Computer Science & Information Technology 41

David C. Wyld
Natarajan Meghanathan (Eds)

Computer Science & Information Technology

The Seventh International Conference on Wireless & Mobile Network
(WiMo 2015)
Sydney, Australia, May 23 ~ 24 - 2015

AIRCC

Volume Editors

David C. Wyld,
Southeastern Louisiana University, USA
E-mail: David.Wyld@selu.edu

Natarajan Meghanathan,
Jackson State University, USA
E-mail: nmeghanathan@jsums.edu

ISSN: 2231 - 5403
ISBN: 978-1-921987-38-0
DOI : 10.5121/csit.2015.51001 - 10.5121/csit.2015.51008

Typesetting: Camera-ready by author, data conversion by NnN Net Solutions Private Ltd., Chennai, India

Preface

The Seventh International Conference on Wireless & Mobile Network (WiMo-2015) was held in Sydney, Australia, during May 23~24, 2015. The Fourth International Conference on Information Technology Convergence and Services (ITCSE-2015), The Fourth International conference on Advanced Computer Science and Information Technology (ICAIT-2015), and The Third International Conference of Networks and Communications (NC-2015) were collocated with the WiMo-2015. The conferences attracted many local and international delegates, presenting a balanced mixture of intellect from the East and from the West.

The goal of this conference series is to bring together researchers and practitioners from academia and industry to focus on understanding computer science and information technology and to establish new collaborations in these areas. Authors are invited to contribute to the conference by submitting articles that illustrate research results, projects, survey work and industrial experiences describing significant advances in all areas of computer science and information technology.

The WiMo-2015, ITCSE-2015, ICAIT-2015, NC-2015 Committees rigorously invited submissions for many months from researchers, scientists, engineers, students and practitioners related to the relevant themes and tracks of the workshop. This effort guaranteed submissions from an unparalleled number of internationally recognized top-level researchers. All the submissions underwent a strenuous peer review process which comprised expert reviewers. These reviewers were selected from a talented pool of Technical Committee members and external reviewers on the basis of their expertise. The papers were then reviewed based on their contributions, technical content, originality and clarity. The entire process, which includes the submission, review and acceptance processes, was done electronically. All these efforts undertaken by the Organizing and Technical Committees led to an exciting, rich and a high quality technical conference program, which featured high-impact presentations for all attendees to enjoy, appreciate and expand their expertise in the latest developments in computer network and communications research.

In closing, WiMo-2015, ITCSE-2015, ICAIT-2015, NC-2015 brought together researchers, scientists, engineers, students and practitioners to exchange and share their experiences, new ideas and research results in all aspects of the main workshop themes and tracks, and to discuss the practical challenges encountered and the solutions adopted. The book is organized as a collection of papers from the WiMo-2015, ITCSE-2015, ICAIT-2015, NC-2015

We would like to thank the General and Program Chairs, organization staff, the members of the Technical Program Committees and external reviewers for their excellent and tireless work. We sincerely wish that all attendees benefited scientifically from the conference and wish them every success in their research. It is the humble wish of the conference organizers that the professional dialogue among the researchers, scientists, engineers, students and educators continues beyond the event and that the friendships and collaborations forged will linger and prosper for many years to come.

David C. Wyld
Natarajan Meghanathan

Organization

General Chair

Natarajan Meghanathan	Jackson State University, USA
Dhinaharan Nagamalai	Wireilla Net Solutions PTY LTD, Australia

Program Committee Members

Abdelhafid Abdelmalek	Tlemcen University, Algeria
Ahmed Mohamed Khedr	University of Sharjah, UAE
Ali Abid D. Al-Zuky	Mustansiriyah University, Iraq
Ali Chaabani	National School of Engineering, Tunisia
Ali Hussein	Alexandria University, Egypt
Ankit Chaudhary	Truman State University, USA
Barbaros Preveze	Cankaya University Ankara, Turkey
Bela Genge	Petru Maior University of Tg. Mures, Romania
Chun-Yi Tsai	National Taitung University, Taiwan
Dac-Nhuong Le	Haiphong University, Vietnam
David C. Wyld	Southeastern Louisiana University, USA
Emilio UR	University of La Rioja, Spain
Farhan	University Of Indonesia, Indonesia
Farzad Kiani	Istanbul Sabahattin Zaim University, Turkey
Foudil Cherif	Biskra University, Algeria
Girija Chetty	University of Canberra, Australia
Grienggrai Rajchakit	Maejo University, Thailand
Gullanar M Hadi	Salahaddin University, Hawler, Iraq
Habib Rasi	Shiraz University of Technology, Iran
Hamadouche M	Hamadouche Maamar, Algeria
Hamdi Hassen	Sfax University, Tunisia
Hamdi M	National Engineering School of Tunis, Tunisia
Héldon José	Professor of Integrated Faculties of Patos, Brazil
Ian Tan	Multimedia University, Malaysia
Iram Siraj	Aligarh Muslim University, India
Isa Maleki	Islamic Azad University, Iran
Israashaker Alani	Minustry If Science and Technolgy, Iraq
Jacques Epounde Ngalle	Robert Morris University, USA
Jose Enrique Armendariz-Inigo	Public University of Navarre, Spain
Koushik Majumder	West Bengal University of Technology, India
Laudson Souza	Professor of Integrated Faculties of Patos, Brazil
Li, Zheng	University of Bridgeport, USA
Majlinda Fetaji	South East European University, Macedonia
Mehdi Nasri	Islamic Azad University, Iran
Meyyappan T	Alagappa University, India
Moez Hizem	Sup'Com, Tunisia
Mohamed Elboukhari	ESTO, Oujda, Morocco

Technically Sponsored by

Networks & Communications Community (NCC)

Computer Science & Information Technology Community (CSITC)

Digital Signal & Image Processing Community (DSIPC)

Organized By

Academy & Industry Research Collaboration Center (AIRCC)

TABLE OF CONTENTS

The Seventh International Conference on Wireless & Mobile Network (WiMo 2015)

The Fourth International Conference on Information Technology Convergence and Services (ITCSE 2015)

Fourth International conference on Advanced Computer Science and Information Technology (ICAIT 2015)

Third International Conference of Networks and Communications (NC 2015)

PERFORMANCE EVALUATION OF ROUTING PROTOCOLS FOR DELAY TOLERANT NETWORKS

Luming Wan[1], Feiyang Liu[1], Juan Zhang[2], and Haibo Zhang[1]

[1]Department of Computer Science, University of Otago, Dunedin, New Zealand
{lwan, feiyang, haibo}@cs.otago.ac.nz
[2]School of Reliability and System Engineering,
Beihang University, Beijing, China
juanzhang@buaa.edu.cn

ABSTRACT.

Delay Tolerant Network (DTN) is a promising technology which aims to provide efficient communication between devices in a network with no guaranteed continuous connectivity. Most of the existing routing schemes for DTNs achieve message delivery through message replication and forwarding. However, due to the lack of contemporaneous end-to-end communication path, designing routing protocols that can achieve high delivery rate with low communication overhead is a challenging problem. Some routing protocols appear with high similarity, but their performance are significantly different. In this paper, we evaluate several popular routing protocols in DTNs, including Epidemic, Spray and Wait, PRoPHET, and 3R through extensive trace-driven simulations. The objective is to evaluate the performance of different routing schemes using different data traces and investigate the optimal configuration setting for each routing scheme. This paper provides important guidances on the design and selection of routing protocols for given delay tolerant networks.

KEYWORDS

Delay Tolerant Network, Performance Evaluation, Routing

1. INTRODUCTION

The increasing popularity of ubiquitous computing and communication leads to a huge desire on data exchanging between wireless mobile devices, *e.g.* cell phones, laptops, tablets, and other carriable devices, regardless of whether any guaranteed end-to-end connection exists. Delay tolerant networks (DTNs) address the technical issue on communication between devices that lose continuous connectivity due to mobility. Currently, DTNs have been applied to a vast of areas, including vehicular networks[1], wildlife tracking[2], and social network analysis[3], etc. However, DTNs are still appearing with numerous of limitations. For example, due to the lack of synchronous end-to-end connectivity in DTNs, mobile devices have to carry the messages and forward them opportunistically upon encountering the destinations, or forward messages to other relays to help the delivery. This could result in incredible long transmission delay and high error

David C. Wyld et al. (Eds) : WiMo, ITCSE, ICAIT, NC - 2015
pp. 01–12, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.51001

rate. In addition, the constraints of wireless devices, such as storage capacity, communication bandwidth and battery power, can significantly impact the delivery rate. Hence, routing protocols for DTNs should be able to adapt to the network variation and efficient to make use of the hardware resources.

Recent studies exhibit that most of the existing routing protocols designed for DTNs appear with high similarity in concepts, but their performance are significantly disparate. Epidemic[4] performs robust and blind flooding of messages to the network. Spray and Wait[5] is a flooding-controlled version of Epidemic, as it simply limits the amount of messages that can be flooded to the network. In most prediction-based schemes, such as PRoPHET[6], MaxProp[7] and PER[8], the message forwarding decisions are made based on a quality metric called encounter predictability which is accumulated from each time of encounter. However, the above prediction-based schemes ignore some important encounter information, such as the contact time. Some studies demonstrate that human activities appear with high repetition, such as weekly meetings. The performance of routing schemes could be enhanced significantly through the exploitation and utilization of those regular patterns. 3R[9] is a fine-grained history-based routing scheme, by which the contact time of encounters are perfectly recorded. Due to the maintenance of fine-grained encounter history, the encounter predictability in 3R is time-dependent, that is, it can be calculated based on only the past contacts that occurred in the same period as the lifetime of the packet, rather than making a long-term average estimation as in most prediction-based protocols, such as PRoPHET. In addition, 3R is a forwarding-based scheme, each node always forwards the original message to the next good relay rather than transmitting replicas as in all the protocols above. However, single-copy for each message is not ideal for DTNs since message could be easily lost due to unstable connectivity and other uncertainties.

It is difficult to design an absolutely perfect routing protocol for all DTN applications. The performance of a routing protocol can be affected by a large number of factors, such as the popularity and active rate of nodes, different setting of parameters for mobile devices, etc. In this paper, we evaluate the performance of several most popular routing protocols, including Epidemic, Spray and Wait, PRoPHET, and 3R through trace-driven simulations. Also, we investigate the impact of different parameters of Spray and Wait and PRoPHET to achieve their optimal settings. The contribution of this paper is to provide some important guidances on routing protocol design and selection for delay tolerant networks.

The reminder of this paper is organized as follows: Section 2 introduces the protocols we are going to evaluate in detail. Section 3 introduces the data traces and experiment setup. Section 4 shows the performance evaluation in two perspectives. Firstly, we present the impact of different parameter settings for Spray and Wait, and PRoPHET protocols. Then, we evaluate the performance of all the protocols with their optimal parameter settings. The paper is concluded in Section 5.

