systems.

Zhou Jingli, a visiting scholar in the US from 1996-1997, graduated from the Electrical Engineering Department of HUST (1969). She is currently a professor in the Department of Computer Science, HUST, and vice dean of the School of Computer Science and Technology, and a member of IEEE and IEEE Computer Society. Her research interests focus on technology and application of multi-media, transmitting systems, computer networks and storage area networks. She has more than 60 papers and two monographs published.

Wang Yunpeng received a BS in electronic engineering from Nanjing University, Jiangsu, China (1989), and an MS in information science from the Electronics and Information Department, Huazhong University of Science and Technology (HUST). He is currently a PhD candidate in HUST, and he is also a senior engineer in the Information Department, China Petroleum & Chemical Corporation (sinopec). His research interests include network congestion control, virtual private network, and wireless network technology.

Yu Shengsheng graduated from the Department of Electrical Engineering, HUST (1966), and was a visiting scholar in former West Germany in 1985. Now he is leading scientist of the National Storage System Lab, and a member of IEEE and IEEE Computer Society. In 1990, he was awarded the honorary title of Excellent Returning Scholars with Significant Achievements. His main research interests are focused on computer system architecture, network communication technology, computer general storage technology, multimedia information processing, etc. He is the author of more than 70 papers, a textbook and one published monograph and holds two patents.
WebGuard: Web Adult Content Detection and Filtering System

Mohamed Hammami, Ecole Centrale de Lyon, France
Liming Chen, Ecole Centrale de Lyon, France

ABSTRACT

This paper describes a Web filtering system “WebGuard,” which aims to automatically detect and filter adult content on the Web. WebGuard uses data mining techniques to classify URLs into two classes: suspect URLs and normal URLs. The suspect URLs are stored in a database, which is constantly and automatically updated in order to reflect the highly dynamic evolution of the Web. When working, WebGuard simply captures a user’s URL, matches it with the suspect URLs stored in the database and takes an appropriate action — filtering or blocking — according to the result of the analysis. We started out with a study of most existing software so as to get to know the possibilities and functionalities available on the market at the moment. This phase enabled us to better evaluate the performances of our product as it was being developed. Thus, the second phase of our work was devoted to research into the usual algorithms regarding their advantages and drawbacks. Having gathered this knowledge, we are currently implementing a system that will combine several algorithms in order to increase the software’s performance. Our preliminary results show that it can detect and filter adult content effectively.

Keywords: adult content detection; data mining; text mining; Web filtering; Web search techniques

INTRODUCTION

Nowadays, the Internet is taking an increasingly pivotal place in everyday life. Not only is the Internet community constantly growing, it is also getting younger. In fact, children each day have easier access to the Internet, which may cause socio-cultural problems. According to a study carried out in May 2000, 60% of the interviewed parents were anxious about their children navigating on the Internet, particularly because of the presence of adult material (Gralla & Kinkoph, 2001). Furthermore, according to the Forrester lookup, a company that examines operations on the Internet, online sales related to pornography adds up to 10% of the total amount of online operations (Gralla & Kinkoph, 2001). This problem concerns parents as well as companies. For example, the company Rank Xerox laid off forty...
employees in October 1999 who were looking at pornographic sites during their working hours. To avoid this kind of abuse, the company installed program packages to supervise what its employees visit on the Net.

Answering this demand, some companies have proposed solutions for Web site filtering. Their products concentrate on IP-based filtering, and their classification of Web sites is mostly manual; that is to say no truly automatic classification process exists. But, as we know, the Web is a highly dynamic information source. Not only do many Web sites appear everyday while others disappear, but site content (especially links) are updated frequently. Thus, manual classification and filtering systems are largely impractical and inefficient. The ever-changing nature of the Web calls for new techniques designed to classify and filter Web sites and URLs automatically (Hammami, Tsishkou, & Chen, 2003; Hammami, Chahir, & Chen, 2003).

In this article, we propose an adult content detection and filtering system called WebGuard, which uses text analyses. Compared to other systems, WebGuard has the advantage of combining several data mining algorithms for Web site classification.

The remainder of this article is organized as follows. We started out with a study of the best-known software on the market. In the next section, the WebGuard architecture is presented. Following that, the extraction of feature vectors from Web pages is reviewed. Subsequently, the classification of URLs through data mining techniques is discussed. Finally, an experimental evaluation and comparison results are presented.

STATE-OF-THE-ART AND ANALYSIS OF THE COMPETITION

We started out with a study of best-known software on the market so as to get

Figure 1. Global test evaluation
to know the possibilities and functionalities available on the market at the moment. This phase enabled us to form an accurate judgement about the performance of our product as it was being developed. We tested the most commonly used filtering software over a database of 400 Web sites. Half of the Web sites were pornographic and the rest were non-offensive. The six products we tested are: Microsoft Internet Explorer (RSACi) [Content Rating Association (ICRA)], Cybersitter 2002 (www.cybersitter.com), Netnanny 4.04 (www.netnanny.com), Norton Internet Security 2003 (www.symantec.com), Puresight Home 1.6 (www.icognito.com), and Cyber Patrol 5.0 (www.cyberpatrol). Figure 1 shows the results of our study. It compares the success rates of the most common software on the market today.

These tests enabled us to bring to light several issues that our software had to face.

The first function, which seems important to users of this product, is the configurability of the level of selectivity of the filter. There are different types of offensive content and our study also shows that, while highly pornographic sites are well handled by the software, erotic sites or sexual education for instance are unaccounted for. That is to say they are either classified as highly offensive or as normal sites.

Another major problem is the fact that all analysers on the market today rely solely on text analysis. This means that they compare the text found in a Web site to a dictionary of prohibited words to determine whether it is pornographic or not. Thus, the efficiency of the analysis greatly depends on the word database, its language, and its diversity. A product using an American dictionary will not detect a French pornographic site. The success rate of this kind of system can reach 90% for the best products.

To sum up, not only must our software be based on a text analyser which is attached to the browser, but it must also use complementary means such as the analysis of different HTML tags of the source code to retrieve more information from the Web site in order to be able to make a decision that is significantly more reliable than any other software available today. The tags are analysed by type (links, images, keywords) and supply a certain

Figure 2. WebGuard architecture
number of pertinent criteria. The advantage of our approach compared to the competition is the algorithm weighting system we developed which uses several data-mining algorithms and thus increases the reliability of the decision criteria.

WEBGUARD ARCHITECTURE

The Web filter system (WebGuard) aims to block those sites with pornographic or other nudity and sexually explicit language. It provides Internet content filtering solutions and Internet blocking of pornography, adult material, and many more categories. The Internet will thus become more controllable and therefore safer for both adults and children.

The formulation of the WebGuard (cf., Figure 2) is as follows:

- Fully automated adult content detection and filtering
- Categorization into “black list” (access denied) and “white list” (access allowed) to speed up navigation
- If the site is not recorded on the “black list” or “white list” the engine will then analyze the textual information and make a further decision on the site’s access allowed/denied status. The black list/white list file is then updated.

The program analyzer was developed by separating it into several modules:

- A robot is needed with the ability to seek the code of a page; this we have called HTTP user.
- Then there is a “parser,” a program item able to detect the HTML tags of the code and to extract the text itself.
- One can then calculate a certain number of criteria (percentage of words identified as pornographic, etc.) by analyzing the text and the tags.
- Finally, rules make it possible to say, by knowing the value of the criteria, if the page is or not authorized.

The analysis of the text consists of traversing the file word by word, and checking if these words belong to a dictionary of “prohibited” words which we have put together. This dictionary is in fact a text file containing words that are explicitly shocking, and it can be supplemented or changed to improve the effectiveness and to extend the application field to other types of Web pages.

The tags are analyzed according to their type (links, image, words) and provide a certain number of relevant criteria (links towards known sites, images with explicit names, explicit keywords...).

The application relies on the principle of analyzing the HTML code of a Web page. We thus should be equipped with a set of functions that make it possible to read from a server then to analyze a page. The analyzer is composed of three main functions: an http client used to connect to the Web server and retrieve the source code, an html flag analyzing function, and a content analyzer to make an initial treatment of the raw data.

HTTP/1.1 Client

The client is in fact a HttpSession function, which takes as a parameter the Uniform Resource Identifier (URI) of the Web page that is being worked on, and eventually the one of the page that link was found on (which is known as the “referrer”). This function allocates a memory block to store
the code of the page and sends back a character pointer to this block.

The client employs handshaking HTTP/1.1 protocol to communicate with the servers — in fact, almost all the actual Web servers use HTTP/1.0 or 1.1 and the standard requires that a server 1.0 is able to answer a “simple” request 1.1.

HTTP protocol is based on a request/answer mechanism: after the connection, the browser sends a HTTP request, having in our case the following form (only part of the fields envisaged are used):

GET <URI> HTTP/1.1
Host: <Hôte>
Referer: <Referer>
Accept: text/*
Accept-Language: fr
Connection: close
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)
TE:

Let us clarify the significance of this request:
1st line: One uses the GET command (simple reading of a page) to obtain object < URI > via handshaking HTTP/1.1.
2nd line (Host): we specify the host of the page. In fact it is possible that the same physical server hosts several different sites (typical case: personal sites: dupond.pagesperso.fr, durand.pagesperso.fr ...) this is why we need this precision envisaged by the protocol.
3rd line (Referer): see above.
4th line (Accept): Data types accepted in answer
5th line (Accept-language): If the desired document exists in several languages, it is the desired language here.
6th line (Connection): asks the server to signal the end of the communication by the closing of the connection (to simplify the recovery of the data).
7th line (User-Agent): indicates the type of browser—necessary because certain sites adapt their contents according to the browser.
8th line (TE): TE means “Transfer Encoding” - specifies the document’s format.

The response of the server takes the following form:

< ErrNo > < ErrMsg >
< Heading >
< - blank line of separation
< Contained >

< ErrNo > and < ErrMsg > are a pair number / error message (or simply information) summarizing the result of the request. Most current are:
- “200 OK”: the required contents follow the heading.
- “302 Found”: the required document exists but with a different address than the one specified in the heading.
- “404 Not Found”: the required document does not exist.

**Figure 3. Algorithm for reading a Web page**

```
Algorithm:
1. break up the URI into a host / document
2. solve the address of the host.
3. connect to the address (port TCP 80).
4. build HTTP request
5. send request.
6. as long as the connection was not closed,
   - wait the reception of data.
   - receive the data.
7. close the connection
8. separate the heading from the contents by decoding the contents.
9. read the error number in the heading.
   - if OK
      return the contents
   - if redirection
      read the document with the new address
   - if error
      return an empty document sign error.
```
The algorithm for reading a page by the handshaking HTTP protocol is shown in Figure 3.

The program analyzer uses an auxiliary function getCode, which after calling HttpGetUrl stores the contents in a file.

**Analyzer of HTML Tags**

The HTML Language is based on a system of tags (character strings delimited by the symbols < and >) making it possible to insert various objects in the text or to modify its properties. With the need to treat these tags in a particular way (particular processing of the key words, links, images...) we developed a function breaking up these tags into a data model easily usable in the analysis.

Structure of a HTML tag:  \(<\) Name Arg1 Arg2 Arg3... ArgN \>  
Structure of the Argx arguments: NameArg or NameArg=ValArg.  
Lastly, structure of the ValArg value: TOKEN or “chaîne” or ‘chaîne.’

A TOKEN is a chain without space (a space being a simple space, a tab character, a line break, a return or a vertical tab).

The function parse_balise takes in input a HTML tag stripped of its opening and closing carets, and returns a character string array whose elements are:

\[ [0]: \text{Name of the tags} \]
\[ [2k – 1]:, k=1..N: \text{name of the argument K} \]
\[ [2k]: \text{value of the argument K (possibly empty)}. \]

One can thus, by means of a function get_value obtain the value of the argument in tags (a distinction is made between empty and non-existent arguments).

**Content Analyzer**

This analyzer works on a class created to represent a site. It asks the flag analyzer to recognize the different zones of data and then treats each type of data.

---

**Figure 4. Structure of site class**

```
Class: Site  
Attributes
char * url; char * referer; int clean; int n_mots; int n_x_mots; int n_images; int n_x_images; int n_liens; int n_x_liens; int n_x_url; int n_xxx_liens; int n_x_meta;  
Methods
Site(char *,int); int get_mots(); int get_n_mots(); int get_pics(); int get_pics(); int get_n_liens(); int get_n_x_liens(); int get_n_x_url(); int get_n_xxx_liens(); int get_meta(); int analyse(); void export(char *); int is_clean(); void traite_balise(char *); ~Site();
```

**Figure 5. Environment of site class**

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.
For instance, it will recognize a text zone and count the number of words that appear in the dictionary of forbidden words. Figure 4 shows the different attributes and methods of the Site class.

Our application is built around the Site class (Figure 5). This class contains on one hand the analytical method which calls up the getCode procedure defined in the client and which goes through the document character by character, detecting the beginnings and endings of the tags and regrouping the words, and on the other hand the export method which is useful at the end of the development but also to store and put into a cache the document parameters to reuse them later. This method adds to each analysis a recording in a text file, which we called data-base.txt.

A site object must be constructed with two parameters: the URL and the class. The latter has a value of “0” for a site which we know has adult content and “1” when the site has no adult content. The Site class needs two source files to function: the list of the URLs and the dictionary.

The tags are subjected to the method traiter_balise() which looks up and treats the following tags:

- **Meta:** if the attribute name is “keywords,” we look up the dictionary words in the content attribute their number is stored in n_x_meta
- **a:** the value of the attribute href is treated – look up if the target URI appears in the already treated and marked document “undesirable.” If it is the case, n_x_linkages is incremented.
- **Frame:** An object of Frame class derived from Site is created with the URI found in the attribute src, the analysis is carried out for this new URI and the results are added to the current page.
- **Img:** n_images is incremented - the attribute src is traversed with the looking up of words of the dictionary, n_x_images is incremented if a word is found. In the future an analysis of images should be implemented on this level.

The words, once delimited, are counted (n_mot and subjected to looking in the dictionary (n_x_mot incremented if found).

**WEB PAGE FEATURE VECTOR EXTRACTION**

Before detecting and filtering the URLs with adult content, we need to know which URLs are sex-oriented and which are not. This is quintessentially a problem of URL classification.

In order to sort the URLs into two classifications, sex-oriented and non sex-oriented, we first decide which features of a URL can be used as its defining features. Considering many sex-oriented Web pages, we use textual signature as the features of a URL. At the same time, many sex-oriented URLs have some pop-up windows and if a Web page links to another Web page it is possible this Web page also has sexual content. Consequently, the number of pop-up windows on a Web page and the nature of a Web page’s links (sex relevant or not) are also important features of an

$$VoW = \{n\_mots, n\_x\_mots, n\_images, n\_x\_images, n\_liens, n\_x\_liens, n\_xxx\_liens, n\_x\_url, n\_meta, n\_x\_meta, pcxmots, pcxmeta, pcxliens, pcximage\}$$
URL. The URLs of many sex-oriented Web sites contain sexually explicit words, which is another clear indication that the site contains sexual content. To summarize the above, we give the feature vector of a Web site, as shown in the box below:

\[
\begin{align*}
n_{\text{mots}} & \text{ is the total number of words on the current Web page,} \\
n_{\text{x_mots}} & \text{ the number of sexually explicit words,} \\
n_{\text{images}} & \text{ the total number of images,} \\
n_{\text{x_images}} & \text{ the number of images whose name contains a sexually explicit word,} \\
n_{\text{liens}} & \text{ the total number of links,} \\
n_{\text{x_liens}} & \text{ the number of links which contain sexually explicit words,} \\
n_{\text{xxx_liens}} & \text{ the number of links that have been classified as sex-oriented,} \\
n_{\text{url}} & \text{ the number of sexually explicit words in the URL,} \\
n_{\text{meta}} & \text{ the total number of keywords,} \\
n_{\text{x_meta}} & \text{ the number of sexually explicit keywords,} \\
n_{\text{liens}} & \text{ the percentage of links containing sexually explicit words and finally} \\
n_{\text{image}} & \text{ the percentage of images whose name contains a sexually explicit word.}
\end{align*}
\]

USING DATA MINING TECHNIQUES TO CLASSIFY URLS

Fundamentals

The creation of a database for the data mining process requires a large number of each category: in our case 1,000 pornographic and 1,000 non-pornographic sites were added to the database. This number of 2,000 is necessary not only to simulate a good representation of Internet content but also because we are collecting a great deal of different information.

We have collected these sites manually from the Internet because we wanted our base to be as representative as possible. Within the adult sites we find content ranging from the erotic to the pornographic, and within the non-adult sites we find health-based information, anti-pornographic, anti-AIDS sites, and etcetera.

Once the feature vectors of all the URLs have been constructed, the task is to construct a classifier to classify these URLs into two classes: adult sexual URLs and other URLs.