2. ROUTING PROTOCOLS

This section introduces four protocols of our comparisons in detail, which are Epidemic, PRoPHET, Spray and Wait, and 3R respectively.

2.1. Epidemic

Epidemic[4] is a pure flooding-based routing protocol for DTNs. Each device (source or relaying node) always propagates message replicas to all of the contactable nodes until the message is received by the destination node or the message deadline expires. Due to the nature of unlimited flooding, Epidemic is able to achieve the best message delivery rate when each node has an infinite memory buffer, but its performance deteriorates significantly when each node only has limited resources. This is because numerous of message replicas have to be dumped due to the memory overflow.

2.2. Spray and Wait

Similar to Epidemic, Spray and Wait [5] is also a flooding-based routing protocol, but it controls the message flooding by directly limiting the number of replicas that an original message can produce. Each message can only propagate a constant amount of replicas in the spray phase. After that, the original message and all of its replicas switch to the wait phase, that is, those messages only wait for the destination node encountering their holders rather than further spreading replicas to other relays. Binary Spray and Wait is a derivation of original version, as nodes spread the replicas of a message in a binary tree manner. Suppose a message is allowed to spread L replicas, the source node will generate L replicas in the beginning and spread them among encountered relays. Suppose any node A that has $n(1 \leq n \leq L)$ replicas of message m, and it encounters a node B with no replica of m previously. A will send $\lceil \frac{n}{2} \rceil$replicas to B and keep the rest of $\lfloor \frac{n}{2} \rfloor$replicas to wait for more relaying nodes. When only one replica is left in A or any other relaying node, they stop spreading any replicas and wait for encountering the destination node. Spray and Wait can achieve a trade-off between the delivery rate and transmission overhead by setting a proper limitation of L.

2.3. PRoPHET

PRoPHET[10] is a prediction-based scheme and it is one of the few DTN routing protocols that have an IETF draft. It implements a quality metric called encounter predictability to measure the capability of the encountering nodes whether or not it can transmit the message to the destination. A replica of a message is propagated to the encountering node if it has a higher predictability than its holder. This guarantees that each time a message can always be propagated to a better relay. The encounter predictability is integrated in three perspectives, which are direct probability, transitivity and ageing. Direct probability is the probability of transmitting a message directly between two nodes. It is updated whenever two nodes directly encounter each other as follows:

$$P(a,b) = P_{old}(a,b) + \left(1 - \delta - P_{old}(a,b)\right) \times P_{encounter} \, , \tag{1}$$

where $P_{old}(a,b)$ is the encounter probability of node a and b before the current encounter occurs, $P_{encounter} \in [0,1]$is a scaling factor at which the probability increases on encounters, and $\delta$is a small positive value to set an upper bound on $P(a,b)$.

Transitivity estimates the probability of indirect contact that through multiple-hop relays, as shown below:

$$P(a,c) = P_{old}(a,c) + \left(1 - P_{old}(a,c)\right) \times P(a,b) \times P(b,c) \times \beta, \tag{2}$$

Where *P(a,c)*is the encounter probability between node a and *c* through relay node b, $P_{old}(a,c)$ is the probability from previous time of update, and $\beta \in$ [0,1] is a scaling factor that decides how large the impact of transitivity should have on the encounter predictability.

PRoPHET uses an ageing mechanism to decay the encounter predictability and eliminate the long-time inactive nodes, as shown below:

$$P(a,b) = P_{old}(a,b) \times \gamma^{k}, \tag{3}$$

where $\gamma \in$ [0,1]is the ageing constant, and K is the number of time units elapsed since the last time*P(a,b)* was aged.

As shown in the above Equations, there are three important parameters for PRoPHET protocol: scaling factors for direct delivery $P_{encounter}$ and transitivity β, and the ageing constant γ. They have significant influence on the performance and overheads of Prophet scheme. PRoPHET can be an efficient routing protocol by setting the proper parameters if the encountering patterns in DTNs are predictable, because the messages and replicas are only forwarded to the relays with more benefits on message delivering. At present, there are two versions of PRoPHET protocol: PRoPHET'03[6] and PRoPHET'12[10]. They have different recommendation settings for these parameters, while the principles of message forwarding for both versions are basically identical.

2.4. 3R

3R[9] characterizes the encounter history in a fine-grained form by storing more information for each encounter, including the nodes that the encounter occurs among, the start time and the end time of this encounter. Each node maintains a table for the encounter information of its contacted nodes. The fine-grained history in a node will group the contacts according to different types of the day *{weekday, weekend}*. Each item in the table represents a fixed time interval of the real world (*e.g.* 1 hour), and it records the overall contact frequency between the table holder and the specified node at exact the same period in the past. Suppose a message *m* has a lifetime that spans*k* slots. Let F_i^a be the number of contacts of node *a* that occurred at a time slot *i* in the past, and F_{ab}^{i} be the number of contacts occurred between node *a* and *b* at time slot *i*, the estimated contact probability p_{ab}^{i} at slot *i* is:

$$p_{ab}^{i} = \frac{F_{ab}^{i}}{F_{a}^{i}}. \tag{4}$$

Suppose the lifetime of a message *m* spans *k* time slots, the overall probability that message *m* can be delivered from node *a* to node *b* before it expired is estimated by integrating the probabilities of these *k* slots as follows:

$$P_{ab}(k) = 1 - \prod_{i=0}^{k}(1 - p_{ab}^{i}). \tag{5}$$

Due to the maintenance of fine-grained history, a time-dependent forwarding prediction is enabled by only figuring out the delivery probability within the lifetime of a message, rather than calculating a long-term average estimation. Similar to PRoPHET, 3R also compares the predictability and forwards messages to nodes with higher delivery probability. Whereas, 3R only directly forwards the original message to the next relay, instead of sending a replica as in PRoPHET. Due to the predictability calculation based on fine-grained contact history and single-copy message forwarding, 3R can effectively reduce delivery overheads with no much loss on message delivery rate.

3. DATA TRACES AND EXPERIMENT SETUP

3.1. Data Traces

We evaluate above four representative routing protocols for DTNs with extensive trace-driven simulations. To increase the accuracy and reliability of our evaluation, we use two realistic DTN data traces: INFOCOM[11] and MIT Reality[12], which are obtained from the open-source website, CRAWDAD[13].

- **INFOCOM:** This data trace consists of Bluetooth sightings and it records 4 days of contacts occurred during the INFOCOM'05 conference. In the experiment, 24 internal devices monitored the network and more than 200 external devices were discovered. Each device had a scan granularity of 120 seconds, and each scan lasted for 5 seconds.

- **MIT Reality:** This trace contains the contact information including communication, proximity, and location from faculties and students at MIT over the course of the 2004-2005 academic year. To collect this trace, 89 devices recorded the contacts, and more than 20,000 devices were discovered. Each device scanned to discover neighbors every 5 minutes. In our simulations, we use the devicespansubtrace that records Bluetooth contacts for 1 month.

3.2. Experiment Setup

In our simulations, the routing schemes are evaluated using the same message trace and contact trace. To avoid the negligible delivery rate caused by numerous long-time inactive nodes which actually seldom communicate, we randomly select the source node and the destination node of a message from 20 most active nodes during the lifetime of this message. In the INFOCOM trace, each source generates a message with the probability of 0.15 in every 600 seconds, whereas the probability is 0.6 in the MIT Reality trace. The lifetime of each message varies from 2 hours to 3 days if not specially notified. The size of each generated message is randomized from 2k bytes to 100k bytes.

For the MIT Reality trace, the network is warmed up for one week at the beginning of each simulation run. This avoids the inaccurate prediction on encounter probability for prediction-based protocols, *i.e.*PRoPHET and 3R, when the network is just booted up. Also, we reserve 3 days at the end of simulations to avoid any message left in the network after the simulation is completed. During these warm-up and sinking periods, messages are not allowed to be generated. This process is not available for the INFOCOM data trace, since it only contains 4 days of contact traces.

4. PERFORMANCE EVALUATION

In this section, we firstly exhibit the performance of Spray and Wait and PRoPHET with different parameter settings using the INFOCOM data trace, since they have some specific parameters. Then we investigate the performance of all four protocols, *i.e.* Epidemic, Spray and Wait, PRoPHET, and 3R, through trace-driven simulations. The performances of these four protocols are investigated in two perspectives: different memory buffer sizes and packet lifetimes. The performance is evaluated using the following three metrics:

- *average message delivery rate:* the proportion between the number of successfully delivered messages against the number of original messages.
- *message overhead (MO):* the ratio of the number of replicas against the number of original messages.
- *communication overhead (CO):* the ratio of the difference between the number of delivered messages and hop-by-hop transmissions to the number of delivered messages.

The parameter setting for Spray and Wait is slightly different in each data trace in accordance with the characteristic of the data trace. The Binary Spray and Wait protocol is implemented in the entire experiments as it performs slightly better than its original version. Due to the huge difference on popularity of nodes in these two traces, the maximum number of replicas for Binary Spray and Wait with INFOCOM trace is set to 5, and 70 with MIT Reality trace. The parameters in PRoPHET are configured with the values provided in PRoPHET'12[10] by default. All the nodes are set with infinite memory buffer if not specified. All the protocols dump messages using the First In First Out policy when the message buffer is full, that is, the first message in the buffer queue (oldest) is dumped to prevent buffer overflow.

4.1. Impact of Parameters for Spray-and-Wait and PRoPHET

Spray-and-Wait: Fig. 1 shows the performance of the original and binary version of Spray and Wait routing protocol with different maximum number of replicas. In Fig.1(a), it can be seen that the original Spray and Wait achieves higher delivery rate than the binary version when the maximum number of replicas is less than 10. With the further increase on the number of replicas, the binary version performs better than the original version. That is because the original scheme can only spread the replicas around the neighbours, while the binary scheme can spread the replicas more widely in a binary-tree manner if enough number of replicas are allowed. We can seethe delivery rate of both protocols remain steady when the maximum replicas exceeds 40.

The highest average delivery rate achieved by the original and the binary versions are 0.9314 and 0.9382, respectively. That means the number of replicas are enough for two Spray and Wait protocols with no further benefits for more replicas. Meanwhile, MO and CO for both schemes appear with the similar increasing trend as shown in Fig. 1(b) and (c). The MO and CO of the original Spray and Wait increase apparently with the maximum replicas less than 40, and they stay constant afterwards also due to the limitation on the number of neighbours, whereas both two kinds of overheads of Binary Spray and Wait grow almost linearly since the replicas can be spread to more relay nodes on the network. From these figures, it can be seen that with a large number of replicas, the multi-hop routing of Binary Spray and Wait is able to perform better than the only two-hop routing in original version, but it suffers almost twice of overheads than the original version.

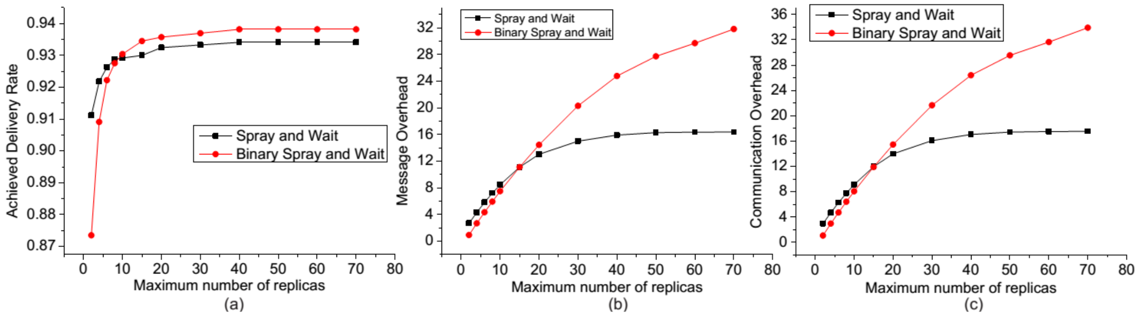


Fig. 1. The performance of Spray and Wait with different number of message replicas, (a) delivery rate, (b) message overhead, (c) communication overhead.

The highest average delivery rate achieved by the original and the binary versions are 0.9314 and 0.9382, respectively. That means the number of replicas are enough for two Spray and Wait protocols with no further benefits for more replicas. Meanwhile, MO and CO for both schemes appear with the similar increasing trend as shown in Fig. 1(b) and (c). The MO and CO of the original Spray and Wait increase apparently with the maximum replicas less than 40, and they stay constant afterwards also due to the limitation on the number of neighbours, whereas both two kinds of overheads of Binary Spray and Wait grow almost linearly since the replicas can be spread to more relay nodes on the network. From these figures, it can be seen that with a large number of replicas, the multi-hop routing of Binary Spray and Wait is able to perform better than the only two-hop routing in original version, but it suffers almost twice of overheads than the original version.

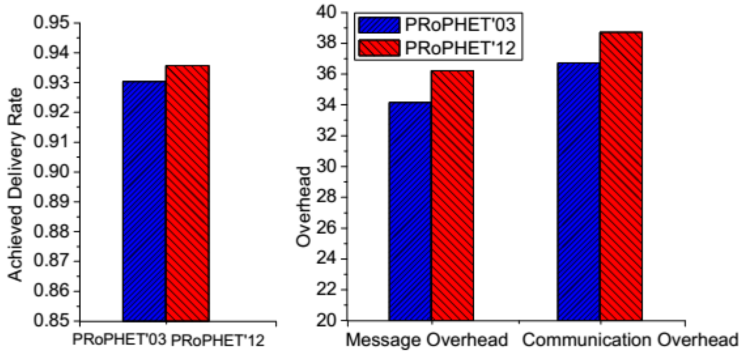
PRoPHET: The performance of PRoPHET routing protocol is critically impacted by the parameters shown in Table 1. PRoPHET provides totally different recommended parameters in their 2012 version [10] compared to the 2003 version [6].

Table 1.PROPHET recommended parameters

Parameters	$P_{encounter}$	β	γ
PRoPHET 2003	0.75	0.25	0.98
PRoPHET 2010	0.5	0.9	0.999

Fig. 2 shows the performance of PRoPHET'03 and PROPHET'12. In these experiments, each node has infinite memory buffer to store messages and replicas. The principles of message forwarding for these two versions are basically identical. It can be seen that in Fig. 2(a), the delivery rate is increased by 0.5% with the parameters of PRoPHET'12 in comparison with PRoPHET'03. A larger ageing parameterγmakes PROPHET'12 better to tolerate the variation of contact patterns and achieve higher delivery rate. However, both MO and CO are raised by 6.1% and 5.4% respectively as shown in Fig. 2(b). This is because a largerβin PRoPHET'12 leads to a higher impact on the transitive connectivity, and a lower $P_{encounter}$ reduces the impact of direct delivery. Hence, more replicas need to be generated and forwarded by the relay nodes.

4.2. Performance Evaluation with Different Memory Sizes



(a) (b)

Fig.2.PRoPHET'03 versus PRoPHET'12, (a) delivery rate, (b) overheads.

In each mobile device, the memory buffer used for message delivery always has a limited size. The available memory has huge influence on the delivery rate and overheads. We evaluate the performance of four routing protocols with different memory sizes using INFOCOM and MIT Reality traces.

(1) INFOCOM Trace: In Fig. 3(a), we evaluate the influence of memory size on the delivery rate for different routings. It can be seen that the delivery rate of all four routing protocols have a sharp increasing trend with a small memory buffer (<10MB), and then they tend to remain stable even if the memory buffer size keeps growing. Binary Spray and Wait has more outstanding performance than other three routings when each node has a memory buffer of 80M bytes or less. It reaches its maximum delivery rate of 0.923 with the memory buffer size increasing to 60M bytes, and the delivery rate tends to be stable afterwards. That is because it is a controlled flooding scheme (5 replicas) and has less message dropped due to memory overflow than other routings. When the memory size exceeds 60MB, there will be no message dumps for this routing and the delivery rate will be stable. Similarly, 3R also has better performance with a smaller memory buffer size compared with Epidemic and PRoPHET. It has less requirement on memory size because of single-message and its limitation on message forwarding. However, it reaches the maximum delivery rate of 0.792 at the memory size of 40M bytes, which is much smaller compared with the other three protocols. That is because 3R only forwards the original message. Although the delivery rates of Epidemic and PRoPHET raise slowly, their maximum delivery rates are higher than 3R and Spray and Wait (0.941 and 0.930 respectively) since they are memory-hungry schemes. Epidemic protocol needs large amounts of memory for unlimited flooding and can achieve the highest delivery rate with enough memory. PRoPHET also needs a large memory size for transmitting the amounts of messages and replicas.

Fig. 3(b) and (c) evaluate the overheads with different memory sizes. We can see that Epidemic always has the highest MO and CO due to the nature of unlimited flooding. Since it needs to generate more replicas when encountering relay nodes with high predictability, PRoPHET also has higher overheads, which is almost half of the overheads as Epidemic. Because the maximum number of replicas in Binary Spray and Wait is set to 5, the maximum MO and CO of Binary Spray and Wait scheme tend to 5 even with a much larger memory size. Since 3R is a single-message forwarding-only scheme, there is no message overhead. The CO of 3R is also lowest because the message is only forwarded to a node with higher predictability according to fine-grained encounter history. The CO of Binary Spray and Wait and 3R appear with a decreasing trend as the memory size increases, since they have limitations on message forwarding and a larger memory size increases the number of delivered messages by reducing message dumps.

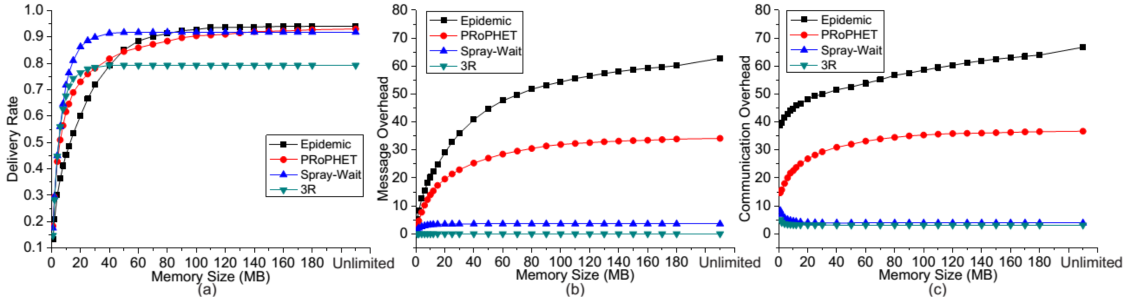


Fig.3.Performance evaluation with different memory sizes using INFOCOM trace,(a) delivery rate, (b) message overhead, (c) communication overhead.

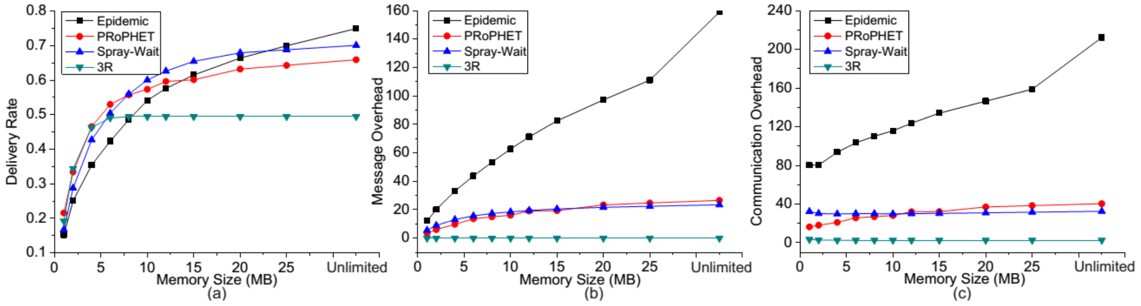


Fig. 4.Performance evaluation with different memory sizes using MIT Reality trace, (a) delivery rate, (b) message overhead, (c) communication overhead.

(2)MIT Reality Trace: As shown in Fig. 4(a), the delivery rate of each routing protocol has similar trend as the memory size increases. However, PRoPHET and 3R have better performance than Epidemic and Spray and Wait when the memory size is small (<5MB). Since the MIT Reality trace lasts for 1 month and these two routings can make use of the contact history to get more accurate prediction for message forwarding, a smaller memory is relatively enough due to less replicas and message forwarding. As the memory size increases, Epidemic and Spray and Wait achieve higher delivery rate because a large number of flooding replicas can be tolerated in the memory with few of message dumps.

Similarly, the message overhead (MO) in Fig. 4(b) and communication overhead in Fig. 4(c) vary in the same way as the memory size increases. Since there are more nodes in the MIT Reality data trace than in INFOCOM trace, the overheads of Epidemic are significantly increased. That means Epidemic is not suitable for DTNs with a large number of devices. We can see that the overheads of Spray and Wait increase with the increase of maximum allowed replicas (from 5 in INFOCOM trace to 70 in MIT Reality trace). PRoPHET and 3R protocols have lower overhead for both data traces, which is determined by the popularity of nodes and the activeness of each node.

From this set of simulations, it can be seen that flooding-based routing protocols can achieve better delivery rate, but require much larger message memory. Whereas, the prediction-based routing protocols are more memory efficient, and can achieve desirable delivery rate with very low overheads as they have an accurate prediction on the contact patterns.