A number of classification techniques from the statistics and machine learning communities have been proposed (Quinlan, 1986; Quinlan, 1993; Weiss & Kulikowski, 1991; Zighed & Rakotomala, 1996). A well-accepted method of classification is the induction of decision trees (Breiman, Friedman, Olshen, & Stone, 1984; Quinlan, 1986; Zighed & Rakotomala, 1996). A decision tree is a flowchart-like structure consisting of internal nodes, leaf nodes, and branches. Each internal node represents a decision, or test, on a data attribute, and each outgoing branch corresponds to a possible outcome of the test. Each leaf node represents a class. In order to classify an unlabeled data sample, the classifier tests the attribute values of the sample against the decision tree. A path is traced from the root to a leaf node, which holds the class prediction for that sample. Let the set of Web sites be:

\[
C : \Omega \rightarrow \varphi = \\
\{ \text{suspect URLs, normal URLs} \}
\]

\[
W \rightarrow C(w)
\]

The observation of \(C(w)\) is not easy; therefore we are looking for mean value to describe class \(C\). The process of graph construction is as follows: we begin with a
sample of sites, both suspect URLs and normal URLs and look for the particular attribute which will produce the best partition. We repeat the process for each node of the new partitions. The best partitioning is obtained by maximizing the variation of uncertainty $\Delta I_{\lambda}$ between the current partition and previous partition. As $I_{\lambda}(S)$ is a measure of entropy for partition $S$, and $I_{\lambda}(S_{i+1})$ is the measure of entropy of the following partition $S_{i+1}$.

The variation of uncertainty is:

$$\Delta I_{\lambda}(S) = I_{\lambda}(S) - I_{\lambda}(S_{i+1})$$

For $I_{\lambda}(S)$ we use the quadratic entropy (a) or Shannon entropy (b) according to the method:

$$(a) \quad I_{\lambda}(S) = \sum_{j=1}^{K} \frac{n_j}{n} \left( - \sum_{i=1}^{m} \frac{n_{ij} + \lambda}{n_{ij} + m\lambda} \left( \frac{n_{ij} + \lambda}{n_{ij} + m\lambda} \right) \right)$$

$$(b) \quad I_{\lambda}(S) = \sum_{j=1}^{K} \frac{n_j}{n} \left( - \sum_{i=1}^{m} \frac{n_{ij} + \lambda}{n_{ij} + m\lambda} \log_2 \left( \frac{n_{ij} + \lambda}{n_{ij} + m\lambda} \right) \right)$$

Where $n_{ij}$ is the number of elements of class $i$ at the node $S_j$ with $I \in$ (Suspect URLs, Normal URLs); $n_j$ is the total number of elements of the class $i$, $n_i = \sum_{j=1}^{K} n_{ij}$; $n_j$ the number of elements of the node $S_j$, $n_j = \sum_{i=1}^{m} n_{ij}$; $n$ is the total number of elements, $n = \sum_{j=1}^{K} n_j$; $m = 2$ is the number of classes (suspect URLs, normal URLs).

As $\lambda$ is a variable controlling effectiveness of graph construction, it penalizes the nodes that are insufficiently effective. This criterion will control and support the fusion between summits. This is a major specificity of Sipina (Zighed & Rakotomala, 1996). The algorithm stops if no changes in uncertainty occur.

**Dictionary and Weighting System**

The effectiveness and the quality of the results obtained by the classification methods for this type of application depend on the nature, language, and diversity of the word database (or dictionary). By contrast to other methods we have used a multilingual dictionary (French, English, German, Spanish and Italian).

In our system, “WebGuard,” several classification methods are used [ID3, C4.5, SIPINA (with two different values for the admissibility constraint) and Improved C4.5] that can be combined in order to ensure a
higher degree of accuracy. Figure 6 shows the individual success rates of these algorithms.

These results are good but not good enough to obtain a product whose efficiency is competitive. In order to determine whether a Web site is pornographic or not, we use a weighting method: each algorithm has a coefficient and the sum of these five coefficients is equal to one. For each algorithm, the program examines the Web site and returns a Boolean value equal to one when the site is classified as pornographic and zero when it is classified as clean. Using the Boolean results from each algorithm and the weighting coefficient, we obtain a decimal number between zero and one. The higher this final result, the more likely the Web site is pornographic.

The calculation process of the coefficients was designed to give the most influence to algorithms with the best failure rate while giving the other ones a share important enough so they can eventually contradict the majority ones if they agree. For each algorithm, the formula considers the a priori success rate and manipulates it so as to compare it to the other algorithms' rates. The difficulty consists in amplifying the differences between the initially obtained coefficients. This was done by using an exponential function and then weighting the obtained figures so that their sum would still be equal to one.

In addition, the user can configure the blocking degree to a level that suits his/her cultural background. Furthermore, the user can protect his/her configuration through a password.

**EVALUATION AND COMPARISON RESULTS**

Three methods were used for evaluating the performance of each algorithm: the random error rates method, cross-validation, and boot-strap.

*Figure 7. Results for ID3*

![Graph showing average global results for ID3 with a priori and a posteriori error rates for pornographic and non-pornographic sites.]

*Figure 8. Results for C4.5*

![Graph showing average global results for C4.5 with a priori and a posteriori error rates for pornographic and non-pornographic sites.]

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.
With regard to the random error rate method, the site base is divided into two subsets: one for learning and one for testing. The method consists of calculating for each classifier the global, a priori, and a posteriori error rates for the results obtained over the testing database. The global rate quantifies the behavior of the algorithm over the whole base but does not hold any information about the success for a specific class. The a priori rate gives this information while the a posteriori rate is linked to the credibility of the classifier. Since those rates depend on the choice of the subsets, the process is repeated several times with randomly chosen subsets.

Cross-validation is based on the same method of randomly chosen subsets for learning and testing. Our site base is divided into ten subsets. Nine subsets are used for learning while the remaining one is used for testing. For that division, the same rates are calculated. Then, another subset is chosen to be the testing database and the same
steps are repeated. Finally, ten tests are made and the final rates obtained are the average of the ten results.

Boot-strap is also based on a random choice of the elements of the learning base. It involves extraction, examination and returning of these elements. This results in a 0.368 probability for a site to be in the testing subset. The experience is repeated 100 times and the average rates are obtained.

**Random Error Rates Method**

- **Average global results for ID3**

  Figure 7 shows the *a priori* and *posteriori* error rates obtained by using ID3 method on pornographic and non-pornographic sites.

- **Average global results for C4.5**

  The different results for this method are shown in Figure 8.

- **Average global results for Improved C4.5**

  The different results can be seen in Figure 9.

- **Average global results for Sipina $\lambda = 5.22$**

  (admissibility constraint of 20)

  Figure 10 shows the results obtained by using the Sipina method with a parameter of $\lambda = 5.22$

- **Average global results for Sipina $\lambda = 12$**

  (admissibility constraint of 50) (cf. figure 11)
Figure 11 shows the results obtained by using the Sipina method with a parameter of $\lambda = 12$

**Cross Validation and Boot-Strap**

The three evaluation methods differ only on the choice of the partition (learning and testing database). As it has been discussed earlier, the partition is randomly chosen for the random error rates method while, for cross validation and boot-strap, a more systematic process is used.

Thus, having calculated the different types of error rates with the random error method, it is sufficient to only get the global error rates over the cross validation and boot-strap methods in order to confirm the obtained results. If the figures are in accordance, then the random error rates method yields significant results that illustrate the performance of our software.

The global rates for cross validation are shown in Table 1 and the global rates for boot-strap are shown in Table 2.

These figures can be summarized and more easily interpreted with Figure 12. Globally, the figures obtained for each algorithm are in accordance, even though it is clear that the representativity of the chosen partition is not the same for all the algorithms.

For instance, for ID3, C4.5 and Sipina 5.22 the random error rates are higher than those obtained with the other methods. This means that the rates obtained are (approximations by excess) of the error rates over boot-strap and cross validation. Thus, the results obtained with the random error rates method are the maximum error rates that can be obtained. Therefore, we can conclude that these rates are low (less than 4%) and that the enrichment of the dictionary and the choice of the parameters significantly influenced their performance.

Regarding Improved C4.5 and Sipina 12, even though the rate obtained with the random error rates method is the not the highest one, it does not differ from the two other ones by more that 1%. Therefore, even though the overall results of these tests are not as good as for the others, we can conclude that the results for these algorithms are encouraging.

The global error rate (global behaviour of the classifier) is low but can still be improved by the implementation of image analysis (Hammami, Chahir, Chen, & Zighed, 2003; Hammami, Chen, Zighed, & Song, 2002). The a priori error rates show that Sipina 12.50 and improved C4.5 are not as efficient as ID3, C4.5 and Sipina 5.22 concerning a priori rates on pornographic sites. This rate is the most impor-
tant for evaluation because it quantifies the efficiency of the classification of pornographic Web sites. This prevails even though Sipina 12.50 and improved C4.5 obtain better results for the a priori non-pornographic error rate because we decided it would be more important to block a non-pornographic site than to let a pornographic one be viewed. Thus, we mainly relied on those a priori pornographic rates to make decisions. Furthermore, the high failure rates for the a priori non-pornographic rate show that these two algorithms (improved C4.5 and Sipina 12.50) do not provide a reliable decision with regard to the classification of non-pornographic sites.

To conclude, the tests allow us to say that the error rates of the algorithms, when using the improved dictionary and the best parameters, are less than 4% (individually). This rate is lowered by the weighting system implemented into our software.

After having determined the optimal decision criteria we went on to perform several tests to evaluate whether our theoretical results correspond to real results. These tests were performed on a totally

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>ID3</th>
<th>C4.5</th>
<th>SIPINA 5.22</th>
<th>ID3+C4.5+SIPINA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success rates</td>
<td>94.09</td>
<td>94.09</td>
<td>95.7</td>
<td>94.99</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>ID3</th>
<th>C4.5</th>
<th>SIPINA 5.22</th>
<th>ID3+C4.5+SIPINA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success rates</td>
<td>5.11</td>
<td>3.41</td>
<td>2.84</td>
<td>3.41</td>
</tr>
</tbody>
</table>

Figure 13. Comparison chart
independent database to that used in the original data mining process. This base is the same as we used to evaluate the other software.

**Results from adult sites**

*Table 3* shows the results obtained from adult sites.

We can see the results confirm the theories we have developed in our approach. SIPINA is certainly the best algorithm, which justifies again the coefficient we gave it in the above section.

**Results from non-adult sites**

*Table 4* shows the results obtained from non-adult sites.

The good filters are distinguished from the less good by their capacity to correctly identify the true nature of the pornographic or non-pornographic sites. Sites containing the word “sex” do not all have to be filtered. Adult sites must be blocked but scientific and education sites must stay accessible.

Here the results are again very good. We have not attained 0% filtering for the non-adult sites, but this was a choice from the beginning of our study. We prefer that some sites of equivocal content, like sexology sites, be filtered rather than some adult sites not be filtered.

We have compared the WebGuard with other Web-based adult content detection and filtering systems. The comparison chart is shown in *Figure 13*. The selected systems are Cyber Patrol, Norton Internet Security, Pure Sight, Cyber sitter, Net Nanny, and IE (Internet Explorer).

The comparison was conducted on 400 Web sites, including 200 adult Web sites and 200 non-adult Web sites. The least effective results come from IE with 18% and 19% success rates while our system is the best with a 95% success rate. Other systems give success rates between 60% (Norton Security) and 88% (Net Nanny).

**CONCLUSION**

In this paper, we have presented the new system WebGuard for detecting and filtering Web pages with adult content in real time. WebGuard uses textual analysis with an adjustable scale. The textual analysis uses several classification approaches that can be combined to give higher accuracy rates. Our experimental evaluation shows the importance of our approach in such systems.

**REFERENCES**


Mohamed Hammami is a PhD student in computer science in Ecole Centrale at the Lyon Research Center for Images and Intelligent Information Systems (LIRIS) associated to the French research institution CNRS as FRE 2672. His research interest is in combining techniques of data mining and image analysis in order to resolve different classification problems such as image classification and Web site filtering. He was a staff member in the RNTL-Muse project. He has served on technical conference committees and as a reviewer in many international conferences.

Liming Chen is presently a professor in the Department of Computer and Mathematical Sciences, Ecole Centrale, Lyon. He received a BSc (1985) in mathematics and computer sciences from the University of Nantes, and graduated with his MSc (1986) and his PhD specializing in database systems from Université Pierre & Marie Curie, Paris VI (1989). Before moving to Ecole Centrale de Lyon, he first joined University Technologie de Compiegne, France, where he set up and led a research team funded by the French government information highway program and industry, working on distributed visual information systems. His current research interests includes visual information indexing, storage, and their fast and reliable access in a fully distributed context.
The Use of Efficient Cost Allocation Mechanisms for Congestion Pricing in Data Networks with Priority Service Models

Fernando Beltrán, University of Auckland, New Zealand
Cesar García-Díaz, University of Groningen, The Netherlands

ABSTRACT

This paper demonstrates the application of two efficient cost allocation mechanisms on a simple FIFO network model in order to find congestion-based prices for network access services. It is shown that it is possible to obtain congestion-dependent prices in data networks (such as the Internet), establishing simultaneously different service quality levels among the users. The rationale behind these efficient distribution mechanisms is an axiomatic framework that determines a set of basic principles, highlighting the need for a fair allocation. Due to the characteristics of the price mechanisms and the possibilities of the parameter space in which these mechanisms might be applied, it is also shown that, in some cases, such mechanisms need additional weighting schemes in order to establish coherent differences among the service quality levels. In conclusion, even though the axiomatic framework guarantees efficient allocation, checking coherency in its use is left to the network manager.

Keywords: Aumann-Shapley pricing; data networks; efficient congestion-based pricing; non-pre-emptive priorities; Shapley value

INTRODUCTION

Acknowledging the cost of congestion imposed by multiple users trying to get access to network services seems to be a promising path to a more efficient use of network resources. Applications with higher demand for resources such as multimedia applications coexist with more conventional less demanding applications, making it necessary to devise an economic solution to manage congestion in addition to the technical solutions already in place. In that direction, our work aims at quantifying the costs due to network congestion, making them explicit to the users in the form of prices.

Several influential articles appeared in the early 1990s (Cocchi et al., 1993; Mackie-Mason & Varian, 1995a; Shenker, 1995) that touched on the issue of designing pricing schemes for the Internet. Later on
there were extensions by Mackie-Mason & Varian (Mackie-Mason & Varian, 1995b) and the proposal for pricing according to expected capacity by Clark (Clark, 1998). A further step was given when priority was considered (Gupta et al., 1998), and the use of dynamic programming tools in congestion dependent prices was attempted (Paschalidis & Tsitsiklis, 2000).

The present article develops a pricing scheme based on a queuing model, extending the concepts originally worked on by McLean & Sharkey (McLean & Sharkey, 1994, 1997; McLean et al., 2004) for models with a first-in first-out (FIFO) discipline.

Our proposed model rests on the assumption that, under a centralized decision-maker acting on behalf of efficiency, the social cost imposed by the demands of users, which lead to congestion, must be acknowledged and dealt with in a fair manner. Such cost can be measured from the traffic characteristics of the different traffic sources. Recognition of the social cost imposed by others may act as a kind of tool to pricing congestion.

In our approach we turn our attention to existing cost-axiomatic methods that can assure the achievement of some desired properties of a cost-based pricing scheme. As Courcoubetis & Weber (2003) put it, “One necessary condition that cost-based prices ought to reasonably satisfy is that of fairness” (p. 164). One view of fairness follows cost causality; in that sense a cost-based pricing scheme should fully distribute costs while recognizing that users that impose the same effect on cost should have the same price.

Two relevant methods appearing in the literature in cost-axiomatic pricing are the Aumann-Shapley (A-S) prices and the Shapley value. These mechanisms are of interest because they fulfill a set of desirable properties, defined as axioms later on, and allow for the inclusion of heterogeneous users with different requirements, characterized by different value perceptions of information and network use.

A-S prices and the Shapley value are used to fully distribute the cost of congestion obtained as we use a queuing model of access service. Such access may well correspond to an Internet Service Provider, ISP, which serves two classes of users. The two-class approach serves to represent the further assumption that one class of users has service priority over the other. The users demand different services with different service characteristics, namely, the users’ arrivals rates and their service distribution rates. Using the average waiting time for a user type in the queue and the value to a user type of a service requirement, we define a congestion-based cost function. Such function is actually reflecting the loss due to the presence of multiple users trying to gain access to the provider.

We believe that embodying such desirable properties in axiomatic framework may help the designer of a pricing scheme understand how fairness in cost scheme due to congestion reflects on prices. As ISPs are growingly stressed to migrate from flat-fees to more flexible pricing schemes, considering the issue of fairness in a congestion-driven pricing scheme could be a tool to the efficient allocation of congestion costs. An additional component of the solution proposed is the possibility of reducing the strategic behaviour of users who attempt to manipulate the resource allocation to their advantage, for instance, in the context of auction-based allocation schemes. In a different but related context, axiom-based cost mechanisms have been considered in traffic-based cost allocation.
methods to avoid hurtful incentives for network players such as traffic resell in a centralized exchanger (Henriet & Moulin, 1996).