4.3. Performance Evaluation with Different Packet Lifetimes

Each message may experience an unpredictable delivery latency in DTNs due to the lack of guaranteed continuous end-to-end connectivity. Therefore, in order to achieve more effective communication for DTNs, each message needs a suitable lifetime either by users setting or default configuration of specific devices.In this set of simulations, we evaluate the influence of different packet lifetimes for the routing algorithms using INFOCOM and MIT Reality data traces respectively. We assume each node has an infinite buffer size.

(1) INFOCOM Trace: Fig. 5 shows the delivery rate and overheads of four routing protocols with different setting of packet lifetimes using the INFOCOM trace. In Fig. 5(a), it can be seen that for each routing protocol the delivery rate appears with a dramatic increase when the packet lifetime increases from 6 to 30 hours. For example, the delivery rate increases from 0.454 to 0.832 for 3R routing. The reason is that the probability of directly encountering the destination or the message delivered to the destination by multi-hop forwarding grows when the packet lifetime increases. This increasing trend slows down when the packet lifetime exceeds 30 hours. That is because the data trace only lasts 4 days, and the contact pattern is short-time dependent. The delivery rate will not increase even with a large packet lifetime since the destination node may only be active in a short time period. The overheads shown in Fig. 5(b) and (c) also demonstrate that, for Epidemic and PRoPHET, the overheads increase as the packet lifetime increases, since more replicas can be generated and forwarded to the destination. Whereas, the lifetime is enough long for message delivery when it exceeds 30 hours in this trace. The overheads of Spray and Wait and 3R keep stable with different packet lifetime, because the number of replicas is limited in Spray and Wait and 3R is a single-message forwarding based scheme.

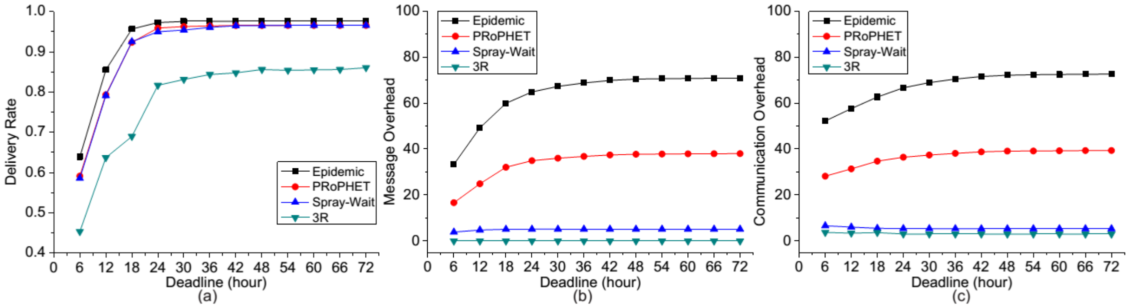


Fig.5.Performance evaluation with different packet lifetimes using INFOCOM trace, (a) delivery rate, (b) message overhead, (c) communication overhead.

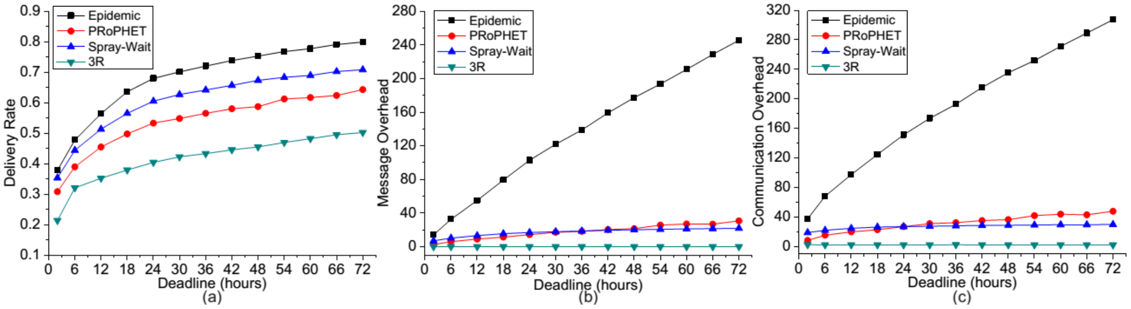


Fig.6.Performance evaluation with different packet lifetimes using MIT Reality trace, (a) delivery rate, (b) message overhead, (c) communication overhead.

(2) MIT Reality Trace: The simulation results with MIT Reality trace are similar as shown in Fig. 6. It can be seen in Fig.6(a) that the delivery rate for all the routing protocols keep growing as the packet lifetime increases. It is because MIT Reality trace lasts for 1 month and it is long-time dependent, thus a longer packet lifetime always provides a larger probability of successful delivery. In this simulation, as shown in Fig.6(b) and (c) the overheads of Epidemic increase dramatically both due to longer packet lifetime and larger device population. Whereas, the overheads variations for other routing protocols are similar to the results using INFOCOM data trace.

This set of simulations demonstrate that a longer packet lifetime can provide a higher delivery rate for DTNs, since there is more chance to encounter the destination node either by direct contact or multi-hop message forwarding. The overheads of flooding-based routings increase significantly, while the prediction-based routings are more efficient, especially when the DTNs have long-time dependent contact patterns.

5. CONCLUSION

Due to the property of Delay Tolerant Networks for no guaranteed continuous connectivity, choosing a proper routing protocol and finding its optimal configuration are challenging problems. In this paper, we evaluate some existing representative routing protocols for DTNs through extensive trace-based simulations. Flooding-based routing protocols, such as Epidemic and Spray and Wait, can achieve better delivery rate by sacrificing the memory, while prediction-based routings, like PRoPHET and 3R, are more efficient when considering the delivery overheads. This paper provides important guidances for routing protocols design. Our future work is to design an efficient routing protocol for DTNs with high delivery rate and low delivery overheads.

REFERENCES

[1] T. Cao, X. Zhang, L. Kong, X. Liu, W. Shu, and M. Wu, "Traffic awarerouting in urban vehicular networks," in WCNC, pp. 2004–2009, 2013.

[2] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein,"Energy-efficient computing for wildlife tracking: Design tradeoffs andearly experiences with zebranet," SIGARCH Comput. Archit. News,vol. 30, no. 5, pp. 96–107, 2002.

[3] Y. Zhang and J. Zhao, "Social network analysis on data diffusion indelay tolerant networks," in MobiHoc, pp. 345–346, 2009.

[4] A. Vahdat and D. Becker, "Epidemic routing for partially-connected adhoc networks," tech.rep., 2000.

[5] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray andwait: An efficient routing scheme for intermittently connected mobilenetworks," in WDTN, pp. 252–259, 2005.

[6] A. Lindgren, A. Doria, and O. Schel´en, "Probabilistic routing in intermittentlyconnected networks," SIGMOBILE Mob.Comput.Commun.Rev., vol. 7, pp. 19–20, July 2003.

[7] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, "Maxprop: Routingfor vehicle-based disruption-tolerant networks," in INFOCOM, pp. 1–11, April 2006.

[8] Q. Yuan, I. Cardei, and J. Wu, "An efficient prediction-based routingin disruption-tolerant networks," IEEE Trans. Parallel Distrib.Syst,vol. 23, no. 1, pp. 19–31, 2012.

[9] L. Vu, Q. Do, and K. Nahrstedt, "3r: Fine-grained encounter-basedrouting in delay tolerant networks," in WoWMoM, pp. 1–6, 2011.

[10] A. Lindgren, A. Doria, E. Davies, and S. Grasic, "Probabilistic routingprotocol for intermittently connected networks," Internet-Draft, 2012.

[11] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau,"CRAWDAD data set cambridge/haggle (v. 2006-01-31)."http://crawdad.org/cambridge/haggle/, 2006.

[12] N. Eagle and A. (Sandy) Pentland, "Reality mining: sensing complexsocial systems," Pers. Ubiq.Comput., vol. 10, no. 4, pp. 255–268, 2006.

[13] D. Kotz, T. Henderson, and I. Abyzov, "CRAWDAD data set dartmouth/campus (v. 2004-12-18)." http://crawdad.org/dartmouth/campus,2004.

AUTHORS

Luming Wan, is a Master candidate from Department of Computer Science, University of Otago, New Zealand. He obtained B.S. degree from University of Otago, New Zealand in 2012. His research interests include Delay Tolerant Network (DTN), Social Network, etc.

Feiyang Liu,is a PhD candidate from Department of Computer Science, University of Otago, New Zealand. He obtained B.S. and M.S. degrees of telecommunication engineering from Xidian University, China in 2009 and 2012, respectively. His research interests include Network on Chip (NoC), Optical Network on Chip (ONoC), Delay Tolerant Networks (DTN), etc.

Juan Zhang, received B.S degree in 2011from Jiangsu University,China, and M.S. degree of control science and engineering in 2014 from Beihang University, China. Her research interests include distributed algorithms and protocols, wireless sensor networks, cyber-physical systems, and data communication, etc.

Haibo Zhang, received the MSc degree in Computer Science from Shandong Normal University, China in 2005, and the PhD degree in Computer Science from the University of Adelaide, Australia in 2009. From 2009 to 2010, he was a postdoctoral research associate at Automatic Control Laboratory, KTH, Sweden. Currently he is a lecturer at Computer Science department of University of Otago, New Zealand. His research interests include real-time industrial wireless communications, wireless sensor/ad hoc networks, delay-tolerant networks, green computing, distributed algorithms and protocol design.

MODELING SOCIAL GAUSS-MARKOV MOBILITY FOR OPPORTUNISTIC NETWORK

GuoDong Kang[1] Jinzhi Ning[1] and GuoLiang Kang [2]

[1]DFH Satellite Co., Ltd., 100094, Beijing, China
kongton584@163.com
[2]University of Technology, Sydney 15 Broadway, Ultimo NSW 2007
kgl.prml@gmail.com

ABSTRACT

Mobility is attracting more and more interests due to its importance for data forwarding mechanisms in many networks such as mobile opportunistic network. In everyday life mobile nodes are often carried by human. Thus, mobile nodes' mobility pattern is inevitable affected by human social character. This paper presents a novel mobility model (HNGM) which combines social character and Gauss-Markov process together. The performance analysis on this mobility model is given and one famous and widely used mobility model (RWP) is chosen to make comparison..

KEYWORDS

Mobility, Social, Gauss-markov, Opportunistic

1. INTRODUCTION

With rapid growing popularity of small portable devices (e.g., smart phones, PDAs, laptops) in personal communication domain, a prospective networking scenario occurs where a large number of mobile nodes share or forward information by means of opportunistic encounter[16]. Since these mobile nodes are carried by human, modeling mobility based on realistic human mobile pattern is becoming an essential and important thing. Normally people live in various social circles such as their family circle, their friends circle and so on. This kind of social property influences human movement pattern and further affects nodes' pattern as well. Sometimes in a social circle some people are more popular than others and have more chance to encounter others. The nodes carried by this kind of person can be regarded as one suitable choice for relaying data with other social circles. In other words, these persons could be selected as heads of the corresponding circles and the nodes carried by them could be regarded as one head node. To construct this kind of social opportunistic mobility, this paper adopts random graph for setting up social relations and Gauss-Markov theory for solving individual mobile pattern.

David C. Wyld et al. (Eds) : WiMo, ITCSE, ICAIT, NC - 2015
pp. 13–25, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.51002

The rest of this paper is organized as follows: Section II describes the allocation method of mobile nodes. Section III proposes the mathematical model of the mobility. Section IV gives analysis of the proposed mobility model. Conclusion is given in Section V.

2. GROUP ALLOCATION METHOD

People's social circle character has a significant impact on nodes mobility pattern. Nodes in different social circles can be allocated into different groups. Node often has more relations with other nodes inside the same group or circle than with those nodes belonging to different groups or circles.

2.1. Node Relation Setup

Normally, there is some kind of social or biological relations among those nodes since they are carried by people. To set up this kind of mutual relation, we adopt the classical method of representing social or biological network, weighted graphs. The strength of mutual relation between any node pair is represented using a value in the range [0, 1]. As a consequence, the network internal relation can be described as a relation matrix with a dimension of N*N where N is the total number of nodes in the network. At present, there are several models that describe the key properties of real-life network, such as random graph, small world, etc. Some research work show that the properties of these random graphs, such as path lengths, clustering coefficients, can't be regarded as accurate models of realistic networks [5, 6, 7]. Here, we choose the geometric random graph to be the network model. In this kind of model, the geometry relations of nodes have strong association with the social relation of nodes. That means when any two nodes are in the radio range of each other, the social relation exists. On the contrary, since they even can't communicate with each other, we think that there is no any social interaction between them. So when the Euclidean distance between any two nodes is smaller than radio range R, the corresponding element of the social relation matrix M is set to be 1 or else set to be 0 as follows.

$$m_{i,j} = \begin{cases} 1 \ \ \textit{if } i \neq j \textit{ and } \left\| P_i - P_j \right\| \leq R, \\ 0 \ \ \textit{otherwise} \end{cases}$$

It shall be emphasized that the relations value of one node and itself is regarded to be zero in the matrix. In [6] it is shown that in two or three dimensional area using Euclidean norm can supply surprisingly accurate reproductions of many features of real biological networks. Fig.1 gives one example of 100 nodes.

2.2. Nodes Group Allocation

Once the relation matrix M is obtained, groups can be detected. Group structure is one of the common characters in many real networks. However, finding group structures within an arbitrary network is an acknowledged difficult task. A lot of work has been done on that. Currently, there are several methods that can achieve that goal, such as Minimum-cut method, Hierarchical clustering, Girvan-Newman algorithm, etc.

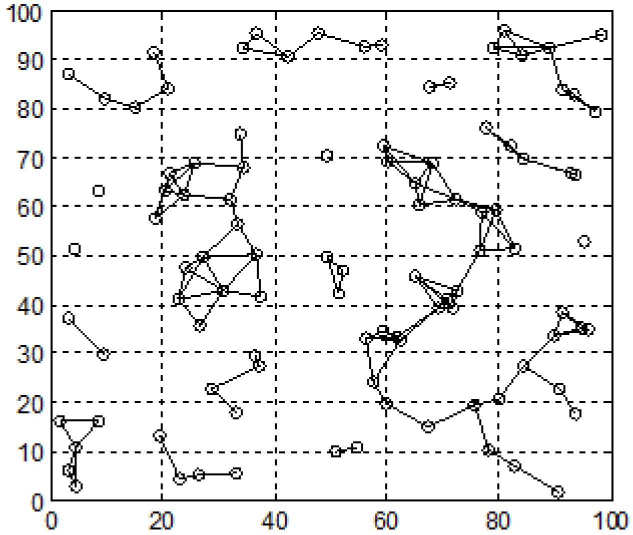


Fig.1. An example of 100 Nodes Relations. (Small circles represent nodes carried by people which are uniformly distributed in one given 100m*100m square; the short straight lines represent the existing relation between node pair.)

Minimum-cut method is one of the oldest algorithms for dividing networks into parts. This method uses in load balancing for parallel computing in order to minimize communication between processor nodes. However, this method always finds communities regardless of whether they are implicit in the structure, and it can only find a fixed number of them. So it is less than ideal for finding community structure in general networks [4].

Hierarchical clustering is another method for finding community structures in network. This method detects the community by defining a similarity measure quantifying some (usually topological) type of similarity between node pairs.

The Girvan–Newman algorithm is one commonly used algorithm for finding communities [12]. It identifies edges in one network that lie between communities and then removes them, leaving behind just the communities themselves. But this method runs slowly which makes it impractical for networks of more than a few thousand nodes [11].

Modularity maximization is one of the most widely used methods for community detection [11]. Modularity is a benefit function that measures the quality of a particular division of a network into communities. This method detects the community structure of high modularity value by exhaustively searching over all possible divisions [8].

In this paper we adopt modularity maximization as the social group detection method. Modularity maximization is one of the most widely used methods for group detection [11].

Modularity is a benefit function that measures the quality of a particular division of a network into groups. This method detects the group structure of high modularity value by exhaustively searching over all possible divisions [8].

In real networks, the modularity value is usually in the range [0.3, 0.7]; 1 means a very strong group structure and 0 means no better than random.

3. MOBILITY MODEL SETUP

3.1. Mobility Scenario Description

One kind of usual mobility scenario appearing in daily life is one people in a group leading the mobility. This person is selected because of his popularity in the group or his familiarity with the environment. He is called as head of the group in this paper. The head moves freely and leads other people to move together. Other people follow head people's direction and adjust their speeds to keep up with him. The head's direction and speed determine the general direction and speed of whole group. Fig.2 gives an illustration of this mobility scenario.

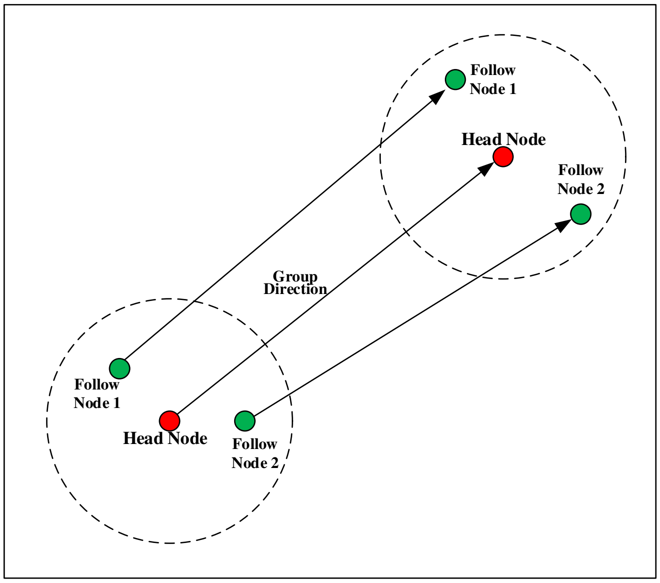


Fig.2. Illustration of mobility scenario. The red solid circle represents head node or head people. The other two blue solid circles represent follow nodes or follow people. The dash circles represent social circles or groups which head node and follow nodes belong to.

3.2. Mobility Mathematical Model

Assume there are total N nodes in one group, the n^{th} node is denoted as node $n\left(1 \le n \le N\right)$

3.2.1 Head Node Determination Scheme

Assume at instant t there are number M nodes which react with node i. The following expression describes the average relation value of node i with other nodes :

$$\overline{m}_i(t) = \frac{1}{M}\sum_{j=1}^{N} m_{i,j}(t) \qquad (i \ne j)$$

where

N is the total number of the network;

M is the number of nodes in the network characterized by $m_{i,j} > 0$

$\overline{m}_i(t)$ is the average relation value of node i at instant t;

$m_{i,j}(t)$ is the element of the relation matrix at instant t;

In this paper, the node which has the greatest average relation values will be assigned to be the Head node of the group and the remaining nodes in the group will be regarded as Follow nodes which will follow the head node. At the beginning, the head node of the group is denoted as node h, it has

$$\overline{m}_h(t \mid t=0) \geq \overline{m}_i(t \mid t=0) \qquad (1 \leq i \leq N, i \neq h)$$

where $\overline{m}_h(t)$ is the average relation value of node h at time t;

Initially, the head node is chosen as the node which is most popular in the group, namely has the strongest relations with other member nodes of the group. However, as the time past, the mutual relation between the initial assigned head node and other nodes in the group may vary. Maybe at time t_e the initial head node don't have the strongest relation with others any more. Consequently, one election shall take place to determine the next new head node who has the strongest relation with other nodes at time t_e. The time t_e is called head node determination period in this paper. The average relation value of the new elected head node, h_E, has

$$\overline{m}_{h_E}(t \big| t = t_e \bullet E) \geq \overline{m}_i(t \big| t = t_e \bullet E) \qquad (1 \leq i \leq N, i \neq h)$$

where

$E \geq 0$ is the head node election number in the group;

t_e is the head node election period.

3.2.2. Gauss-Markov process setup

A Gauss-Markov process is a simple stochastic process with many social applications. In terms of continuous time, a stationary Gauss-Markov process can be described by the following autocorrelation function [16,18]:

$$R_f(\tau) = E\left[f(t)f(t+\tau)\right] = \sigma^2 e^{-\beta|\tau|} \qquad (1)$$

where

σ^2 is the variance;

$\beta \geq 0$ determine the degree of memory in the process.

Assume to discretize time into serial time slots Δt (normalized to 1 throughout this paper),

$$f(t) = f(s \cdot \Delta t)$$

Define $\alpha = e^{-\beta\Delta t}$, then the discrete representation of (1) is

$$f(s \cdot \Delta t) = \alpha f\left[(s-1) \cdot \Delta t\right] + (1-\alpha)\mu + \sqrt{1-\alpha^2}\, d_{Gauss}$$

where

$0 \le \alpha \le 1$;

μ is the asymptotic mean of f when s approach infinity.

d_{Gauss} is an independent, uncorrelated, and stationary Gaussian process, with mean μ_{Gauss} equals to zero and standard deviation σ_{Gauss} equals to σ when s approaches infinity.