However, we also show that equity in the cost allocation problem (i.e., consumption price per unit of system requirement increases when priority gets higher) is not fully guarantee due to the way the prices are designed. This suggest two things: (a) in some cases, the cost allocation mechanism detects that the relative contribution per unit of system requirement of each user type is not consistent with priority ordering, and, (b) the axiomatic body should be complemented with a network administration action in order to avoid wrong price signals due to “scale” effects. We explore these issues further below.

MECHANISMS FOR EFFICIENT DISTRIBUTION

Aumann-Shapley (A-S) Prices in the Data Network Framework

Let us define a set of user types by \( N = \{1,\ldots,n\} \) whose possible vector of rates of arrivals \( \lambda \) is defined in the positive region of the \( n \)-Euclidean space \( \mathbb{R}^n, \mathbb{R}^n_+ \). One cost allocation problem is a pair \((F, \lambda)\), where \( \lambda \in \mathbb{R}^n_+ \) and \( F : \mathbb{R}^n_+ \rightarrow \mathbb{R}^n_+ \) is a function of costs due to losses from congestion, continuous and differentiable, where in the total absence of congestion \( F(0) = 0 \) (i.e., there is no inclusion of fixed costs). Additionally, let \( c \) be the class of all cost allocation problems with \( F \) and \( \lambda \) defined as above. A cost allocation mechanism is defined as a function that maps each problem \((F, \lambda)\) in \( c \) to the price vector \((P_1(F, \lambda), \ldots, P_n(F, \lambda))\) in \( \mathbb{R}^n \) (McLean & Sharkey, 1997).

Each price vector should fulfil several desirable properties, which are now described:

1. **Cost division**: states that the total income must exactly cover total costs.
2. **Monotonicity**: implies that the user types that have a non-negative arrival rate and that non-negatively contribute to congestion costs should have a non-negative price assigned.
3. **Additivity**: implies that if the cost function can be decomposed into two totally independent functions, then the price for congestion found according to the total cost function should be equal to the sum of the contributions derived from each of the component functions.
4. **Rescaling invariance**: indicates that if a linear change in the units of measurement occurs, a similar or proportional change takes place in prices, such that units of measurement do not affect the price mechanism.
5. **Consistency**: means that user types with the same effect on cost must bear the same price.

According to Billera & Heath (1982) and Mirman & Tauman (1982), there exists a unique mechanism that satisfies all of the aforementioned properties, and that mechanism is the Aumann-Shapley price mechanism. This mechanism is defined for each \( i \in N \), as:

\[
P_i^{AS}(F, \lambda) = \int_0^{\lambda_i} \frac{\partial F}{\partial \lambda_i}(t\lambda) dt \quad (1)
\]

Where \( F(0) = 0 \) and \( F \) is differentiable. Let the vector of arrival rates to the network \( \lambda \) consist of \((\lambda_1, \ldots, \lambda_n)\), where each \( \lambda_i \) represents the average quantity of information per unit of time that the user
of type i needs to have processed by a server. Since each λ, has the same unit of measurement, it is appropriate to represent the aggregate quantity of information per unit of time as \( \lambda = \sum \lambda_i \). The A-S prices reflect the sum of marginal costs along the ray \( t \lambda_i, 0 \leq t \leq 1 \).

**Shapley Value (SV) Approach to a Data Network Considered as a Cooperative Game**

Let us define a game (with transferable costs) with a set of \( n \) types of users or “participants.” Suppose that this game is related to the cost allocation problem \((F, \lambda)\), where \( F \) represents the total cost of losses due to congestion, as in the previous section, and \( \lambda \), similarly, represents the vector of processing demand rates, that is, the “jobs” that reach the system per unit of time (for example, requirements/sec.). Formally, let us define the set of game participants as \( N = \{1, \ldots, n\} \). The problem consists in efficiently distributing the congestion \( F \) among the \( n \) players, in a way that the distribution criteria meet a series of desirable properties (McLean & Sharkey, 1997).

Imagine a set of possible functions \( \psi \) from which we can extract a solution. Also define a function \( v : 2^n \rightarrow \mathbb{R} \) assuming that \( v(\emptyset) = 0 \) and \( v(N) = F \). For each of the possible subsets \( S \subseteq N \), \( v(S) \) represents the cost incurred by the subset or coalition \( S \) due to its “joint action” on the network.

If \( G_N \) denotes the space of all of the possible games in \( N \), then formally it is possible to describe the aforementioned class of functions \( \psi : G_N \rightarrow \mathbb{R}^n \) where for each \( v \in G_N \) and every \( i \in N \), \( \psi_i(v) \) is the distribution of costs that user type \( i \) receives when the effects of joint actions of the coalitions are quantified by \( v \). It is desirable that the class of functions \( \psi \) fulfills the following properties (McLean & Sharkey, 1994):

1. **Additivity:** means that the original problem may be separated into two distinct problems (for example, in the case of finite queuing models, the original total cost distribution problem can be divided into the problem of cost distribution due to service delays and problem of cost distribution due to a system lock-up). If this is the case, it is desirable that the sum of the solutions to the components of the problem is equal to the solution to the original problem.

2. **Efficiency:** refers to the fact that the sum of the charges assigned to each player in \( N \) should be the same as the total cost to be allocated.

3. **Symmetry:** means that if individuals contribute with the same costs to each coalition, then they should pay the same charge.

4. **Monotonicity:** This requires that if the player contributes a non-negative cost to the coalition, then the fee for that player should also be non-negative.

Among all of the possible functions \( \psi \), there exists only one function \( \phi : G_N \rightarrow \mathbb{R}^n \) that fulfills the aforementioned properties. Such function is the **Shapley value**. In order to define more specifically the Shapley value, it is necessary to define the set \( \Omega \) of \( n! \) possible orderings of the members in \( N \). For each user type \( i \in N \) and each ordering \( R \in \Omega \), define the set \( X_i(R) = \{ j_1, \ldots, j_{k_i}\} \) as the set of predecessors of \( i \) in the order \( R \), with \( i = j_{k_i} \). \( R = \{j_1, \ldots, j_n\} \). From the previous paragraph, recall that \( v \) is the function of costs derived from the cost distribution problem \( F \) and that it represents the cost or penalty assigned to a specific group of users. If we note \( v(S) = F(\lambda^S) \), where \( \lambda^S \in \)
\( \Re^a \) is a vector with \( \lambda^S_i = \lambda_i \) if \( i \in S \), and \( \lambda^S_i = 0 \) if \( i \notin S \), the Shapley value may be defined as follows:

\[
\varphi_i(v) = \sum_{R \subseteq \{1,2\}} \frac{1}{n!} \left[ F(\lambda^{X_i(R)}_{(n)}) - F(\lambda^{X_i(R \cup \{i\})}_{(n)}) \right]
\]

(2)

In this case, the price that user type \( i \) should pay for each of its service requirements per unit of time (for example, an Internet Service Provider, ISP) should be defined as:

\[
P^SV_i(F, \lambda) = \frac{\varphi_i(v)}{\lambda_i}
\]

(3)

A CONGESTION MODEL

We will now describe two models to represent an ISP that satisfies the Internet service application needs of a set of users, such as e-mail, video Web commercials, bank transactions, Web purchasing, and etcetera. In this case, two types or classes represent such a user group.

Each class can be characterized by an arrival rate \( \lambda_i \), and a service distribution rate \( \mu_i \), \( i = 1,2 \). The arrival process is described as a Poisson process, which means that the arrivals can be represented by a unique rate \( \lambda = \lambda_1 + \lambda_2 \), while the service times of each user are represented by an exponential distribution with mean \( 1/\mu_i \). Additionally, we assume that user type 1 has priority over user type 2. It is possible to construct transition diagrams for two specific cases: when service distribution rates are the same \( (\mu_1 = \mu_2) \), and the case when the service rates differ depending on user type \( (\mu_1 \neq \mu_2) \).

We assume also that the system has infinite queuing capacity, meaning that there is no open exclusion of user’s needs for those who arrive to the system, and arriving users are simply placed in a line to wait until they receive the corresponding service.

Model 1: Unique Service Rates

\( (\mu_1 = \mu_2 = \mu) \)

The aggregate characterization of this model using a transition diagram is identical to the queuing type M/M/1 (Wolff, 1989) and can be observed in Figure 1.

Model 2: Different Service Rates \( (\mu_1 \neq \mu_2) \)

In the case of different service rates, it is necessary to define a transition diagram considerably different than the one presented for model 1. This diagram is shown in Figure 2.

This transition diagram uses a state space that is represented in the form \{\((n_1, n_2, i)\)\}; \( n_j = 0,1,2,\ldots, j=1,2 \), where \( n_j \) is the

---

Figure 1. Transition states and rates in the presence of infinite capacity and unique service rates

---

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.
number of type j clients in the system. We define two types of clients, \( i = 1, 2 \). We also state that, when the system is empty, \( i = 0 \) (only the state (0,0,0) needs this assumption). Independent of whether or not the service rates are the same, the representation of the waiting times corresponds to the same expressions as in model 1, with the exception that here, in order to calculate the prices, we must take into account that \( \mu_1 \neq \mu_2 \). Maintaining the definition of \( \rho = \rho_1 + \rho_2 \), where \( \rho = \lambda/\mu \) (the global arrival and service parameters) and \( \rho_i = \lambda_i/\mu_i \), it can be seen that \( 1/\mu = P_1(1/\mu_1) + P_2(1/\mu_2) \), with \( P_i = \lambda_i/\Sigma \lambda_i \).

### Calculation of the Loss Function Due to Congestion

In order to calculate the prices, it is necessary to establish the loss function due to congestion, \( F \). For both of the previously described models, the function \( F \) is defined as \( F(\gamma, \lambda, \mu) = \gamma_1 \lambda_1 d_1 + \gamma_2 \lambda_2 d_2 \), where \( d_i \).
corresponds to user i’s average wait time in line and \( \gamma_i \) to the valuation of each service requirement (also for user type i). Following standard queuing theory (Wolff, 1989), these wait times are defined in the following way for systems that operate with non-pre-emptive priorities:

\[
d_i = \frac{\lambda_i E(S^2)}{2(1-\sum_j \rho_j)(1-\sum_i \rho_i)}
\]

Where \( \sum_i \rho_i < 1 \) and \( \sum_j \rho_j < 1 \), i = 1, 2, and S is a random variable that represents the service time, based on the exponential distribution. It can be easily shown that, for model 1, the cost function \( F(\gamma, \lambda, \mu) \) corresponds to the following expression:

\[
F(\gamma, \lambda, \mu) = \gamma_1 \frac{\lambda_1 \rho}{\mu(1-\rho_1)} + \gamma_2 \frac{\lambda_2 \rho}{\mu(1-\rho_2)(1-\rho)}
\]

where \( \rho_i = \lambda_i/\mu_i \) and \( \rho = \rho_1 + \rho_2 \).

For model 2, the function \( F(\gamma, \lambda, \mu) \) represented by:

\[
\gamma_1 \frac{\lambda_1 E(S^2)}{2(1-\rho_1)} + \gamma_2 \frac{\lambda_2 E(S^2)}{2(1-\rho_2)(1-\rho)}
\]

where \( \rho_1 = \lambda_1/\mu_1 \), \( \lambda = \lambda_1 + \lambda_2 \), \( \rho = \rho_1 + \rho_2 \), and additionally:

\[
E(S^2) = \int_0^1 \left[ \frac{\lambda_1}{\lambda} \mu_i e^{-\mu_i t} + \frac{\lambda_2}{\lambda} \mu_i e^{-\mu_i t^2} \right] dt = \sum_{i=1}^2 \frac{\lambda_i}{\lambda} \left[ \frac{2}{\mu_i^2} \right]
\]

A brief summary of expressions related to the calculation of prices is shown in the Appendix.

### MAIN FINDINGS

#### Shapley Value Under Equal Valuations in Model 1

In the case when the user types have equal values for their information (\( \gamma_1 = \gamma_2 \)) in model 1, and even more so when \( \rho \) is large, the prices according to the Shapley value rule are applicable (that is that \( P_1^{SV} > P_2^{SV} \) if and only if \( \lambda_2 > \lambda_1 \)). We can show by simply numerical computation that the requirement \( \lambda_2 - \lambda_1 > 0 \) becomes a sufficient condition for \( P_1^{SV} > P_2^{SV} \) to be true. However, this requirement can be relaxed when \( \rho \) tends to zero, because the prices in this case will tend to be equal.

This is the case when two types of users have, for instance, the same willingness to pay but have a different consumption habit. In this case, price ordering is not aligned with priority ordering: only when the consumption rate of the lowest priority user type is higher, the ordering of the prices turn out to be consistent with the priority ordering. The reason for this is quite simple: the user type with the highest priority might take advantage of some “scale effects,” because it splits its contribution to congestion cost (which is the highest) among the consumption units (i.e. \( \lambda_i \)). This might generate wrong incentives if we consider that \( \lambda_i \) could represent the number of type 1 consumers per unit of time, since consumption per unit of system requirement is perceived lower for the highest priority.

For the above mentioned case (i.e. when \( \gamma_1 = \gamma_2 \) and \( \lambda_1 > \lambda_2 \)) in model 1, and even more when \( \rho \) is large, a price
mechanism based on the weighted Shapley value should be applied. If it is the case that we have $\lambda_1 > \lambda_2$ and equal valuations, it is intuitive that the one that register a higher priority in network use should pay a higher price. This means that the average of incremental values doesn’t capture the relative “penalty” a user type should encounter for a have a more exclusive service. For that reason, it is necessary to introduce a correction mechanism to ensure that $P_1^{SV} > P_2^{SV}$. This mechanism converts the Shapley value outlined in point 2 into the weighted Shapley value.

The weighted Shapley value replaces the term $(1/n!)$ in the equation with the term $q^w(R)$ (MacLean & Sharkey, 1994). If we define $w = (w_1, \ldots, w_n)$ and $R = (j_1, \ldots, j_n)$, the term $q^w(R)$ is represented in the following expression:

$$q^w(R) = \frac{1}{w_j} \left[ \frac{w_{j+1}}{w_j + w_{j+1}} \right] \sum_{t=1}^{n} \frac{w_{jt}}{w_{jt} + \sum_{t=1}^{n} w_{jt}}$$

(8)

Table 1. A-S /SV prices with changes in $\lambda_1$, $\lambda_2$, $\gamma_1 = 2$, $\gamma_2 = 1$ and $\mu_1 = \mu_2 = 10$

<table>
<thead>
<tr>
<th>$\lambda_1$</th>
<th>$P_1^{AS}$</th>
<th>$P_2^{AS}$</th>
<th>$P_1^{SV}$</th>
<th>$P_2^{SV}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>0.042</td>
<td>0.030</td>
<td>0.042</td>
<td>0.031</td>
</tr>
<tr>
<td>1.50</td>
<td>0.070</td>
<td>0.051</td>
<td>0.069</td>
<td>0.052</td>
</tr>
<tr>
<td>2.00</td>
<td>0.105</td>
<td>0.078</td>
<td>0.104</td>
<td>0.079</td>
</tr>
<tr>
<td>2.50</td>
<td>0.152</td>
<td>0.115</td>
<td>0.150</td>
<td>0.117</td>
</tr>
<tr>
<td>3.00</td>
<td>0.217</td>
<td>0.169</td>
<td>0.214</td>
<td>0.171</td>
</tr>
<tr>
<td>3.50</td>
<td>0.318</td>
<td>0.256</td>
<td>0.314</td>
<td>0.260</td>
</tr>
<tr>
<td>4.00</td>
<td>0.506</td>
<td>0.428</td>
<td>0.500</td>
<td>0.433</td>
</tr>
<tr>
<td>4.50</td>
<td>1.031</td>
<td>0.933</td>
<td>1.023</td>
<td>0.941</td>
</tr>
</tbody>
</table>

A numerical example shows that when $\gamma_1 = 2$, $\lambda_1 = 2$, $\lambda_2 = 1$ and $\mu = 5$, the prices result in 0.56 and 0.68 for user types 1 and 2, respectively. If we would like the price of user type 1 to be greater, we can use weighted Shapley value with a weighting vector $w = (2, 1)$. The resulting prices would be 0.66 and 0.49, for user types 1 and 2, respectively. It is worthwhile to note that total income, with or without the weighting scheme, is equal and efficient.

**Shapley Value Prices Under Model 2**

Model 2 brings a higher complexity to the analysis. Numerical results show that, in some cases when the user with the lowest priority handles small size applications (e.g., $\mu_2$ tends to be large), the price ordering is consistent with the priority ordering.