3.2.3. Gauss-Markov mobility discussion

In this paper, nodes' speed and direction are both regarded as Gauss-Markov processes which are correlated with time. Nodes' speed and direction discrete representations of (1) can be described as follows:

$$V(s \cdot \Delta t) = \alpha V\left[(s-1) \cdot \Delta t\right] + (1-\alpha)\mu_v + \sqrt{1-\alpha^2}\, d^{v}_{Gauss}$$
$$\Phi(s \cdot \Delta t) = \alpha \Phi\left[(s-1) \cdot \Delta t\right] + (1-\alpha)\mu_\phi + \sqrt{1-\alpha^2}\, d^{\phi}_{Gauss}$$

Gauss-Markov allows previous speed or direction to influence current speed or direction. Moreover, Gauss-Markov is a memory-tunable process accounting for its suitableness for the description of relations of follow nodes and head node about speed and direction choices. The parameter, α, is tunable which determines the amount of memory and variability in movement. The following discusses three cases of parameter α.

Case 1: $\alpha = 0$

When $\alpha = 0$, Gauss-Markov becomes memoryless; The speed and direction of nodes completely depend on their average μ_v, μ_ϕ and the Gauss variables.

$$V(s \cdot \Delta t) = \mu_v + d^{v}_{Gauss}$$
$$\Phi(s \cdot \Delta t) = \mu_\phi + d^{\phi}_{Gauss}$$

This paper thinks this case is quite suitable for setting up the relations of speed and direction between follow nodes and head node by setting μ_{fnv}, $\mu_{fn\phi}$ of follow nodes to identify μ_{hv}, $\mu_{h\phi}$ of head node since follow nodes have one common character that is they always try to keep up with head node movement.

Case 2: $\alpha = 1$

When $\alpha = 1$, Gauss-Markov loses all the variables becoming predictable pattern. The current speed and direction is totally identical to the previous values. The movement becomes a straight line.

$$V(s \cdot \Delta t) = V\left[(s-1) \cdot \Delta t\right]$$
$$\Phi(s \cdot \Delta t) = \Phi\left[(s-1) \cdot \Delta t\right]$$

This paper thinks this case is suitable for head node mobility since his destination is explicit.
Case 3: $0 < \alpha < 1$

This is intermediate randomness case between the two above cases. This case won't be taken into account too much here.

3.2.4 Head node mobility setup

Normally, the head node in a group shall know its destination explicitly. Once he determines the destination, he will move straightly toward the destination maintaining his speed and direction until he reaches the destination. During head node mobility, he may have several destinations thus has several corresponding mobility sections. Each mobility section is one $\alpha = 1$ case Gauss-Markov process. The speed and direction of different mobility sections are different. Head node mobility is set up as follows.

Assume at instant t_i one destination of head node h is chosen and at instance t_j the head node reaches this destination. The destination of head node can be denoted as

$$P_{dh}(t \,|\, t_i \le t \le t_j) = \left[X_{dh}(t_i) \quad Y_{dh}(t_i)\right]$$

Normalize Δt to be 1, at instant t ($t_i < t \le t_j, t = s \cdot \Delta t$), the position of node h is a function of coordinate $X_h(t)$ and $Y_h(t)$ which can be expressed as follows:

$$P_h(s \cdot \Delta t) = \begin{bmatrix} X_h(s \cdot \Delta t) \\ Y_h(s \cdot \Delta t) \end{bmatrix} = \begin{bmatrix} X_h\left[(s-1) \cdot \Delta t\right] + V_h(s \cdot \Delta t) \cdot \cos\Phi_h(s \cdot \Delta t) \\ Y_h\left[(s-1) \cdot \Delta t\right] + V_h(s \cdot \Delta t) \cdot \sin\Phi_h(s \cdot \Delta t) \end{bmatrix}$$

where

s is the number of Δt at current instant;

$s-1$ is the number of Δt at previous instant;

Φ_h is the direction function of node h which can be defined as

$$\Phi_h(t) = arctg\left(\frac{|Y_{dh}(t_i) - Y_h(t_i)|}{|X_{dh}(t_i) - X_h(t_i)|}\right)$$

V_h is the speed function obeying normal distribution which is determined with destination synchronously.

The research in [13] has shown that the walking speed of a human obeys normal distribution. The measurement in [14] shows that the mean speed of a walking human is a range which is from 1.16 m/s to 1.58 m/s representing walking normally or walking fast. In [15], the Manual of Uniform Traffic Control Devices (MUTCD) shows that human walk with a normal speed of 1.2 m/s (4 ft/sec). [10] indicates a statistics that walking speed for younger human (ages 14 to 64) was 1.25 m/sec (4.09 ft/sec); for older human (ages 65 and over) it was 0.97 m/sec (3.19 ft/sec). For designing purposes values of 1.22 m/sec (4 ft/sec) for younger human and 0.91 m/sec (3 ft/sec) for older human are appropriate [10]. In this paper, the mean value of walking speed of the head node, μ_{hv}, equals to 1.22 m/s [10] and its standard deviation σ_{hv} is 0.26 m/s[13]. So the head node's speed at time t is

$$V_h(t) = V_h(t_i) \sim N(\mu_{hv}, \sigma_{hv})$$

The probability density function of $V_h(t)$ can be expressed as:

$$f(v) = \frac{\exp\left(-\frac{(v-\mu_{hv})^2}{2\sigma_{hv}^2}\right)}{\sigma_{hv}\sqrt{2\pi}}$$

Once head node reaches the destination, he will go towards his next destination which is uniformly chosen in given movement area.

3.2.5 Follow nodes mobility setup

Follow nodes' speed and direction discrete representation of (1) can be described as follows according to case 1:

$$V_{fn}(s \cdot \Delta t) = \mu_{fnv} + d_{Gauss}^{v}$$

$$\Phi_{fn}(s \cdot \Delta t) = \mu_{fn\phi} + d_{Gauss}^{\phi}$$

where

μ_{fnv} equals to μ_{hv};

d_{Gauss}^{v} is an independent, uncorrelated, and stationary Gaussian process, with mean μ_{Gauss}^{v} equals to 0 and standard deviation σ_{Gauss}^{v} equals to σ_{hv}.

Similarly, this paper chooses $\mu_{fn\phi} = \Phi_h$ and $\sigma_{Gauss}^{\phi} = \pi/12$.

Consequently, the position of follow node fn is a function of coordinate $X_{fn}(s \cdot \Delta t)$ and $Y_{fn}(s \cdot \Delta t)$ which can be expressed as follows:

$$P_{fn}(s \cdot \Delta t) = \begin{bmatrix} X_{fn}(s \cdot \Delta t) \\ Y_{fn}(s \cdot \Delta t) \end{bmatrix} = \begin{bmatrix} X_{fn}\left[(s-1) \cdot \Delta t\right] + V_{fn}(s \cdot \Delta t) \cdot \cos\Phi_{fn}(s \cdot \Delta t) \\ Y_{fn}\left[(s-1) \cdot \Delta t\right] + V_{fn}(s \cdot \Delta t) \cdot \sin\Phi_{fn}(s \cdot \Delta t) \end{bmatrix}$$

The mean of speed and direction of follow nodes equaling to the values of head node interprets that their mobility shall follow group's movement. The difference of standard deviation of normal distribution between follow-node and head-node reflects these follow-nodes have their freedom degree in mobility to some extent.

4. MODEL CHARACTER ANALYSIS

4.1 Classical Individual Mobility Model-RWP

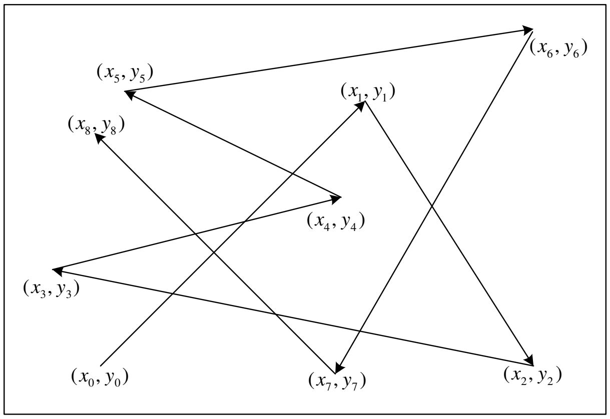


Fig.3. An illustration of RWP

Josh et al. [8] present a Random Waypoint model. In this model a mobile node is initially placed at random location in the simulation area with a destination and speed according to the random distribution. Then the mobile node moves from its current location to its destinations via a straight line. After arrival at the destination, the mobile node determines a pause time according to a random distribution, and after that chooses a new destination.

4.2 Analysis and Comparision

To analyze the character of these two mobility models, two important parameters inter-contact time and contact time are adopted. These two parameters describe the characteristics of connection opportunities of the network, i.e. how many and when they occur, how often and how long. Contact time is defined as the time interval during which the two nodes can keep contact. The time interval from this contact to next one is defined as inter-contact time during which nodes can't communicate [1]. Fig.4. gives an illustration. These two parameters are very important for opportunistic network. Contact time can help us determining the capacity of opportunistic networks. Inter-contact time is a parameter which strongly affects the feasibility of the opportunistic networking.

The new mobility model in this paper is named as Head Node Group Mobility Model (HNGM) here. We let each mobility model run 1000 seconds for one experiment and made 50 times similar experiments for each mobility model. Head node election period is set to be 1000 seconds here. Figure 5 and Figure 6 give the contact time distribution and inter-contact time distribution in different coordinate systems.

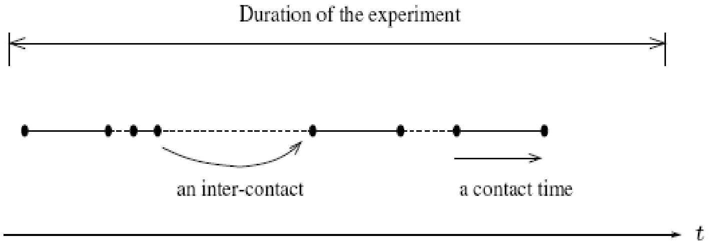


Fig.4. Contact and Inter-contact times for a pair of nodes[1].

In Fig. 5 (a), we can see that the new mobility model' inter-contact time distribution behave an exponential distribution plot using log-log coordinate system. In Fig. 5 (b), the contact time distributions show more evident difference. The RWP mobility models' curves are still exponential-like curves. In contrast, the shape of HNGM model' curves are changed. The HNGM model's curve is one kind of transition from exponential-like curve to power- law-like curve. This shows HNGM model behaves stronger group character than RWP model. In Fig. 6 we can see the difference between the curves more clearly using semi-log coordinate system. The exponential nature of the inter-contact time shown in Fig. 5 (a) becomes straight line form in Fig. 6 (a).

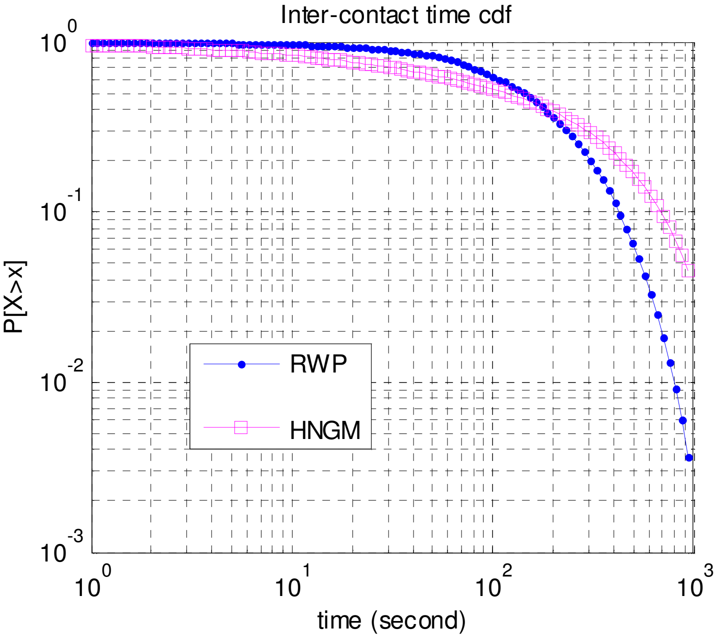


Fig5 (a)

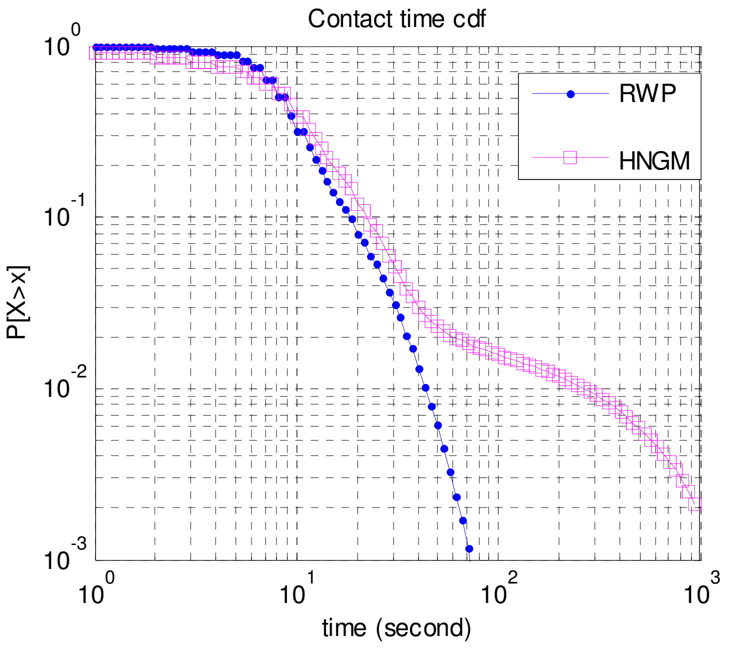


Fig5 (b)

Fig. 5. Inter-contact time cdf and contact time cdf in log-log coordinates

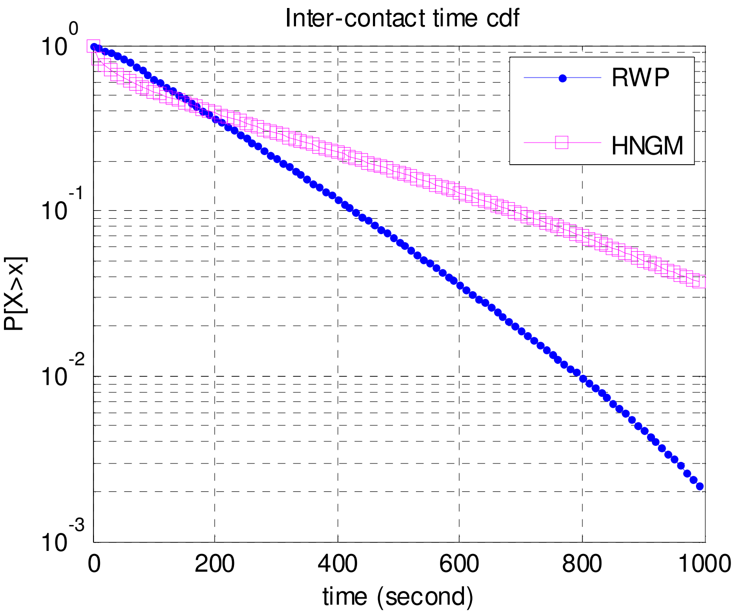


Fig6 (a)

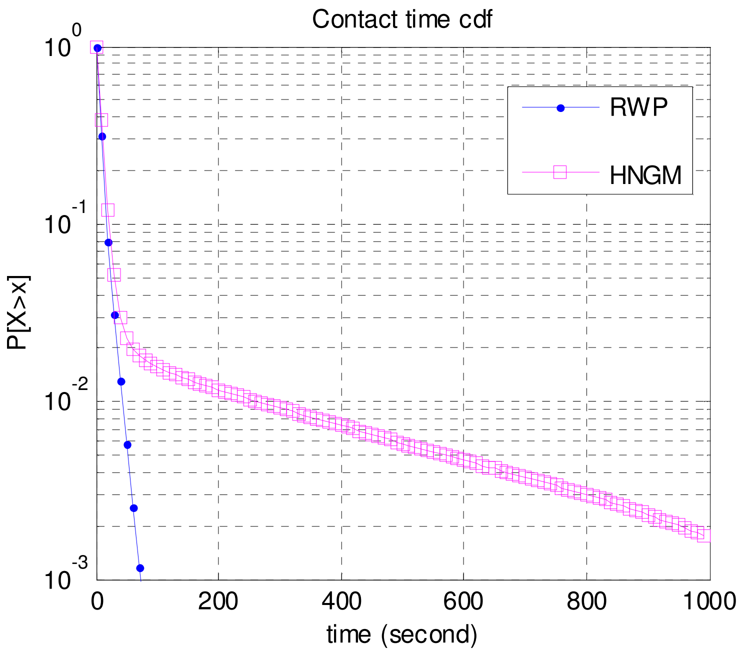


Fig 6 (b)

Fig. 6. Inter-contact time cdf and contact time cdf in semi-log coordinates.

5. Conclusion And Future Work

This paper proposed one new group mobility model using social Gauss-Markov scheme which is based on one social realistic mobility scenario. This paper also analyzes its characters about contact time, inter-contact time. The curves of contact time and inter-contact time show different shapes with the classical individual model RWP. By comparison, HNGM model shows obvious group character than RWP model. In the future work, the author will undertake further research on mobility about $0 < \alpha < 1$ case of Gauss-Markov process.

Acknowledgement

The author would like to thanks DFH Corporation support.

References

[1] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, “Pocket Switched Networks: Real-world mobility and its consequences for opportunistic forwarding,” Technical Report UCAM-CL-TR-617, University of Cambridge, Computer Laboratory, February 2005.

[2] J Broch, DA Maltz, DB Johnson, YC Hu, J Jetcheva,” Multi-hop wireless ad hoc network routing protocols”, in Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 1998), pages 85-97, 1998.

[3] Y. Ko and N.H. Vaidya, “Location-aided routing (LAR) in mobile ad hoc networks,” in Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom), pages 66-75, 1998.

[4] M. E. J. Newman, Detecting community structure in networks. Eur. Phys. J. B, 38, 321–330 (2004)..

[5] E. de Silva and M. Stumpf, “Complex networks and simple models in Biology”, J. R. Soc. Interface, 2 (2005), pp. 419-430..

[6] N. Przulj, D. G. Corneil, and I. Jurisica, “Modeling interactome:Scale- free or geometric?”, Bioinformatics, 20 (2004), pp. 3508-3515..

[7] D. J. Watts and S. H. Strogatz, “Collective dynamics of 'small-world' networks”, Nature, 393 (1998), pp. 440-442.

[8] M.E.J. Newman and M. Girvan. “Finding and evaluaing community structure in networks”. Physical Review E,68,2003.

[9] Mirco Musolesi, Cecilia Mascolo, “A community based mobility model for ad hoc network research”, in Proceedings of the 2nd international Workshop on Multi-Hop Ad Hoc Networks: From theory To Reality (REALMAN '06), Florence, Italy, May 2006, pp 31-38.

[10] R.L. Knoblauch, M.T. Pietrucha, M. Nitzburg. Field studies of pedestrian walking speed and start-up time. Transportation Research Board Records No. 1538, 1996.

[11] Newman, M. E. J. Fast algorithm for detecting community structure in networks. Phys. Rev. E 69,066133 (2004)

[12] Girvan, M. and Newman, M. E. J., Community structure in social and biological networks, Proc. Natl. Acad.Sci. USA 99, 8271–8276 (2002).

[13] L. Henderson. The statistics of crowd fluids. Nature, Vol. no. 229, 381-383, 1971.

[14] Finnis, K.K. and Walton, D. Field observations of factors influencing walking speeds. Ergonomics, 2006.

[15] LaPlante, John and Kaeser, Thomas P. “A History of Pedestrian Signal Walking Speed Assumptions”, in Proceedings of the 3rd Urban Street Symposium: Uptown, Downtown, or Small Town: Designing Urban Streets That Work, 2007.

[16] L. Pelusi, A. Passarella, and M. Conti, “Opportunistic networking: data forwarding in disconnected mobile ad hoc networks,” Communications Magazine, IEEE, vol. 44, no. 11, pp. 134 –141, november 2006.

[17] B. Liang and Z. Haas. Predictive distance-based mobility management for PCS networks. In Proceedings of the Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), March 1999

[18] E. Wong and B. Hajek, Stochastic Processes in Engineering Systems,Springer-Verlag, 1985

AUTHORS

GuoDong KANG is an telecommunication engineer of DFH Satellite Co., Ltd., 100094, Beijing, China.

INTENTIONAL BLANK

A TRIANGLE-TRIANGLE INTERSECTION ALGORITHM

Chaman L. Sabharwal and Jennifer L. Leopold

Missouri University of Science and Technology
Rolla, Missouri, USA – 65409
{Chaman,leopoldj}@mst.edu

ABSTRACT

The intersection between 3D objects plays a prominent role in spatial reasoning, geometric modeling and computer vision. Detection of possible intersection between objects can be based on the objects' triangulated boundaries, leading to computing triangle-triangle intersection. Traditionally there are separate algorithms for cross intersection and coplanar intersection. There is no single algorithm that can intersect both types of triangles without resorting to special cases. Herein we present a complete design and implementation of a single algorithm independent of the type of intersection. Additionally, this algorithm first detects, then intersects and classifies the intersections using barycentric coordinates. This work is directly applicable to (1) Mobile Network Computing and Spatial Reasoning, and (2) CAD/CAM geometric modeling where curves of intersection between a pair of surfaces is required for numerical control (NC) machines. Three experiments of the algorithm implementation are presented as a proof this feasibility.

KEYWORDS

Intersection Detection, Geometric Modeling, Classification Predicates, Spatial Reasoning, Triangle-Triangle Intersection.