But again, there are some other interesting cases where prices are put upside down according to priority orderings. When $\mu_1 > \mu_2$ and $\gamma_1 > \gamma_2$ (but neither $\gamma_1 > > \gamma_2$ nor $\mu_1 > > \mu_2$), then, $\lambda_1 >> \lambda_2$ implies that $P_1^{SV} < P_2^{SV}$. Under the same conditions, $\lambda_1 < \lambda_2$ implies that $P_1^{SV} > P_2^{SV}$. This suggests once again that the network administrator should apply the weighted Shapley value in some cases, in order to guarantee a consistent price ordering.

**Aumann-Shapley and Shapley Value Prices Under Model 1 with Different Valuations**

For model 1, if the arrival rates are the same and $\gamma_1 > \gamma_2$, it is guaranteed that $P_1^{SV} > P_2^{SV}$. The same result is valid for the AS prices. This simply indicates that, for the case when $\lambda_1 = \lambda_2$ and $\mu_1 = \mu_2$, the network administrator should assign the

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.
highest priorities according to the highest values (if he has the faculty to do so). In the case of the SV prices, suppose that \( \lambda_1=\lambda_2=2 \), \( \mu_1=\mu_2=5 \). If \( \gamma_1=2 \) and \( \gamma_2=1 \), we observe that \( P_{1\text{SV}}=1 \) and \( P_{2\text{SV}}=0.87 \). Even though the average time for the entire system, \( W_q=\rho/[(\mu(1-\rho))]=0.80 \) (following the model \( M/M/1 \)), is independent of any priority rule, the cost of losses due to congestion is minimized only if highest priority is assigned to the user type with the highest value. With the assignment of priorities following this criterion, \( F(\gamma,\lambda,\mu)=3.73 \). On the contrary, \( F(\gamma,\lambda,\mu) \) takes the value 5.87.

A similar result can be derived from the A-S prices. Calculating the A-S prices with the values from the previous example, we obtain \( P_{1\text{AS}}=1.01 \) and \( P_{2\text{AS}}=0.86 \). Table 2 presents the behaviour of the A-S and SV prices under the changes in arrival rates, when \( \mu_1=\mu_2=10 \), \( \gamma_1=2 \) and \( \gamma_2=1 \). Note that even though the values tend to be similar, the SV prices present a more homogeneous congestion cost distribution than the A-S prices, and consequently the price difference between the priorities is smaller.

For model 1, when \( \gamma_1=\gamma_2 \), the A-S prices are always the same, independent of the value that \( \lambda_1 \) and \( \lambda_2 \) take. The same property holds on the SV prices only if all of the parameters are the same.

We present a simple proof of this result for the A-S prices, which can be extended to any number of user types. We obtain the result for the SV prices by looking at equations at the Appendix.

The loss function due to congestion in the case of an infinite buffer depends on the user’s wait time. This function corresponds to \( F(\gamma,\lambda,\mu) = \Sigma\gamma\lambda_i d_i \). Using the relationship \( W_q=\Sigma(\lambda_i/\lambda)d_i \), derived from Little’s Law (Wolff, 1989), we observe that:

\[
F(\gamma,\lambda,\mu) = \sum_{i=1}^{n} \gamma_i \lambda_i d_i = \gamma \sum_{i=1}^{n} \lambda_i d_i
\]

\[
= \gamma W_q \lambda = \frac{\gamma \rho}{\mu(1-\rho)} \lambda = \frac{\gamma \rho^2}{(1-\rho)}
\]

(9)

\( W_q \) is the entire system’s average waiting time, where user type does not matter (this is the same result found for the case when \( M/M/1 \) with an unlimited queue). We next calculate the partial derivative of the cost function with respect to the arrival rate, which produces the following expression for user type 1:

\[
\frac{\partial F}{\partial \lambda_i} = \gamma \left[ \frac{2\rho(1/\mu)(1-\rho) + (1/\mu)\rho^2}{(1-\rho)^2} \right] = \gamma \left[ \frac{2\rho(1-\rho) + \rho^2}{(1-\rho)^2} \right] = \gamma \left[ \frac{2\rho - \rho^2}{(1-\rho)^2} \right]
\]

(10)

\[
= \gamma \left[ \frac{2\rho - \rho^2}{(1-\rho)^2} \right] \]

Table 2. AS prices with changes in \( \gamma_i \) when \( \lambda_i=1<<\lambda_2=40 \), \( \mu=50 \) (model 1)

<table>
<thead>
<tr>
<th>( \gamma_1 )</th>
<th>( \gamma_2 )</th>
<th>( P_{1\text{AS}} )</th>
<th>( P_{2\text{AS}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>1</td>
<td>0.0911</td>
<td>0.0911</td>
</tr>
<tr>
<td>1.1</td>
<td>1</td>
<td>0.0920</td>
<td>0.0911</td>
</tr>
<tr>
<td>1.2</td>
<td>1</td>
<td>0.0928</td>
<td>0.0912</td>
</tr>
<tr>
<td>1.3</td>
<td>1</td>
<td>0.0937</td>
<td>0.0912</td>
</tr>
<tr>
<td>1.4</td>
<td>1</td>
<td>0.0946</td>
<td>0.0912</td>
</tr>
<tr>
<td>1.5</td>
<td>1</td>
<td>0.0954</td>
<td>0.0912</td>
</tr>
<tr>
<td>1.6</td>
<td>1</td>
<td>0.0963</td>
<td>0.0912</td>
</tr>
<tr>
<td>1.7</td>
<td>1</td>
<td>0.0971</td>
<td>0.0913</td>
</tr>
<tr>
<td>1.8</td>
<td>1</td>
<td>0.0980</td>
<td>0.0913</td>
</tr>
<tr>
<td>1.9</td>
<td>1</td>
<td>0.0989</td>
<td>0.0913</td>
</tr>
<tr>
<td>2.0</td>
<td>1</td>
<td>0.0997</td>
<td>0.0913</td>
</tr>
</tbody>
</table>

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.
The A-S price for the first user type would be given by:

\[
P_{1}^{AS}(F, \lambda) = \int_{0}^{\lambda} \frac{\partial F}{\partial \lambda} d\lambda = \int_{0}^{\lambda} \frac{2t\rho - (t\rho)^2}{(1-t\rho)^2} dt
\]

\[
\gamma \int_{0}^{1} \frac{2t\rho - (t\rho)^2 + (1-1)}{(1-t\rho)^2} dt = \gamma \int_{0}^{1} \frac{1-(t\rho)^2}{(1-t\rho)^2} dt
\]

Setting \( u = 1-t\rho \) and \( du = -\rho dt \), the integral can be expressed as:

\[
P_{1}^{AS}(F, \lambda) = \gamma \int_{0}^{1} \frac{1-u^2}{u^2} du = \gamma \int_{u_0}^{u_1} \frac{u^2}{\mu_0} du = \gamma \int_{u_0}^{u_1} \frac{1}{\mu_0} du
\]

\[
= \gamma \frac{u}{\mu_0} \left[ 1 + \frac{1}{\mu_0} \right] = \gamma \frac{\rho}{\mu_0} \left[ 1 + \frac{1}{\mu_0} \right] = \gamma \frac{\rho}{\mu_0} \left[ 1 + \frac{1}{\mu_0} \right] = \gamma \frac{\rho}{\mu(1-\rho)}
\]

Note that \( \frac{\partial F(\gamma, \lambda, \mu)}{\partial \lambda} = \frac{\partial F(\gamma, \lambda, \mu)}{\partial \lambda_1} = \cdots = \frac{\partial F(\gamma, \lambda, \mu)}{\partial \lambda_i} \), from which it can be concluded that \( P_{1}^{AS}(F, \lambda) = P_{2}^{AS}(F, \lambda) = \cdots = P_{r}^{AS}(F, \lambda) \).

This result shows that the SV prices are more sensitive to parameter configuration than the A-S prices. This also shows that A-S prices are equal if valuations are equal, no matter what the intensity of use, the rate of arrivals and priority orders are.

**Aumann-Shapley Prices**

**Under Model 1 and Large Difference in Arrival Rates**

For model 1, even in cases when \( \lambda_1 < \lambda_2 \) or \( \lambda_1 > \lambda_2 \), it is possible to guarantee that \( P_{1}^{AS} > P_{2}^{AS} \) only when \( \gamma_1 > \gamma_2 \). Furthermore, an increase in \( \gamma_i, i=1,2 \), generates a linear increase in the A-S prices (and in total income): A numerical example of this result is seen in Table 2.

**Weighted A-S Prices**

Assume that \( \gamma_1 > \gamma_2 \) and \( \mu_1 = \mu_2 \) (but not \( \mu_1 > \mu_2 \)), it implies that \( P_{1}^{AS} < P_{2}^{AS} \). Under the same conditions, \( \mu_1 < \mu_2 \) implies that \( P_{1}^{AS} > P_{2}^{AS} \). Additionally, an increase in \( \mu_i, i=1,2 \) generates an exponential decrease in total congestion cost.

This result suggests that, for the network administrator, in some cases it is not sufficient to have the higher priority assigned to the user type with higher value, given that depending on the quantity of requirements that must be satisfied by the system (per unit of time), the assigned price for this user type can be lower than the price assigned to a lower priority, making the price mechanism not consistent with priority ordering.

In order to get \( P_{1}^{AS} > P_{2}^{AS} \), it is necessary to apply a weighting mechanism analogous to the one shown for the SV

<table>
<thead>
<tr>
<th>( w_1 )</th>
<th>( w_2 )</th>
<th>( w_3 )</th>
<th>( w_4 )</th>
<th>( w_5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_{1}^{W-AS} )</td>
<td>0.08</td>
<td>0.10</td>
<td>0.11</td>
<td>0.11</td>
</tr>
<tr>
<td>( P_{2}^{W-AS} )</td>
<td>0.11</td>
<td>0.09</td>
<td>0.08</td>
<td>0.08</td>
</tr>
</tbody>
</table>
prices. Following the definition of McLean, Pazgal, & Sharkey (McLean et al., 2004), we construct a sequence of “weights” \( W = \{w_i\}, i=1,\ldots,\infty \). Then the price weighting mechanism for the A-S prices (\( P_{w^{\text{AS}}} \)) is defined for each \((F, \lambda)\) by the following equation:

\[
P_{w^{\text{AS}}}(F, \lambda) = \int_0^1 \frac{\partial F}{\partial \lambda_i}(\gamma i, \lambda_i) w_i t^{\lambda_i-1} dt
\]

(12)

We construct a numerical example where \( \lambda_1 = \lambda_2 = 1, \gamma_1 = 2, \gamma_2 = 1, \mu_1 = 10 \) and \( \mu_2 = 5 \). In this case, \( P_1^{\text{AS}} = 0.081 \) and \( P_2^{\text{AS}} = 0.110 \). If we wish to apply the price mechanism \( P_{w^{\text{AS}}} \), it is necessary for the network administrator to define and apply weights to guarantee the coherency of the price mechanism. Using the results for the calculation of \( \frac{\partial F(\gamma, \lambda, \mu)}{\partial \lambda_1} \), but substituting \( \frac{\partial F(\gamma, t, \lambda, \mu)}{\partial \lambda_1} \), we observe that, for the case when \( w_2 = 1 \), the prices for different values of \( w_1 \) produce the results shown in Table 3.

### A-S Prices and Two Part Tariffs

A two-part fee based on the A-S prices not only takes into account the arrival rates but also the waiting times for service. For instance, let us assume a case where there exists a normalized unique service rate, \( \mu_1 = \mu_2 = 1 \). Changing the variable, \( \alpha = 1/\mu \), we rewrite, starting with equation (5), \( \frac{\partial F(\gamma, \lambda, \mu)}{\partial \lambda_1} \) and \( \frac{\partial F(\gamma, \lambda, \mu)}{\partial \lambda_2} \) and we calculate \( \frac{\partial F(\gamma, \lambda, \mu)}{\partial \mu} \).

Suppose now that \( \lambda_1 = 0.4, \lambda_2 = 0.2, \gamma_1 = 2, \gamma_2 = 1 \). The resulting prices are \( P_1^{\text{AS}} = 1.181, P_2^{\text{AS}} = 0.889, P_\alpha^{\text{AS}} = 0.65 \). A user type with low priority could maintain a bandwidth reserve to obtain a guaranteed service level equal to \( Q_0 \), without lowering the service quality of the high priority users. Define \( Q_i, Q_i = -d_i \), as the service quality of user type \( i \), \( Q_i \in (-\infty, 0] \).

In accordance to the prices calculated above, the user type with higher priority should pay a network access fee of 1.181 per requirement per unit of time, a higher price than the low priority users who pay only 0.889 for access to the same network. However, the consumption fee per “unit of network service,” \( P_\alpha^{\text{AS}} \) is the same (McLean & Sharkey, 1997). This demonstrates that the pricing structure reflects a mechanism where, despite the fact that the network service cost is the same for each user type, an access fee should be paid according to the preferred service level. In this way, the network administrator can guarantee access to those users who hold lower value or use intensity than others, offering them a lower access price. In the example, the lower priority users would pay an access fee of 0.889 for the service quality guaranteed by the network administrator and equal to \( Q_0 = Q_2 = -d_2 = -2.5 \). If a low priority user decides to become a member of the high priority users (keeping in mind that the user’s inclusion does not generate changes in the aggregate arrival rate), an additional fee of 1.181-0.889=0.29 should be paid for the increase in service quality \( \Delta Q = Q_1 - Q_2 = -1.00 - (-2.50) = 1.50 \).

For this model, it is important to see that the total income generated as a result of the access fees is equal to the income from network use. In the earlier example, observe that \( \lambda_1 P_1^{\text{AS}} + \lambda_2 P_2^{\text{AS}} = 1.181(0.4)+0.889(0.2) = 0.65 = \alpha P_\alpha^{\text{AS}} \). If \( \lambda_1 \) changes to 0.7, \( \lambda_1 P_1^{\text{AS}} + \lambda_2 P_2^{\text{AS}} \) it would be the same as \( 0.7(5.897)+0.2(4.86)= 5.10 = \alpha P_\alpha^{\text{AS}} \). This indicates that, in model 1 with the
normalized rate, the consumption fee is equal to $P_{n}^{AS} = F(\gamma, \lambda, \mu)/2$. This is, the expected contribution to total congestion cost by each user type $i$ is equal to $\lambda_iP_{\lambda_i}^{AS} + (\lambda_i/\lambda)\alpha P_{\alpha}^{AS}$.

**CONCLUSIONS**

It is important to note that in some of the cases presented it was assumed that the average service time of user type 1 was less than the average time of user type 2. Although presumably this is not usually the situation faced by an ISP, through these results we have attempted to establish a criterion for determining prices according to the priority order, and not to, for instance, the average requirement “size” (i.e., service time per requirement).

The alignment of the mechanism with priority ordering is not always directly obtained. The results show that sometimes it is necessary to implement a price weighting mechanism to guarantee a coherent set of prices per system requirement. Nevertheless, it is not always desirable to assign a higher price to higher priority. If we examine the case where $\gamma_i$ is the social value of information, even though the average “size” of user type 1’s information might be smaller, its value and service intensity can place it in the highest priority, while receiving a lower price. A similar conclusion can be reached from the example in the A-S Prices.

Although it might usually occur that users who have a higher value for their information may be willing to pay more for better service, we also showed cases where there are equal valuations and different priorities. In practice, it is worthwhile to ask, “When is this the case?” As mentioned before, $\gamma_i$ refers to the value, for user type i, of each requirement or application to be processed by the server. It is natural to think that the network administrator assigns higher priorities in accordance with higher values in order to optimize the allocation of network resources; but it is also possible that the user type is willing to pay for a certain priority, in which case the user type’s budget constrain should apply.

In the first case (priority assignment according to valuations), it is necessary to recall from the results that equal values imply equal A-S prices in model 1, independent of the arrival rate value. For model 2, these A-S prices are different. From the network administrator’s point of view, the assignment of higher priority to a user type may result in minimizing the total cost of losses due to congestion. For example, if $\gamma_1 = \gamma_2 = 2$, $\lambda_1 = 3.5$, $\lambda_2 = 1$, $\mu_1 = 6$ and $\mu_5 = 5$, a total cost of losses equal to 5.35 is generated. However, if the network administrator decides to assign higher priority to user type 2 and lower priority to user type 1, this cost would be 5.88, worsening the network’s performance.

The modified virtual lines, proposed by Shenker (Shenker, 1995), are a form by which it is possible to assign certain bandwidth reserves to a group of users that distribute this right according to a hierarchy system. This type of organization can lead various users of a certain type (that is, who share the same value) to form a new user type with a different number of system arrivals and requirements to be processed per unit of time. It is worthwhile to mention that this type of bandwidth reserve can alleviate social equity problems in access to network services.

Now, if the assignment of priorities depends on the users, they will simply choose to pay a price up to the point when the budget restriction binds. However, note
APPENDIX: CALCULATION OF PRICES

Aumann-Shapley Prices (A-S Prices)

For model 1, the calculation of the Aumann-Shapley prices (AS prices) for user type 1 takes the following form:

\[ P_i \triangleq (F, \lambda) = \frac{\gamma_i}{\mu \rho_1} \left\{ \frac{\rho}{1-\rho} \ln(1-\rho) - \rho \right\} + \frac{\gamma_i}{\mu} \left\{ \frac{A_i}{1-\rho} + \frac{A_i}{\rho} \ln(1-\rho) - \frac{A_i}{\rho} \ln(1-\rho) \right\} \]

\[ \text{Eq.}(A-1) \]

Where the \( A_i \)'s are calculated using the following matrix expression:

\[
\begin{bmatrix}
1 & 1 & 1 & 1 & A_1 \\
\rho^2 & 0 & \rho^2 & 0 & A_2 \\
\rho^2 + 2\rho \rho_1 & \rho^2 & \rho^2 + 2\rho \rho_1 & \rho_1 & A_3 \\
-(2\rho + \rho_1) & -2\rho & -(2\rho + \rho_1) & -2\rho \rho_1 & A_4
\end{bmatrix}
\]

\[ \text{Eq.}(A-2) \]

For model 2, the price assigned to user type 1 is obtained from the following expression:

\[ P_i \triangleq (F, \lambda) = \frac{\gamma_i}{2} C_i(\lambda, \mu) \int \frac{t}{1-\rho} dt + \frac{\gamma_i}{2} \rho_1 \lambda E(S^2) \int \frac{t^2}{1-\rho} dt \]

\[ + \frac{\gamma_i}{2} C_i(\lambda, \mu) \int \frac{t}{1-\rho} (1-\rho) dt - \frac{\gamma_i}{2} \frac{\lambda E(S^2)}{\mu_1} \int \frac{(\rho + \rho_1 - 2\rho)^2}{1-\rho} dt \]

\[ \text{Eq.}(A-3) \]

Where,

\[ C_i(\lambda, \mu) = \frac{\partial [\lambda E(S^2)]}{\partial \lambda_i} = (2\lambda_1 + \lambda_2) E(S^2) + \lambda_1 \lambda_2 \left[ \frac{1}{\mu_1} - \frac{1}{\mu_2} \right] \]

\[ \text{Eq.}(A-4a) \]

\[ C_i(\lambda, \mu) = \frac{\partial [\lambda E(S^2)]}{\partial \lambda_i} = \lambda_2 E(S^2) + \lambda_1 \lambda_2 \left[ \frac{1}{\mu_1} - \frac{1}{\mu_2} \right] \]

\[ \text{Eq.}(A-4b) \]

The price for user type 2, in both model 1 and model 2, is calculated using the cost division axiom:

\[ P_i \triangleq (F, \lambda) = \frac{1}{\lambda_2} \sum_{i=1}^{\lambda_2} \gamma_i \lambda_i d_i - \lambda_2 P_i \triangleq (F, \lambda) \]

\[ \text{Eq.}(A-5) \]
Shapley value prices (SV prices)

From the waiting time results for type M/M/1 queues, where for a unique user type we have \( F(\gamma, \lambda, \mu) = \gamma \lambda \rho / (1 - \rho) \), independent of the model to be used, it is possible to construct an incremental cost table by user type according to each possible order (Table A1):

<table>
<thead>
<tr>
<th>Arrival order</th>
<th>Order probability</th>
<th>Aggregate cost for user type 1</th>
<th>Aggregate cost for user type 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,2</td>
<td>1/2</td>
<td>( \gamma \rho_1^2/(1-\rho_1) )</td>
<td>( F(\gamma, \lambda, \mu) - \gamma \rho_1^2/(1-\rho_1) )</td>
</tr>
<tr>
<td>2,1</td>
<td>1/2</td>
<td>( F(\gamma, \lambda, \mu) - \gamma \rho_2^2/(1-\rho_2) )</td>
<td>( \gamma \rho_2^2/(1-\rho_2) )</td>
</tr>
</tbody>
</table>

The price calculations are made using weighted sums of the aggregate costs in Table A1, following the order probabilities presented. For model 1, these prices result in:

\[
P_{i}^{SV}(F, \lambda) = \frac{1}{2} \left[ \frac{\gamma \rho_1}{1-\rho_1} \right] + \frac{1}{2} \left[ F(\gamma, \lambda, \mu) - \gamma \rho_1^2/(1-\rho_1) \right] = \frac{1}{2} \left[ \frac{\gamma \rho_1}{1-\rho_1} \right] + \frac{1}{2} \left[ F(\gamma, \lambda, \mu) - \gamma \rho_1^2/(1-\rho_1) \right]
\]

For model 2, the expressions that represent the prices take the following form:

\[
P_{i}^{SV}(F, \lambda) = \frac{1}{2} \left[ \frac{\gamma \rho_1}{1-\rho_1} \right] + \frac{1}{2} \left[ F(\gamma, \lambda, \mu) - \gamma \rho_1^2/(1-\rho_1) \right] = \frac{1}{2} \left[ \frac{\gamma \rho_1}{1-\rho_1} \right] + \frac{1}{2} \left[ F(\gamma, \lambda, \mu) - \gamma \rho_1^2/(1-\rho_1) \right]
\]

It can be seen, for both model 1 and model 2: \( \lambda, \mu, P_1^{SV}(F, \lambda) = \lambda, \mu, P_2^{SV}(F, \lambda) = (1/2)(\gamma \rho_1^2/(1-\rho_1)) + (1/2)F(\gamma, \lambda, \mu) - (1/2)(\gamma \rho_1^2/(1-\rho_1)) = F(\gamma, \lambda, \mu) \), which indicates the efficient nature of the SV prices.
that neither the A-S prices nor the SV prices consider this type of constraint in the price calculation, and for this reason it is possible to obtain prices higher than user’s willingness to pay.

Although this problem can be alleviated through the use of budget constraints (Mirman & Tauman, 1982; McLean & Sharkey, 1994), we think that if the valuations represent willingness to pay, they may reflect such budget constraints, while real prices may give signals for capacity expansion. That is, a high congestion level may also reflect prices that no user is willing to pay. Congestion levels could be an appropriate sign of the need for network expansion in a way that acceptable prices are generated. The problem of optimal capacity expansion was considered by Mackie-Mason & Varian (1995a), who showed that capacity should be expanded when the income from congestion exceeds the cost of providing capacity. In reference to the models discussed in this article, it may be said that the decision to offer network services as a result of capacity expansion should occur when the prices established according to the network congestion level (in steady state) generate a loss in the number of users (given that the price is higher than their willingness to pay) whose costs exceed the cost of expansion. Users’ losses may also occur when the high prices generate incentives for users to construct their own networks - that is, the network structure is no longer sustainable.

The models developed here are based on infinite queuing capacity. One extension of the infinite queue model is to include an additional term in the cost function that depends on the probability of a system lock-up. In the presence of finite queues, the average system waiting time will continue to be the same weighted average of the times of each class, used for the infinite queues (given that the proportion of requirements accepted by the system for each user type is the same). However, the calculation of the waiting times cannot be carried out in the same way, given that the arrival of the requirements to be processed is not a Poisson process.

Under the assumptions of model 1, it is possible to set up the problem of finding the lock-up probability where the queue is defined as M/M/1 with a finite queue. From the transition diagram for model 2, we see that finding the system lock-up probabilities and the waiting times by class might be a virtually impossible exercise to carry out analytically, and for this reason it is necessary to implement methods of approximation in order to find reasonable solutions.

REFERENCES


(Eds.), *Internet Economics*. Cambridge, MA: MIT Press.


ENDNOTES

1 We are grateful with two anonymous reviewers for their insightful comments on this article. This work was done when the authors were affiliated to the Center for Studies in Management of Network Services, CGSR, at Universidad de Los Andes, Bogotá, Colombia.

2 We are not concerned with a broader look at fairness as in (Mitchell and Vogelsang, 1991) where three other important notions of fairness are presented: fairness as an economic right, the status quo approach, and fairness defined by due process and equal opportunity.

3 The term “transferable costs” refers to costs that can be divided among the $n$ types of users in any way.

4 Also note that $X_i(R) = \emptyset$ if $i = j$.

5 It is reasonable to assume that once a user has received service, independent of a user with higher priority entering the system, this service will not be interrupted until the action has been completed. In the queuing literature this type of priority is called *non-preemptive priorities* (see, for example, Wolff, 1989).

6 It is not possible to generate a two-part fee using the service rate as a parameter in the calculation of the partial derivatives, given that, as observed in the SV prices, when the service rate increases, total cost decreases. This is inconsistent with the expected behavior of any cost function as a result of varying its input parameters. It can be shown that taking the service rate as a parameter in the calculation of the partial derivatives of the A-S prices leads to an indetermination.
Fernando Beltrán (f.beltran@auckland.au.nz) holds a BSc in electrical engineering from the Universidad de Los Andes (Colombia), and an MSc and PhD in applied mathematics from the State University of New York at Stony Brook. Currently, he is a senior lecturer at the Information Systems and Operation Management Department, University of Auckland, New Zealand.

César García-Díaz (c.e.garcia@eco.rug.nl) holds a BSc in industrial engineering from the Pontificia Universidad Javeriana (Colombia), and an MSc in industrial engineering from the Universidad de Los Andes (Colombia). Currently, he is a PhD candidate at the Department of International Economics and Business at the University of Groningen, The Netherlands.
Mining Parallel Patterns from Mobile Users

John Goh, Monash University, Australia
David Taniar, Monash University, Australia

ABSTRACT

Mobile technology is a widely adopted technology and is used for mobile users to stay connected. Mobile data mining is an extension of data mining for the purpose of developing innovative ways to extract useful knowledge out from mobile users to the decision makers. Our proposed method, parallel pattern, provides a method for finding out similar decisions by mobile users. Parallel patterns are divided into physical parallel pattern and logical parallel pattern. Physical parallel pattern finds the similarities of physical movement decisions among mobile users. Logical parallel pattern finds the logical similarities of logical theme movements among mobile users. In physical parallel pattern, mobile users can only occupy one physical location at any one time while in logical parallel pattern, mobile users can occupy more than one logical theme at any one time and each static node can have more than one logical theme at any one time. Our performance evaluation shows that the method for mobile parallel pattern mining suits for the real world problem in both the physical and logical parallel pattern mobile mining paradigm.

Keywords: frequency pattern; knowledge discovery; mobile data mining; mobile knowledge extraction; parallel pattern

INTRODUCTION

Mobile technologies have benefited the society by connecting people without the barrier of physical distance. Mobile technologies have been widely adopted and are still on track of further adoption in terms of more services in single mobile equipment. Mobile technology is heading toward a bright future and research into this area is needed to create innovations in both the business side of the technology and technical side of the technology. Innovations in particular will provide new ideas that enable users to get jobs done or be entertained by using mobile technology without the barrier of physical distance in a way that is cost efficient and socially acceptable performance. Since mobile technologies have been widely adopted, there is an opportunity to find out the behaviour patterns of mobile users by observing their physical and logical movements (Goh & Taniar, 2004a; Wang et al., 2003).
Data mining (Agrawal & Srikant, 1994; Agrawal & Srikant, 1995; Eirinaki & Vazirgiannis, 2003; Han et al., 2000; Han et al., 1998; Han et al., 1999) is an area of research that focuses on the problem of how to retrieve useful knowledge out from a set of raw data collected from many sources. The volume of source data is often huge, and contains a lot of irrelevant variables or datasets. The art of data mining is to create an efficient method, which later can be translated into algorithm, which the method is smart enough to discard irrelevant dataset and provide a clear analysis of the source data.

Mobile data mining (Goh & Taniar, 2004a; Goh & Taniar, 2004b; Wang et al., 2003; Lim et al., 2003) is an extension of the data mining research field, which specifically combines the goal of both data mining along with the new environment and challenges posed in the mobile environment. Instead of looking at just market basket analysis (Agrawal & Srikant, 1994), which is a common area of analysis for classical data mining, mobile data mining looks into the patterns of the mobile users. The patterns of the mobile users (Goh & Taniar, 2004a) present a behavioural description of the mobile users.

This paper proposes a method for mining useful knowledge out from mobile users by using parallel pattern technique. Parallel patterns are movement decisions of mobile users that are of similar directions. It could be described in both physical and logical way. The aim of this method is to provide a highly accurate knowledge that is relevant to decision makers who are interested in finding out the behaviour of mobile users. This could range from physical movement decisions and logical taste selection decisions.

The paper is organised in the following manner. The next section provides the background knowledge necessary in understanding the concept of parallel pattern. These include the underlying concepts of how the mobile environment works and operates, how the mobile users can be identified in different ways and what are their advantages and drawbacks. The third section provides an account of related work surrounding the area of mobile data mining. This includes the frequency pattern and density pattern. The fourth section describes the proposed parallel pattern in detail. The parallel pattern section is subdivided into physical parallel pattern and logical parallel pattern. The fifth section provides the algorithm for the parallel pattern mining techniques. The sixth section provides performance evaluation for both physical and logical parallel pattern. Finally, the last section concludes.

BACKGROUND

The mobile environment will usually consist of static nodes and mobile nodes (Goh & Taniar, 2004a). A static node is a generic term to describe equipment in the mobile coverage area that the physical location of the mobile equipment generally does not change, or only changes a little. A mobile node is also a generic term to describe all equipment in the mobile coverage area in which the physical location of the entity moves from one place to another from time to time. An example of static node is the wireless network point and an example of mobile node is the wireless Personal Digital Assistant (PDA).

Figure 1 shows the representation of the mobile environment (Goh & Taniar, 2004a). It can be observed that each circle represents a mobile node, or mobile user. Each triangle represents a static node. Each arrow represents the direction of
movement of the mobile users. It can be observed that some mobile users can choose not to move at all, while some mobile users are moving constantly around the static nodes. However, static nodes are static and thus cannot be moved at all. It is also observed that mobile nodes are often staying close to a particular static node in order to receive certain services from the static nodes, such as using the bandwidth or simply just attracted to something that is offered by the particular static node.

In order to assist mobile data mining, the mobile nodes must have avenues to be identified. In general, they can be identified in two different ways that are: identification by means of physical address of the mobile equipment, or identification by the logical username of the mobile users. Both ways for identification of mobile equipment or mobile users have their own advantages and disadvantages. There is a trade-off in selecting any side of the identification method between cost and accuracy. Our philosophy is to be focused on the accuracy of the knowledge rather than the cost (Goh & Taniar, 2004b).

Using physical address, which is a globally unique address to identify particular mobile equipment, does the identification of mobile equipment. The usual convention for this type of address consists of vendor identification in the first part of the address space and the individual equipment identification code in the second part of the address space. The physical address is usually presented by using hexadecimal digits. For logical identification method, using the username of the mobile user will be accurate, as mobile user who carries more than one mobile equipment piece will still be classified into one but not more than

Figure 1. Mobile environment

![Diagram of mobile environment](image-url)
one entity. An example of physical address is: 00-03-47-27-D7-B8. On the other hand, an example of logical address is lastname.firstname@context.domain.country.

There is a need for the proposed method, parallel pattern, because of the weaknesses of the existing data mining methods. Existing data mining methods, such as association rules and sequential patterns, are transaction based, and orient towards the analysis of individual transactions. The closest existing method towards mobile data mining is geographical data mining, such as spatial data mining and spatial temporal data mining. However, mobile data mining orients towards the mining from the source data generated specific from mobile users, and therefore aims at dynamic movement of mobile users. Spatial temporal data mining, however, orients towards looking at the fixed physical location (areas) in the maps.

The most distinctive feature of mobile data mining is that it does not need to preload an accurate mapping of the geographical information, unlike existing methods, which require precision-based preload of geographical information to perform data mining tasks. In mobile data mining, all methods are tailored and view specific with either dynamic mobile users itself or static node itself, and analyse the ongoing pattern without a preloaded maps. It further extends itself in the logical paradigm by looking at mobile users in an abstract manner, which is not restricted by physical location boundaries, such as virtual movements of mobile users.

RELATED WORK

Recent related work done by others in the field of data mining includes the dense region finding (Yip et al., 2004), finding negative rules (Thiruvady & Webb, 2004), secure association rules sharing (Oliveira et al., 2004), discovery of maximally frequent pattern tree (Miyahara et al., 2004), spatial association rules (Koperski & Han, 1995) and time association rules (Barbara et al., 2004). It can be observed that the act of data mining is widely recognised and research in the field has tried to extend the data mining into different areas, such as time series and geographical. In our case, we have focused on the area of mobile data mining, which involves gathering source data from mobile users and performing data mining. Mobile data mining is still at an infant stage, with only a few works having been done. One significant work done by others is group pattern (Wang et al., 2003) which aims to find out the group characteristics of mobile users. Group pattern contains some inefficiency, including that a high amount of processing power is required, and weakness, such as only physical domains of the problem is being observed. Our previous work, frequency pattern (Goh & Taniar, 2004a) has aimed to solve this shortcoming.

Our previous related work in the field of mobile data mining is the frequency pattern. Frequency pattern (Goh & Taniar, 2004a) is a method in finding out the relationships between the mobile users by means of observing the frequency of communication in between these mobile users. This method was inspired by the fact that group pattern, another proposed method that examines the group relationships of mobile users by using physical distance, does not address the fundamental challenge of mobile environment, that is to stay in touch without distance barriers.

The frequency pattern (Goh & Taniar, 2004a) is designed to use frequency of
communication between each mobile user, coupled with the pre-specified criteria in order to further enhance the accuracy of frequency pattern. The pre-specified criteria configure the method in order to allow the decision maker to place different amount of emphasis on different parts of time zone concerned. For instance, in the business environment, mobile communications that occur most recently serve as the strongest indication of whether a relationship will be realised by having the pre-specified criteria to configure high emphasis on the most recent communications.

Frequency pattern therefore uses pre-specified criteria in order for the decision maker to place different emphasis on different time zones of the window size into consideration. Sometimes, it not the recent communication that needs to be taken into consideration, but a time somewhere just before the recent communication. The ability to adjust and place different emphasis at different parts of the time series allows better control for the decision maker. The use of frequency in calculating the confidence of group characteristics of mobile users allows better representation of group formations of mobile users regardless of their physical location. In this way, a family group that is spread throughout different parts of the world but remain constantly in touch with each other will be detected as a group.

Figure 2 shows a diagrammatic representation of frequency pattern. It can be observed that each mobile node has a specific relationship confidence calculated based on the frequency of communication between them. Each arrow represents the logical relationship between two mobile nodes. A, B, C and D are a set of mobile nodes in the mobile environment that have been determined by the algorithm as logically close to each other. The pre-specified criteria allow different emphasis on different parts of the time series. It can be observed that the calculated confidence at the left diagram, and when the confidence threshold of 0.6 is set, those relationships which have less confidence are discarded and the right diagram represents the final outcome of frequency pattern; that is, a list of group of mobile users that are frequently staying in touch with each other.

Finally, another related work in the mobile data mining research area is the density pattern. Density pattern is an ongoing work that examines the density of the mobile users. It is hypothesized that

![Figure 2. Frequency pattern](image-url)
the density of the mobile users must have some relationship with some other variables, such as the closeness of the mobile users or the potential similarity of decisions for these mobile users. More interesting methods will be produced from the density pattern mobile data mining area in the near future. Our related work, frequency pattern and density pattern aims to innovate the whole mobile data mining research in order to develop creative ways in finding out knowledge out from mobile users to support decision makers. In searching of useful knowledge out from mobile users, we aim to place first priority for the relevancy of the knowledge.

Figure 3 shows the fundamental concept of parallel pattern, which is to find the similarities of arrows that move in the similar directions. The objects in the picture can be physical locations or logical domains, while the square box indicates a starting position. The goal of parallel pattern is to find out the similarities in decisions, such as similarities of decisions to move or similarities of decision to change taste from one starting point to another among many mobile users. The result of this exercise is a better understanding of the behaviour of mobile users.

Our related work aims to address different parts of the nature of the problem faced in finding useful knowledge by from mobile users. Frequency pattern addresses the the issue of using frequency rather than physical distance in order to determine relative closeness. The density pattern addresses the intensity of the mobile users by knowing how intense a particular location or logical domain is of interest of

Figure 3. Concept of parallel pattern

Legend

- Mobile Nodes
- Static Nodes
- Direction of Movement
mobile users. The parallel pattern, on the other hand, addresses the interesting issue of movement decisions of mobile users and is a method that proactively seeks and determines the behaviour of mobile users.

**PROPOSED METHOD: PARALLEL PATTERN**

*Parallel pattern* is a new way to find out useful patterns from mobile users. Parallel pattern has the ability to determine the similarity of decision-making, which is described in the form of parallel patterns. A parallel pattern represents similarity of decision-making in which the similarity can be categorized into physical similarity or logical similarity.

A physical similarity represents the similarity of decisions to move in the physical dimension. For instance, a similarity of decisions among mobile users to move from one location to another in the same direction is the physical parallel pattern. A logical similarity represents the similarity of the decisions to switch from one theme to another. For instance, the similarity of a decision among mobile users to switch from watching a movie to playing sports is an example of logical parallel pattern.

**Physical Parallel Pattern**

*Physical parallel pattern* represents the similarity of the decisions to move from one physical location to another. In order to find the physical parallel pattern, the source data collected from mobile users must first be transformed into the action structure. The action is a structure of defining the change of state of one mobile user from one state to another. It can be represented as: *Mobile User 1 (State 1 → State 2).*

Let $M$ be a list of mobile users $\{M_1, M_2, \ldots, M_n\}$. Let the reference physical location be $L$. Let $A$ be a list of actions done by mobile users $\{A_1, A_2, \ldots, A_n\}$. Each action $A_n$ will be associated with a mobile user identification, location from and location to, such as $A_n = \{M_n, {L}_{from}, {L}_{to}\}$. The source data is first represented into action format. By observing the similarities of decision, such as there are a lot of mobile users that move from one particular physical location to another within the short period of window timeframe, the movement pattern of the action is determined as the physical parallel pattern. In other words, there is high confidence that the mobile users are likely to move from one particular physical location to another particular physical location.

The nature of time and space constraints in the real world has to be observed by the mobile user. A mobile user can only be present at one physical location at any one time. This is due to the fact that a mobile user occupies a physical location at any one time. The change of state can only occur from one state to another. The destination of the change of state must be different from the source in order for the change of state to be complete.

Physical parallel pattern aims to find out the similarities of surface movements among the mobile users in the mobile environment. For instance, it is of great interest for the decision makers to know the similarities of movement of trends, such as the similarity of movements for one physical location then to another physical location. It is also possible to determine the leader and the followers in the physical parallel pattern, by identifying which groups of mobile users have initiated the move and which group of mobile users then follows the similar movement.
Logical Parallel Pattern

Logical parallel pattern represents the similarity of a decision made by mobile users in terms of choosing logical decisions. These logical decisions are items such as themes, context, taste and interest. In this paper, a common name known as theme will be used in order to ease discussions. Logical parallel pattern aims to find out the logical patterns of mobile users in such a way that is independent from physical location constraints.

Let $S$ be a list of static nodes $\{S_1, S_2, \ldots, S_n\}$. Let $M$ be a list of mobile users $\{M_1, M_2, \ldots, M_n\}$. Let the reference physical location be $L$. Let $A$ be a list of actions done by mobile users $\{A_1, A_2, \ldots, A_n\}$. Each action $A_n$ will be associated with a mobile user identification, location from and location to, such as $A_n = (M_n, L_{\text{from}}, L_{\text{to}})$. Let $L$ be a list of logical themes $\{L_1, L_2, \ldots, L_n\}$. There can be more than one logical theme be present within a static node. Each static node is assigned with a particular set of logical themes based on the current situation. For example, $S_n = \{L_1, L_7, L_{10}\}$. It means that location $S_n$ contains logical themes from logical theme $L_1$, $L_7$, and $L_{10}$.

For instance, the similarity of change of theme for mobile users can occur when a group of mobile users starts to buy a particular kind of fashion design clothing in many different locations. Although there is change of state, these changes of states do not take the location of mobile users into consideration. These changes of state take account of change of theme and not physical location. Therefore, a logical parallel pattern that describes the change of state of logical themes can occur in many fashion stores in many different locations, within a city, country or continent.

Due to the fact that theme is an abstract and subjective term, a mobile user at any point in time will most likely be exposed to multiple themes. For instance, a mobile user currently watching a movie in a cinema could be experiencing the following themes: entertainment, movie, happiness and humour. Therefore, the logical parallel pattern pattern must take this real world situation into account and model logical parallel pattern to the real world model.

The action in the logical parallel pattern model will consists of the mobile user identification and a list of source theme and a list of destination theme. Movements from the source theme to the destination theme need not necessarily to be unique. An example of logical parallel pattern action will be: Mobile User 1 (Theme 1, Theme 2, Theme 3 $\rightarrow$ Theme 3, Theme 4, Theme 5). It should be noted that Theme 3 occurs in both source and destination, and this models according to the real world model as switch in the time series may involve change in theme but may still involve the same theme in both time points.

ALGORITHMS

The following represents the description and the core algorithms for finding out both physical and logical parallel pattern in the mobile environment. It is important to note that the two algorithms have a core difference, which is in the calculation of confidence part. Due to the different nature of physical and logical of real world, the algorithm is modelled in order to reflect the real world model. The nature of physical and logical concepts such as physical location and theme is incorporated into the algorithm.
The algorithm is implemented to define a window size in which transactions within a particular timeframe are taken into the system and actions formed based on the list of transactions. The purpose of window size is to keep a boundary of the problem in consideration in order to generate knowledge specific to that particular timeframe.

**Physical Parallel Pattern**

The algorithms to perform mobile data mining in search of physical parallel pattern can be done by the following steps.

*Step 1: Transform Dataset to Physical Actions*

In this step, all datasets must be transformed into the form of actions. Each action consists of mobile user identification and the identification of one source and one destination physical location. For example: Mobile User 1 (Source, Destination). After the transformation steps are completed, each action is validated against the rule of the real world model in which the movement of physical location must not happen to be the same one.

*Step 2: Select Window Size*

In this step, the window size is determined. Window size is a length defined that takes the defined amount of actions into consideration. The purpose of window size is to limit the number of actions taken into data mining and by selecting a specific desired size of window to suit the decision making purpose. The window size will affect the sensitivity of the algorithm and should be chosen carefully. For instance the window size of 10 could accommodate 10 sets of actions into the window for mobile data mining purposes. After the window size is selected, the window is populated with the latest physical actions.

*Step 3: Determine Unique Physical Actions*

In this step, each unique physical action is determined. For instance, if there are two occurrences of change of state from A to B in the window, regardless of which mobile user it is associated to, the frequency of occurrence is 2 in the window. Each of the unique actions is then assigned with a frequency number. For example: A → B: 2, B → C: 5, C → B: 2. It must be highlighted in this point that the direction of movement is significant as physical movement from Point A to Point B is totally unique from physical movement from Point B to Point A.

*Step 4: Calculate Confidence*

After each unique action is determined and the frequency is determined, the confidence of each unique action can be evaluated. The formula for confidence of a unique action is:

\[
\text{Confidence} = \frac{\text{Frequency}}{\text{Window Size}}.
\]

In this formula, it can be clearly seen that window size and frequency affects the confidence value, and is represented as a percentage value.

In order to achieve the high confidence, there must be many similarities of decision to move from and to a physical location in a very short period of timeframe. Increasing the window size will reduce the confidence found, having all other factors held constant. The increase in the frequency will directly influence the
increase of the confidence value with other factors held constant. Figure 4 shows the algorithm for calculating the confidence.

**Step 5: Generate Physical Parallel Pattern**

The final step in this mobile data mining process is the generation of physical parallel pattern. The generation of physical parallel pattern is by setting a confidence threshold. The confidence threshold is developed as a means for the decision maker to adjust the level of sensitivity that is required from the system. The higher the confidence threshold, the less the volume of output will be. At the same time, the knowledge generated will be of higher quality, as this knowledge must have high confidence in order to be displayed. On the other hand, with a low level of confidence threshold the volume of knowledge generated will be high and the quality of this knowledge may be low.

In order to generate these physical parallel patterns, the confidence of each unique action is taken and compared against the confidence threshold. If the confidence from the unique action is greater or equal to the confidence threshold, the unique physical parallel pattern will be taken for output to the decision maker. On the other hand, unique actions that do not meet the confidence threshold requirement will be discarded and not generated as physical parallel pattern. Figure 4 shows the algorithm for checking the confidence of a unique action.

**Logical Parallel Pattern**

The steps described below can achieve the algorithms for mobile mining in search of logical parallel pattern.

**Step 1: Transforming Data to Logical Actions**

First of all, the source data have to be transformed into the form of logical actions. The data structure for a logical action consists of mobile user identification and a set of source themes along with a set of destination themes. In order to model this method to the real world, these themes must be able to have duplicates and must be location independent but time dependent. The logical actions are then verified against its validity by ensuring the presence of themes in both sides of source and destination sets in the logical action.

**Step 2: Select Window Size**

The window size is then selected. This procedure is similar to the physical parallel pattern as the window size represents a timeframe, which the lists of logical actions are captured, and to be analysed. The higher the window size, the more logical actions will be captured and, therefore, may indicate better quality of knowledge. But there is a point whereby further increasing of window size will be inefficient as all the confidence values of unique actions that are happening frequently starts to be distorted due to the increase in window size.

**Step 3: Calculate Confidence**

In this step, the confidence for each logical action will be evaluated. The method for calculation in this step will be significantly different from the calculation method for physical parallel pattern because provisions have to be built in order to model the real world. In the real world, it is possible to have more than one theme at any one time, and when switching to
Figure 4. Algorithm for finding physical parallel pattern

Function Transform Dataset (ID, Source, Destination)
Verify ID, Source, Destination to be Valid
Check Source Not Equal to Destination
// One Mobile User Cannot Occupy 2 Physical Location At The Same Time

If Source = Destination Then {
    Remove Item; // Due to Inaccuracy
}
Else {
    Output ID, Source, Destination; // Valid Itemset
}
End If
End Function

Function Select Window Size (Input)
// Within Range is Greater Than 0 and Lesser Than Total Input;

If Input is Within Range Then {
    Permits Window Size;
}
Else {
    Deny Window Size;
}
End If
End Function

Function Determine Unique Physical Actions (Input)
Declare Array of Unique Actions;

For Each Input Do
    If Current Input Exist in Array of Unique Actions Then {
        Discard Current Input As it Already Exist;
    }
    Else {
        Add Current Input to Array of Unique Actions;
    }
End For
End Function

Function Calculate Confidence (Window)
For Each Unique Action in Window Do
    Unique-Action.Confidence = Unique-Action.Frequency/Window Size
End For
Return Confidence
End Function

Function Check Confidence (Confidence)
If Unique-Action.Confidence >= Confidence Threshold Then
    Display Unique-Action = TRUE
Else
    Display Unique-Action = FALSE
End If
End Function

Function Generate Physical Parallel Patterns (Unique Actions) {
For Each Unique Actions Do {
    If Check Confidence (Unique Action) = TRUE Then {
        Display Unique Action, Confidence;
    }
    Else {

another time point, the themes may be and may not be the same.

The formula for calculating the confidence value for logical parallel pattern has to take account into the multiple themes in one particular time point. Therefore, the confidence value will be calculated as follows: \( \text{Confidence} = \frac{\text{Frequency of Theme} \times \text{No of Themes}}{\text{Window Size}} \). The formula places particular attention towards each theme in providing an accurate measure of the confidence. Figure 5 provides the algorithm for calculating the confidence.

**Step 4: Generate Logical Parallel Pattern**

The final step in finding out logical parallel pattern will first require a confidence threshold to be set. After the confidence threshold has been set, the confidence level of each parallel logical pattern will be compared against the confidence threshold. Only those parallel patterns with confidence \( \geq \text{confidence threshold} \) will be taken into consideration for final output.

**CASE SCENARIO**

This section presents a case scenario in order to work as an example for mining parallel pattern from the set of source data generated by a specific static node. In a mobile environment, there are a set of static nodes and mobile nodes. The movement of mobile nodes to and from the static nodes are recorded down as the source data. The main difference between physical parallel pattern and logical parallel pattern is that, in physical parallel pattern, each mobile user can only occupy a single physical location at any one time, while in logical parallel pattern each mobile user can occupy more than one logical theme at any one time.

**Physical Parallel Pattern**

The source data can be gathered by the records created by the static nodes from the history of movements among mobile users within that particular static node. Once these are gathered, the source data are related particular to a specific static node.

Figure 6 describes the list of source data extracted directly from a particular static node. In this case, the static node in concern is labelled as A. The movement of the users are recorded down, and the movements that are not within the specific window size are removed. In this figure, the window size is 10. The movement of the users is also known as action, in which each action represents a change of state, in the physical parallel pattern, change of physical location from one physical location to another physical location.

In Figure 7, the list of unique actions from the previous dataset is found. The previous dataset is further filtered in which only unique actions are presented. Each unique action represents a unique movement from one state to another within the specific window size. The unique action is critical in finding both physical and logical parallel patterns. Each unique action can be actioned by more than one mobile user. The frequency of this action within the particular window size is then obtained.

In Figure 8, the frequencies of each unique action are found. The confidence of each unique action is determined by the percentage of frequency of occurrence within the particular window size. In this case, the window size is 10. Therefore, the
Figure 5. Algorithm for calculating confidence in logical parallel pattern

<table>
<thead>
<tr>
<th>Function Transform Dataset (ID, Source, Destination)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verify ID, Source, Destination to be Valid</td>
</tr>
<tr>
<td>// One Mobile User CAN Occupy 2 Logical Theme At The Same Time</td>
</tr>
<tr>
<td>If (ID, Source, Destination are Invalid) Then {</td>
</tr>
<tr>
<td>Remove Item; // Due to Inaccuracy</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>Else {</td>
</tr>
<tr>
<td>Output ID, Source, Destination; // Valid Itemset</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>End If</td>
</tr>
</tbody>
</table>

| End Function                                        |

<table>
<thead>
<tr>
<th>Function Select Window Size (Input)</th>
</tr>
</thead>
<tbody>
<tr>
<td>// Within Range is Greater Than 0 and Lesser Than Total Input;</td>
</tr>
<tr>
<td>If Input is Within Range Then {</td>
</tr>
<tr>
<td>Permits Window Size;</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>Else {</td>
</tr>
<tr>
<td>Deny Window Size;</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>End If</td>
</tr>
</tbody>
</table>

| End Function                                        |

<table>
<thead>
<tr>
<th>Function Determine Unique Logical Actions (Input)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Declare Array of Unique Actions;</td>
</tr>
<tr>
<td>For Each Input Do</td>
</tr>
<tr>
<td>If Current Input Exist in Array of Unique Actions Then {</td>
</tr>
<tr>
<td>Discard Current Input As it Already Exist;</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>Else {</td>
</tr>
<tr>
<td>Add Current Input to Array of Unique Actions;</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>End For</td>
</tr>
</tbody>
</table>

| End Function                                        |

<table>
<thead>
<tr>
<th>Function Calculate Confidence (Window)</th>
</tr>
</thead>
<tbody>
<tr>
<td>For Each Unique Action in Window Do</td>
</tr>
<tr>
<td>Confidence = (Frequency of Theme/No of Themes)/Window Size</td>
</tr>
<tr>
<td>Return Confidence</td>
</tr>
<tr>
<td>End For</td>
</tr>
</tbody>
</table>

| End Function                                        |

<table>
<thead>
<tr>
<th>Function Check Confidence (Confidence)</th>
</tr>
</thead>
<tbody>
<tr>
<td>If Unique-Action.Confidence &gt;= Confidence Threshold Then</td>
</tr>
<tr>
<td>Display Unique-Action = TRUE</td>
</tr>
<tr>
<td>Else</td>
</tr>
<tr>
<td>Display Unique-Action = FALSE</td>
</tr>
<tr>
<td>End If</td>
</tr>
</tbody>
</table>

| End Function                                        |

<table>
<thead>
<tr>
<th>Function Generate Logical Parallel Patterns (Unique Actions)</th>
</tr>
</thead>
<tbody>
<tr>
<td>For Each Unique Actions Do {</td>
</tr>
<tr>
<td>If Check Confidence (Unique Action) = TRUE Then {</td>
</tr>
<tr>
<td>Display Unique Action, Confidence;</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>Else {</td>
</tr>
<tr>
<td>Discard Unique Action;</td>
</tr>
<tr>
<td>}</td>
</tr>
</tbody>
</table>

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.
In Figure 9, the confidence of each unique action is determined. The confidence threshold is determined to filter out all those unique actions that have low confidence. Those unique actions that have high confidence will then be determined as parallel pattern, in this case, physical parallel pattern.

Logical Parallel Pattern

The source data can be gathered by the records created by the static nodes from the history of movements among mobile users within that particular static node. Once these are gathered, the source data are related particular to a specific static node.

Figure 10 represents the source data in the logical parallel pattern environment. It can be seen that for each movement of the mobile user, there can be movements to multiple different destination states from the same source state. This is due to the fact that logical parallel pattern is based on logical themes, in which each static node can have more than one theme at any one time.

In Figure 11, the list of unique actions from the previous dataset is found. The previous dataset is further filtered in which only unique actions are presented. Each unique action represents a unique movement from one state to another within the specific window size. Each unique action can be actioned by more than one mobile user. The frequency of this action within the particular window size is then obtained.

In Figure 12, the frequencies of each unique action are found. The frequency of each unique action is determined by the confidence of each of the unique actions are determined.

In Figure 13, the confidence of each unique action is determined. The confidence threshold is determined to filter out all those unique actions that have low confidence. Those unique actions that have high confidence will then be determined as parallel pattern, in this case, physical parallel pattern.

Logical Parallel Pattern

The source data can be gathered by the records created by the static nodes from

<table>
<thead>
<tr>
<th>User</th>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>3</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>4</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>5</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>6</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>7</td>
<td>A</td>
<td>D</td>
</tr>
<tr>
<td>8</td>
<td>A</td>
<td>D</td>
</tr>
<tr>
<td>9</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>10</td>
<td>A</td>
<td>E</td>
</tr>
</tbody>
</table>

Figure 6. Source data

<table>
<thead>
<tr>
<th>Unique Action</th>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>3</td>
<td>A</td>
<td>D</td>
</tr>
<tr>
<td>4</td>
<td>A</td>
<td>E</td>
</tr>
</tbody>
</table>

Figure 7. Unique actions

<table>
<thead>
<tr>
<th>Unique Action</th>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>3</td>
<td>A</td>
<td>D</td>
</tr>
<tr>
<td>4</td>
<td>A</td>
<td>E</td>
</tr>
</tbody>
</table>

Figure 8. Frequency of unique actions

<table>
<thead>
<tr>
<th>Unique Action</th>
<th>Source</th>
<th>Destination</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>B</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>C</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>A</td>
<td>D</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>A</td>
<td>E</td>
<td>1</td>
</tr>
</tbody>
</table>
PERFORMANCE EVALUATION

Performance evaluations are all tested on IBM platform. The capability of the processor is Pentium IV with 384MB of random access memory with at least 1GB of hard disk storage capacity. Random data sources are drawn from random.org, which is a site that generates random numbers by using the atmospheric noise as the seed. Atmospheric noise is unpredictable and is definitely random and generation of random numbers by using this as the seed will provide a true randomness of number, that is, each number will have equal chance of occurrence.

Efforts have been spent in collecting real-life datasets for mobile environments. However, due to the fact that mobile data mining is in the beginning state, real-life datasets are not yet available. Group patterns (Wang et al., 2003), another recent method in the field of mobile data mining developed by others, have also used synthetic datasets for performance evaluations. The advantage of synthetic dataset is that performance evaluators actually knows what is happening in the dataset, as a formula or patterns are setup for the dataset generation. The disadvantages of using real-life dataset are that often more rigorous testing and data inconsistencies removal have to be done.
so that it is fit for mining and, secondly, random noise, which occurs randomly may be added to the dataset, distorting itself at the very beginning. In order to evaluate the performance and not to reduce the relevancy, three datasets are developed based on statistical concepts.

First dataset, Dataset \( A \), is a random dataset, whereby each mobile user arrivals occur randomly. The second dataset, \( B \), is a random dataset with reoccurrence for every 10 transactions. The third dataset, \( C \) is a random dataset with reoccurrence for every 20 transactions. The development of datasets is based on the belief that a normal dataset has to be present with special datasets, which contains patterns compared against the normal dataset. Dataset \( A \), being the normal dataset, is suitable, over a long period of time, the arrival distribution of mobile users should get closer and closer to the normal distribution. If there are significant patterns, then the dataset should be similar to Dataset \( B \) and Dataset \( C \).

The performance and evaluation is about analyzing the relationships between the variables and constants in the system. In this mobile data mining system, the factors involved are: the number of mobile users, the geographical size covered, the number of context taken into account, the number of output found, the accuracy of output, the size of the window, the level of various thresholds, the explicit and implicit cost of the methods and finally the time taken from the point of input to system output of knowledge.

### Physical Parallel Pattern

Figure 14 shows the performance parameters of physical parallel pattern. The table compares the different test results of different window sizes and different number of physical parallel patterns obtained. A representation of this table in a graph can be found in Figure 15. The window size is the size of window in which the number of transactions can be obtained. The Dataset \( A \) represents a random source data, in which each transaction has an equally chance of occurring. The Dataset \( B \) represents a modification from a random

**Figure 13. Confidence of unique actions**

<table>
<thead>
<tr>
<th>Unique Action</th>
<th>Source</th>
<th>Destination</th>
<th>Frequency</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>B</td>
<td>4</td>
<td>40%</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>C</td>
<td>3</td>
<td>30%</td>
</tr>
<tr>
<td>3</td>
<td>A</td>
<td>D</td>
<td>2</td>
<td>20%</td>
</tr>
<tr>
<td>4</td>
<td>A</td>
<td>E</td>
<td>1</td>
<td>10%</td>
</tr>
</tbody>
</table>

**Figure 14. Performance parameters of parallel pattern**

<table>
<thead>
<tr>
<th>Window Size</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dataset A</td>
</tr>
<tr>
<td>20</td>
<td>6</td>
</tr>
<tr>
<td>40</td>
<td>6</td>
</tr>
<tr>
<td>60</td>
<td>6</td>
</tr>
</tbody>
</table>
source data in which repetitions are performed for every 10 transactions. The Dataset C represents an even less random source data, in which a modification from a random source data in which repetitions is performed for every 20 transactions.

Figure 15 shows the performance result of the number of parallel pattern found compared to the size of the window. It can be seen that for a random dataset, which is Dataset A, the number of parallel pattern found remains the same regardless of the size of the window. The second observation is that for non-random datasets, which are Dataset B and Dataset C, the number of parallel pattern reduces as the window size increases. The lesson learnt from this performance and evaluation is that the size of the window must be adjusted according to the characteristics of the source data in order to find the result. This is due to the fact that the calculation of confidence using the formula (count/window size) has the window size located at the denominator. On the other hand, other mechanisms can be developed to dynamically determine the size of the window, which is outside the scope of this paper. Dataset A is a random sample, while Dataset B contains repetitions from the random sample while Dataset C contains even more repetitions from the random sample. Therefore, Dataset A, B, C are getting more and more less random continuously.

Figure 16 shows the relationship between the average physical confidence and the confidence threshold. It can be observed that the line starts at 100 and stops at 0. The non-linear behaviour of the line shows the non-randomness of the source data in which for each step of increasing level of confidence threshold, the physical confidence value reduces at different rates. The performance evaluation shows us that it is important to observe the source data in order to strike a balance between the desired accuracy and desired sensitivity. Each dataset at different size and different randomness will require a different threshold value in order to be optimum for the decision maker.

Figure 17 shows the relationship between the physical confidence verses the
Figure 16. Physical confidence and confidence threshold

![Graph showing average physical confidence and confidence threshold.]

Figure 17. Physical confidence versus number of unique actions

![Graph showing physical confidence vs. no of unique actions.]

number of unique actions. A unique action is a non-repetitive movement of state within a particular window size. It can be observed as the physical confidence is always greater with a small size of window and the physical confidence gradually reduces as the window size increases. This indicates that window size actually dilutes the confidence calculation and, therefore, a point will be reached where further increase will lead to low confidence and further decrease will lead to poor quality of output. This is because of confidence = (frequency/window size). In a random sample arrival situation, since window size is at the denominator, the confidence will be reduced as the window size increases. The window size is the size of the window, also known as the duration of when the samples are collected from the mobile users. Different sizes of window size will show different amounts of sample data collected. Figure 18 shows the performance result represented as a table.
for the physical parallel pattern found versus different window sizes.

**Logical Parallel Pattern**

Figure 19 shows the performance evaluation of the logical parallel pattern found versus the size of the window. It is found that the performance evaluation reacts similarly to physical parallel pattern, in which, as the number of window size increases, the number of parallel pattern found, regardless of whether it is physical or logical parallel pattern, will decrease. Therefore, it can be learned that determining correct window size is essential to producing useful knowledge to decision makers. Dataset A contains random sample while Dataset B contains more repetition thus lesser random sample and finally Dataset C contains even more repetition and thus the least amount of randomness in Dataset C. The chart is derived with the same table represented as Figure 15 as it is assumed that each static node comes with one logical theme.

**Figure 18. Table of results for number of parallel pattern versus different window sizes**

<table>
<thead>
<tr>
<th>Unique Actions</th>
<th>Window Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>0.1000</td>
</tr>
<tr>
<td>2</td>
<td>0.2000</td>
</tr>
<tr>
<td>3</td>
<td>0.3000</td>
</tr>
<tr>
<td>4</td>
<td>0.4000</td>
</tr>
<tr>
<td>5</td>
<td>0.5000</td>
</tr>
<tr>
<td>6</td>
<td>0.6000</td>
</tr>
<tr>
<td>7</td>
<td>0.7000</td>
</tr>
<tr>
<td>8</td>
<td>0.8000</td>
</tr>
<tr>
<td>9</td>
<td>0.9000</td>
</tr>
<tr>
<td>10</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

**Figure 19. Logical parallel pattern versus window size**
Figure 20 shows the relationship between logical confidence and the number of themes present in the actions. A static node can have more than one logical theme, as logical themes are logical attributes related to a particular static node. For instance, a static node in a shopping mall will have a logical theme of {fashion, shopping, grocery, entertainment}. It can be observed that there is a linear relationship between the logical confidence and the number of themes. The more the number of themes, higher the confidence will be. The second observation from this performance and evaluation process is that the higher the number of interest theme in over the total theme, the lower the confidence calculated. The interest themes are the number of themes that are of interest, such as when there are four themes — sports, leisure, academic, and movie — and the interest theme for the decision maker can be sports and leisure only.

Figure 21 shows the relationship between average logical confidence and the confidence threshold. The single line is derived from the average of multiple

Figure 20. Logical confidence versus number of themes

![Logical Confidence vs. No of Themes](image)

Figure 21. Logical confidence versus confidence threshold

![Average Logical Confidence vs. Confidence Threshold](image)
amounts of tests. Due to the variability of the source data, it can be seen that in this case, the variation of the behaviour of mobile users’ changes as the confidence threshold gets to a higher value.

Figure 22 shows the relationship between the logical confidence and the theme of interest. It can be observed that there is a linear relationship between these two variables. It can also further be observed that with increasing size of window, the number of logical confidence will drop linearly in different level of gradient. Therefore, the higher the theme of interest will lead to higher level of confidence to be calculated. This represents a correct real world model, as the aim for calculation of confidence is to have high confidence for unique logical actions in which the theme is of interest and receives
high frequency. The window size is the duration for which samples will be collected from the mobile users.

Figure 23 shows the highest found confidence level against the number of context in a logical parallel pattern-mining paradigm. It can be seen that for an environment with a small number of window size (window size 10), the highest confidence number found is initially very high. This is due to the low amount of unique context, resulting strong differentiation and therefore, strong highest confidence. As the number of context increases, it can be seen that the highest confidence found gradually drops, at number of context = 15, the highest confidence found is actually the same with the highest confidence found with window size 30. The value in the bracket in this graph is the value of window size in the unit of time in seconds.

Above all, the trend of the highest confidence found versus the number of context is clear. The highest confidence found for window size 10 is often higher than the highest confidence found for window size 20 and window size 30 except in few cases. Same trend for window size 20 and window size 30, where window size 20 always have a higher or equal amount of confidence than window size 30. Therefore, a conclusion can be found from this performance and evaluation exercise that the highest confidence found in the mining process will be determined by the window size. Higher window sizes will always returns a lower highest confidence. On the other hand, as the number of context increases, the highest confidence found decreases. Therefore, both window size and the number of possible unique context determine highest confidence count.

**Physical versus Logical Parallel Pattern**

Figure 24 shows the number of parallel pattern versus the number of unique actions with the window size of 10. The unique action along the x-axis represents the number of totally different types of actions that can be found within the window size. The purpose of using unique action is to measure the relationship between the numbers of unique actions towards the

![Figure 24. Number of parallel pattern versus number of unique actions](image)
number of parallel pattern that can be found. By right, the more the unique actions, the more the number of parallel patterns should be found, because each unique action is a potential pattern. The only remaining condition for it to become a parallel pattern is to have frequent occurrence of the unique action. A comparison is made between the physical parallel pattern and the logical parallel pattern. It can be observed that there are changes in the relationship in between the two, that is, sometimes one is higher than another and other times, vice versa. The values for these two lines are closely drawn together as the window size is still small and there is not much room for movement.

Figure 25 shows the number of parallel pattern versus the number of unique actions with the window size of 20. Along the x-axis, the values are represented from 1 to 8, which represents different amount of

---

**Figure 25. Number of parallel pattern versus number of unique actions**

![Graph showing No of Parallel Pattern vs. No of Unique Actions (Window Size = 20)](image)

**Figure 26. Number of parallel pattern versus number of unique actions**

![Graph showing No of Parallel Pattern vs. No of Unique Actions (Window Size = 30)](image)
unique actions. There can be many similar movements within the window. Each particular similar movement is a unique action. A comparison is made between the physical parallel pattern and the logical parallel pattern. It can be observed that the relationship between these two lines is non-linear and contains start and end points very similarly coupled together while data lines in the middle remains the relationship of physical parallel pattern drawing higher number of parallel pattern than logical parallel pattern.

Figure 26 shows the number of parallel pattern versus the number of unique actions with the window size of 30. A comparison is made between the physical parallel pattern and the logical parallel pattern. It can be observed that the relationship between physical parallel pattern and logical parallel pattern is not linear. In this case, the logical parallel pattern always draws a higher number of parallel patterns while the physical parallel pattern always draws somewhat lesser amount of parallel pattern in the beginning, but almost the

Figure 27. Number of physical parallel pattern versus number of unique actions

![Figure 27](image1)

Figure 28. Number of logical parallel pattern versus number of unique actions

![Figure 28](image2)
same as the logical parallel pattern in the end. It can also be observed that the trend is that increasing number of unique actions will increase the number of parallel patterns found.

Figure 27 shows the relationship between the numbers of physical parallel pattern versus the number of unique actions. It can be observed that the trend of increasing numbers of physical parallel pattern with an increase of number of unique actions, without a linear relationship. It can also be observed at the beginning of the graph, the values crosses with each other that indicate the small amount of differences in the beginning and, therefore, draw a very close result. It can also be observed that the lower the window size, the higher the confidence drawn.

Figure 28 shows the relationship between tests involving different amounts of window size on the number of logical parallel pattern found versus the number of unique actions. It can be observed that the number of logical parallel pattern increases as the number of unique action increases, but at different rates. But the overall trend is still to increase and never decrease. It can also be observed that the lower the window size, the higher the amount of logical parallel pattern is found.

It can also be observed that the result for window size = 20 and window size = 30 is relatively near in the middle of the graph. It indicates that the optimal window size will be between 10 to 20 and not 20 to 30, as the difference that it makes between 20 to 30 is not much. The optimal window size in this condition is estimated to be at window size 15.

Figure 29 shows the relationship of response between the number of physical parallel pattern and logical parallel pattern against the threshold value. It can be seen that due to variability of the data, physical and logical can respond differently. But one fact that can be learned from this performance and evaluation is that the threshold value correctly controls the sensitivity of the result. For example, a threshold value of 100% displays no parallel pattern at all, which will not help the decision maker. On the other hand, a threshold value of 0% will display all unique

Figure 29. Number of parallel pattern versus threshold value
parallel patterns in the system, which does not help the decision maker either.

There is a need to strike a balance in between the two ends of threshold values. It is likely for different sets of data to have different variability of curves and, therefore, different number parallel patterns. The decision maker must use correct judgment before deciding to follow with the given parallel pattern, or simply perform another test again with another level of threshold values. By generating the result in a graphical form, the result can be easily compared and decided upon.

CONCLUSION AND FUTURE WORK

This paper has introduced the challenging problem in the mobile environment. The interesting opportunities of mobile data mining have also been discussed. A few of our previous related works were given as a guide on how to find out interesting knowledge from mobile users in the mobile environment. The fact that mobile equipment are designed to be kept along with the mobile user at most of the times to reduce the physical barrier of communication presents a golden opportunity for mobile data miners. Not only does it provides new opportunities to find out interesting knowledge, but it also provides the accuracy of the knowledge, as it will encompass the whole life sequence of the mobile users.

The physical and logical parallel patterns are introduced in detail and their performance and evaluation have also been done in detail. It can be concluded that there is no one method better than another, but the method must be chosen by the decision maker. If the decision maker has to know the physical movement trend, then physical parallel pattern will be the one to use and if the decision maker wishes to know the global trend of theme of movement, the logical parallel pattern have to be used. Both patterns are different in terms of its modelling of the real world environment. In conclusion, physical and logical parallel pattern will definitely help the decision maker in making a more informed decision through the source data collected from mobile users.

Some future work in this area includes examining the relationship among parallel patterns. Parallel pattern discovers the similarities of movement among mobile users in the same direction. There is an opportunity to examine the relationship among these parallel movement decisions in such a way that the result will be such as when mobile users moves from A to B, there is also a strong tendency for another group of mobile users to move from A to C, at a different direction. Knowing such behaviour will provide even clearer representation of the mobile users.

REFERENCES


John Goh is a PhD student at the School of Business Systems, Faculty of Information Technology, Monash University, Australia. He is conducting research in the field of mobile data mining, which focuses on extracting interesting patterns and knowledge out from raw data collected from mobile users, including mobile phones and personal digital assistants. Since his enrolment into a research degree in late 2003, he has published a number of papers in the area of mobile data mining.

David Taniar holds bachelor’s (Honors), master’s, and PhD degrees — all in computer science, with a particular speciality in databases. Dr. Taniar’s research areas now expand to data mining. He has published more than 100 research articles that have appeared in international journals and conference proceedings. He has also published a number of books, including the forthcoming book on object-oriented Oracle. Dr. Taniar is now with the School of Business Systems, Faculty of Information Technology, Monash University, Australia. He serves on the editorial board of a number of international journals. He was elected as a fellow of the Institute for Management Information Systems (FIMIS).
Data Communications and Networks are no longer peripheral issues best left to the IT team, they are integral components of any modern information system and in many cases are essential to the success or failure of the organisation. This journal aims to disseminate practical and theoretical information, which will enable readers to understand, manage, use and maintain business data communication networks more effectively. **International Journal of Business Data Communications and Networking** covers both guided and wireless communications of voice, data, images and video and their impact on the organisations in which they are used. The journal aims to address the key issues for businesses utilising data communications and the increasing importance of networking technologies in business.

**MISSION**
This journal disseminates practical and theoretical information, which will enable readers to understand, manage, use and maintain business data communication networks more effectively.

**TOPICS OF INTEREST:**
- Emerging networking trends
- Business applications of telecommunications
- Policy-based security systems
- Mobility and m-commerce issues
- and many more! Visit www.idea-group.com/ijbdcn

All submissions should be emailed to:

**Jairo Gutierrez**
University of Auckland, New Zealand
j.gutierrez@auckland.ac.nz

**CALL FOR PAPERS**

**International Journal of**

**Business Data Communications and Networking**

*An official publication of the Information Resources Management Association!*

Now when your institution’s library subscribes to any IGP journal, it receives the print version as well as the electronic version for one inclusive price. For more information contact Corrina Chandler, Electronic Content Coordinator, 717/533-8845, ext. 14

**ORDER FORM**

- I wish to join the INFORMATION RESOURCES MANAGEMENT ASSOCIATION (Membership entitles you to IJBDCN and the semi-annual Information Management). $65 Individual*; $195 Institutional*
- I wish only to subscribe to the IJBDCN.* Individual $85.00 Institutional $195.00

* Outside USA, add $15

**IDEA GROUP PUBLISHING**

701 E. Chocolate Ave., Suite 200
Hershey, PA 17033-1240, USA
Tel: 717/533-8845
Fax 717/533-8661

**METHOD OF PAYMENT**

- Check or money order payable to Idea Group Publishing
- Please bill my company
- Please charge my MasterCard VISA Am. Express

Card No. Expiration Date

Signature

**PLEASE RECOMMEND THIS JOURNAL TO YOUR LIBRARY!**
The InfoSci-Online database is the most comprehensive collection of full-text literature published by Idea Group, Inc. in:

- Distance Learning
- Knowledge Management
- Global Information Technology
- Data Mining & Warehousing
- E-Commerce & E-Government
- IT Engineering & Modeling
- Human Side of IT
- Multimedia Networking
- IT Virtual Organizations

**BENEFITS**

- Instant Access
- Full-Text
- Affordable
- Continuously Updated
- Advanced Searching Capabilities

"The Bottom Line: With easy to use access to solid, current and in-demand information, InfoSci-Online, reasonably priced, is recommended for academic libraries." - Excerpted with permission from Library Journal, July 2003 Issue, Page 140

Start exploring at www.infosci-online.com

Recommend to your Library Today!

Complimentary 30-Day Trial Access Available!

A product of:

Information Science Publishing®
Enhancing knowledge through information science

*A company of Idea Group, Inc.
www.idea-group.com
BROADEN YOUR IT COLLECTION
WITH IGP JOURNALS

is an innovative international publishing company, founded in 1987, specializing in information science, technology and management books, journals and teaching cases. As a leading academic/scholarly publisher, IGP is pleased to announce the introduction of 14 new technology-based research journals, in addition to its existing 11 journals published since 1987, which began with its renowned Information Resources Management Journal.

Free Sample Journal Copy
Should you be interested in receiving a free sample copy of any of IGP’s existing or upcoming journals please mark the list below and provide your mailing information in the space provided, attach a business card, or email IGP at journals@idea-group.com.

Upcoming IGP Journals
January 2005
☐ Int. Journal of Data Warehousing & Mining
☐ Int. Journal of Business Data Comm. & Networking
☐ International Journal of Cases on E-Commerce
☐ International Journal of E-Business Research
☐ International Journal of e-Collaboration
☐ Int. Journal of Electronic Government Research
☐ Int. Journal of Info. & Comm. Technology Education

Established IGP Journals
☐ Annals of Cases on Information Technology
☐ Information Management
☐ Information Resources Management Journal
☐ Information Technology Newsletter
☐ Int. Journal of Distance Education Technologies
☐ Int. Journal of IT Standards and Standardization Research
☐ Int. Journal of Web Services Research
☐ Journal of Database Management
☐ Journal of Electronic Commerce in Organizations
☐ Journal of Global Information Management
☐ Journal of Organizational and End User Computing

Visit the IGI website for more information on these journals at www.idea-group.com/journals/

IDEA GROUP PUBLISHING
A company of Idea Group Inc.
701 East Chocolate Avenue, Hershey, PA 17033-1240, USA
Tel: 717-533-8845; 866-342-6657 • 717-533-8661 (fax)

Journals@idea-group.com www.idea-group.com
LIBRARY RECOMMENDATION FORM for IGI Journals

Please forward to your institution’s librarian

Dear ______________________________

I would like to recommend that our library subscribe to the following Idea Group Inc. publications. (I have indicated the justification for the subscription(s) below.)

JOURNALS

- Journal of Cases on Information Technology / ISSN 1548-7717/Organizational rate US$245
- Information Resources Management Journal / ISSN 1040-1628/Organizational rate US$315
- Journal of Organizational and End User Computing / ISSN 1546-2234/Organizational rate US$315
- Journal of Global Information Management / ISSN 1062-7375/Organizational rate US$315
- Journal of Database Management / ISSN 1063-8016/Organizational rate US$315
- Journal of Electronic Commerce in Organizations / ISSN 1539-2937/Organizational rate US$225
- Int'l Journal of Distance Education Technologies / ISSN 1539-3100/Organizational rate US$225
- Int'l Journal of IT Standards & Standardization Research / ISSN 1539-3062/Organizational rate US$205
- Int'l Journal of Web Services Research / ISSN 15398-3062/Organizational rate US$225
- Int'l Journal of Infor. and Communication Technology Education / ISSN: 1550-1876/Organizational rate US$195
- International Journal of Knowledge Management / ISSN: 1548-0666/Organizational rate US$195
- International Journal of Intelligent Information Technologies / ISSN: 1548-3657/Organizational rate US$195
- International Journal of Technology and Human Interaction / ISSN: 1548-3908/Organizational rate US$195
- International Journal of Data Warehousing and Mining / ISSN: 1548-3924/Organizational rate US$195
- International Journal of E-Collaboration / ISSN: 1548-3673/Organizational rate US$195
- International Journal of Enterprise Information Systems / ISSN: 1548-1115/Organizational rate US$195
- Int'l Journal of Business Data Communications & Networking / ISSN: 1548-0631/Organizational rate US$195
- International Journal of E-Business Research / ISSN: 1548-1131/Organizational rate US$195
- International Journal of Electronic Government Research / ISSN: 1548-3886/Organizational rate US$195
- Journal of Cases on Electronic Commerce / ISSN: 1548-0623/Organizational rate US$195
- International Journal of Semantic Web and Information Systems / Organizational rate US$195
- Information Management / ISSN 1080-286x/Organizational rate US$70
- Information Technology Newsletter / ISSN 1057-7939 (US$35.00/year)

THE BENEFIT OF THE JOURNAL(S) FOR OUR LIBRARY IS:

I have indicated the appropriate response (1=highest benefit; 2=moderate benefit; 3=little benefit).

1 2 3 REFERENCE: For research articles in the field of information technology management.
1 2 3 STUDENT READINGS: I plan to recommend articles from the above to my students.
1 2 3 PUBLICATION SOURCE: This journal is a suitable source for my current research results and findings.
1 2 3 PEER EVALUATION: This journal is highly regarded by my peers and is cited often in other’s research work.

(Recommended by) ______________________________ (Department or Division) ______________________________

Note to the librarian: Idea Group Inc. (IGI) journals can be ordered directly or through your designated subscription agency.

Questions or orders may be forwarded to:

Idea Group Inc.
701 E. Chocolate Avenue, Suite 200
Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661
E-mail: cust@idea-group.com; http://www.idea-group.com
Join the international organization serving information providers and users!

INFORMATION RESOURCES MANAGEMENT ASSOCIATION

Name: ___________________________________________
Affiliation: _______________________________________
Address: _________________________________________
City/State/ZIP ____________________________________
Country: _________________________________________
Telephone: _______________________________________
Fax: ____________________________________________
E-mail: __________________________________________

Please check one or more of the following selections.

☐ IRMA Basic Membership+  
( Includes subscription to Information Management and one journal. )

<table>
<thead>
<tr>
<th>Individual -One Year US$65/ Two Years US$120*</th>
<th>Organizational-One Year US$195/Two Years US$380*</th>
</tr>
</thead>
<tbody>
<tr>
<td>US$_____</td>
<td>US$_______</td>
</tr>
</tbody>
</table>

Please check your preferred journal.

☐ Information Resources Management Journal  (4 issues)  
☐ Journal of Organizational & End User Computing  (4 issues)  
☐ Journal of Cases on Information Technology  (4 issues)  
☐ Intl. Journal of IT Standards & Standardization Research  (2 issues)  
☐ Journal of Electronic Commerce in Organizations  (4 issues)  
☐ Intl. Journal of Cases on Electronic Commerce  (4 issues)  
☐ Intl. Journal of Knowledge Management  (4 issues)  
☐ Intl. Journal of Tech & Human Interaction  (4 issues)  
☐ Intl. Journal of E-Collaboration  (4 issues)  
☐ Intl. Journal of Electronic Govt Research  (4 issues)  

☐ Subscription to each additional journal(s)-Please check above

<table>
<thead>
<tr>
<th>Individual -One Year US$50/Two Years US$90*</th>
<th>Organizational-One Year US$80/Two Years US$150*</th>
</tr>
</thead>
<tbody>
<tr>
<td>US$_____</td>
<td>US$_______</td>
</tr>
<tr>
<td>Student -One Year only US$20 ™ *</td>
<td></td>
</tr>
<tr>
<td>US$_______</td>
<td></td>
</tr>
</tbody>
</table>

* Outside U.S., add $15 per year for each journal ordered.

TOTAL US$_______

+ Lifetime Basic Membership can be obtained for $800 with subscription to additional journal(s) available for an additional $750 each.

☐ Check or money order payable to IRMA (and listing a U.S. corresponding bank)

☐ Credit Card     ___ MasterCard  ___ VISA   ___ American Express

Amount that you are authorizing to be charged US $ _____________

Credit Card #:_________________________________________ Expiration Date: ____________________

Signature _____________________________________________

3-digit security code (MC and Visa) on signature panel or 4-digit code (AmEx) on card front: ___________________

Federal ID# 25-1896021

Some of the benefits of IRMA membership include:

• Journals crucial to your professional enrichment.
• Discounts on your IRMA International Conference registration and association publications.
• Interaction with colleagues around the world.
• Participation on professional leadership committees of IRMA.
E-Commerce and M-Commerce Technologies
P. Candace Deans, University of Richmond, USA

E-Commerce and M-Commerce Technologies explores the emerging area of mobile commerce. The chapters in this book look specifically at the development of emerging technologies and their application in Internet commerce. From E-business to mobile database developments, this book offers a compilation of readings that are useful to individuals and organizations in the academic study and research surrounding mobile commerce as well as in the practical application of these technologies.

ISBN 1-59140-566-7 (h/c) • US$79.95
ISBN 1-59140-239-5 (s/c) • US$64.95
280 pages • Copyright © 2005

Innovations of Knowledge Management
Bonnie Montano, Georgetown University, USA

Innovations of Knowledge Management highlights the broad range of topics that fall under the term knowledge management, thus emphasizing the large role knowledge management plays in organizations. As a compilation of some of the most recent work in the field, the included chapters truly present innovations in how organizations can and should manage their knowledge.

ISBN 1-59140-281-6 (h/c) • US$79.95
ISBN 1-59140-229-8 (s/c) • US$64.95
364 pages • Copyright © 2005

It’s Easy to Order!
www.idea-group.com
717/533-8845 x10 • Toll free 1-866-342-6657
Mon-Fri 8:30 am-5:00 pm (est) or fax 24 hours a day 717/533-8661

IRM Press
Hershey • London • Melbourne • Singapore

An excellent addition to your library!