1. INTRODUCTION

The intersection between 3D objects plays a prominent role in several areas including spatial reasoning[1], real-time rendering [2], collision-detection, geology [3], geometric modeling [4],and computer vision. Traditionally there are specialized algorithms for cross intersection and coplanar intersection. The intersection detection is a byproduct of actual intersection computations. Most of the time intersection detection should be done before the actual intersection. For example, in qualitative spatial reasoning, intersection detection is sufficient, actual intersection is not even needed, whereas in geometric applications like surface-surface intersection, it is desirable to have actual intersections. For early detection of intersection, we present a complete uniform algorithm independent of whether the triangles have cross intersection or coplanar intersection. Herein we illustrate this with an algorithm, its Python implementation is supported with output displayed in tables and figures. Typically, the boundary of a3D object is represented as a triangulated surface and a triangle-triangle intersection is the computational basis for determining intersection between objects. Since an object boundary may

David C. Wyld et al. (Eds) : WiMo, ITCSE, ICAIT, NC - 2015
pp. 27–35, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.51003

contain thousands of triangles, algorithms to speed up the intersection detection process are still being explored for various applications, sometimes with a focus on innovations in processor architecture[5].

In qualitative spatial reasoning, spatial relations between regions are defined axiomatically using first order logic [6] or the 9-Intersection model [1]. It has been shown in [7] that it is sufficient to define the spatial relations by computing 4-Intersection predicates, (namely, Interior–Interior (IntInt), Boundary–Boundary (BndBnd), Interior–Boundary (IntBnd), and Boundary–Interior (BndInt)) instead of 9-Intersections. In order to implement these algorithms, we must first implement the triangle-triangle intersection determination.

In geometric modeling, the surface-surface intersection problem occurs frequently. Since a free-form surface is represented as triangulated mesh, intersection between 3D objects amounts to detecting and computing intersection between 3D triangles. The algorithm presented here is extremely useful for geometric modeling where intersection between objects occurs thousands of times. For geometric applications cross intersection is most often used to the determine curves of intersection between surfaces [4]. Most of the approaches to these problems are based on code optimization of same or similar cross intersection algorithms. No attempt is made to design a single algorithm that handles both intersections simultaneously. Herein we present a new algorithm that fills the gap.

This paper is organized as follows: Section 2 discusses the background of possible cross and coplanar intersection between a pair of triangles. It describes the cross intersection, coplanar area intersection algorithm, and composite algorithms for general triangles. Section 3 develops the new generic single algorithm for triangle-triangle intersection, and classifies the intersections. Section 4 describes implementation experiments and the timing results. Section 5describes the two of the applications where this approach is directly applicable. Section 6concludes and references are given in Section 7.

2. BACKGROUND: TRIANGLE-TRIANGLE INTERSECTION

There is an abundance of papers devoted to intersection between a pair of triangles [8,9,10,11,12]. Interestingly, most of the papers concentrate on cross intersection, they simply reinvent the algorithm and optimize the code to implement it slightly differently and more efficiently, with no innovation. The cross intersection may result in a no intersection, a single point intersection or a line intersection. For coplanar triangles, it can result in area intersection as well. For coplanar intersection, such cross intersection algorithms do not work, there are separate algorithms for area intersection [10,11,12]. In these approaches, they first try to compute the intersection. Whether intersection is found or not, then it leads to conclude on the existence of intersection. The paper [12]surveyed various approaches for determining the cross intersection detection, and developed a fast vector version of the cross intersection detection, as well as classification of the type of intersection. The papers [8] and [11] also compare hardware implementation of their algorithm based on the number of arithmetic operations: +, -, *, /. Another paper [10] also compares the optimized intersection times. The papers[10,11] considered an approach for determining the intersection detection covering coplanar intersection, using edge to edge intersections. These approaches [10,11,12] led to two separate algorithms one for cross and one for coplanar intersection. There is no single algorithm that can handle both cross and coplanar intersection. We present a new algorithm that is analytically different, exhaustive and more rigorous than the previous algorithms [8,9,10,11,12]. It computes intersection using barycentric coordinates and vector equations very judiciously. Logical rather than computational tests are used to detect whether the intersection is a single point, or a line or an area. This new algorithm encompasses all cases implicitly and is different from previous ones in that we use only one equation rather than different algorithm for each special case. We show that it is possible to detect the existence of intersection in a uniform way before the precise intersection computations are computed.

2.1 Specialized Intersection Methods and Algorithms

Here we describe the conventional approach to triangle-triangle intersection in terms of two algorithms, one for cross intersection and one for coplanar intersection. Then we derive the composite algorithm for triangle-triangle intersection. Here we give the line intersection for cross intersection and area intersection for coplanar triangles separately and combine the two into one algorithm as is conventionally done. Then in Section 3 we present our new algorithm, which is a single algorithm that employs a more singular, seamless approach.

2.1.1 The Triangle-Triangle Line Intersection Algorithm

The cross intersection algorithm encompasses the single point and edge intersection cases. Here we give a solution *different* from previous approaches.

Algorithm for cross intersection line Algorithm
Input: Two triangles ABC and PQR
Output: Determine if they cross intersect. Return true if they intersect, otherwise return false.
Boolean crossInt (tr1 = ABC, tr2 = PQR)

The vector equations for two triangles ABC and PQR are

$$R_1(u, v) = A + u\,U + v\,V,\ 0 \leq u, v, u + v \leq 1$$

$$R_2(s, t) = P + s\,S + t\,T,\ 0 \leq s, t, s + t \leq 1$$

where U = B-A, V = C - A, and S = Q-P, T = R – P are direction vectors along the sides of triangles.

Let $N_1 = UxV$, $N_2 = SxT$ be normals to the planes supporting the triangles. The triangles intersect if there exist some barycentric coordinates (u, v) and (s, t) satisfying the equation

$$A + u\,U + v\,V = P + s\,S + t\,T$$

Since $N_1 x N_2 \neq 0$ for *cross* intersecting triangles, and S and T are orthogonal to N_2, the dot product of this equation with N_2 eliminates S and T from the above equation to yield

$$u\,U \bullet N_2 + v\,V \bullet N_2 = AP \bullet N_2$$

This is the familiar equation of a line in the uv - plane for real variables u, v. The vector equation using real parameter λ becomes

$$(u, v) = AP \bullet N_2 \frac{(U \bullet N_2, V \bullet N_2)}{U \bullet N_2^2 + V \bullet N_2^2} + \lambda(V \bullet N_2, -U \bullet N_2)$$

If there is a λ satisfying three equations such that $0 \leq u, v, u + v \leq 1$, then the triangles are ensured to intersect. The solution in λ is the range of values $\lambda_m \leq \lambda \leq \lambda_M$ for some λ_m and λ_M.

This gives the line of intersection of uv-parameter triangle with the st-parameter plane. Similarly the line of intersection of st-triangle with the uv-plane is computed. Then the common segment if any is the line intersection between the two triangles, for details see [9,13].This algorithm works only if the triangles *cross* intersect.

2.1.2 The Triangle-Triangle Area Intersection Algorithm

The area intersection is possible for coplanar triangles only. If the triangles ABC and PQR are coplanar and a vertex of PQR is in the interior of ABC (or the converse is true), then an area intersection occurs. If there is no *edge-edge* intersection and no vertex of one triangle is inside the other triangle (or the converse is true), then they do not intersect, hence they are disjoint. The input, output, method prototype and pseudocode are given below.

Algorithm for area intersection
Input: Two triangles ABC and PQR coplanar
Output: If they do not intersect, then return false otherwise return true and the intersection area.
Boolean coplanarInt (tr1 = ABC, tr2 = PQR)

The derivation of the algorithm is based on extensive use of the intersections of edges of ABC triangle with the edges of the PQR triangle and collecting the relevant interactions to form the area if any, for details see [11]. This algorithm can solve for intersection of *coplanar* triangles only.

The classical approach is to use crossInt (tr1, tr2), algorithm from section 2.1.1, when the triangle cross, and to use the second algorithm coplanarInt (tr1, tr2), algorithm from section 2.1.2, for coplanar triangles.

Thus separate algorithms are used to determine intersections on case-by-case bases. There is no single algorithm that detects intersection. We present a new single algorithm that does not depend on case-by-case intersections.

3. ALGORITHM FOR TRIANGLE-TRIANGLE INTERSECTION

Here we present a single algorithm that is analytically different, more comprehensive, robust and rigorous; it is implicitly capable of handling any specific type of intersection, which may be no intersection, a single point, a segment or an area. The Triangles may be coplanar or crossing. This single algorithm is not a modification of any previous algorithm, it is a new approach different from the other strategies.

The Main Algorithm
Input: Two triangles ABC and PQR in 3D, triangles may be coplanar or crossing
Output: If the triangles do not intersect, return false otherwise return true and the intersection, which may be single point, a line segment or an area.
Boolean triTriIntersection (tr1 = ABC, tr2 = PQR)

The triangles ABC and PQR are
$X = A + u\,U + v\,V$ with $U = B - A$, $V = C - A$, $0 \leq u, v, u + v \leq 1$
$X = P + s\,S + t\,T$ with $S = Q - P$, $T = R - P$, $0 \leq s, t, s + t \leq 1$

To detect the intersection between the triangles ABC and PQR, we must attempt to solve $A + u\,U + v\,V = P + s\,S + t\,T$for some values $0 \leq u, v, u + v, s, t, s + t \leq 1$,

Rearranging this equation, we have
$uU + v\,V = AP + s\,S + t\,T$ (1)
where$AP = P - A$ is a vector

To solve the equation (1) for u,v, we dot equation (1) with (UxV)xV and (UxV)xU, we quickly get u and v as

$u = -\,(\,\gamma\bullet V + s\,\alpha\bullet V + t\,\beta\bullet V)$
$v = \gamma\bullet U + s\,\alpha\bullet U + t\,\beta\bullet U$
$u + v = \gamma\bullet\,(U - V) + s\,\alpha\bullet\,(U - V) + t\,\beta\bullet(U - V)$
where α, β, γ, δ are:
$\delta = (U\text{x}V)\bullet(U\text{x}V)$, $\alpha=\dfrac{S\times(U\times V)}{\delta},\beta=\dfrac{T\times(U\times V)}{\delta},\gamma=\dfrac{AP\times(U\times V)}{\delta}$

In order to satisfy the constraints $0 \leq u, v, u + v \leq 1$, we get three inequalities in parameters s and t

(a) $-\,\gamma\bullet U \leq \alpha\bullet U\,s + \beta\bullet U\,t \leq 1 - \gamma\bullet U$
(b) $-\,1 - \gamma\bullet V \leq \alpha\bullet V\,s + \beta\bullet V\,t \leq -\,\gamma\bullet V$
(c) $-\,\gamma\bullet\,(U - V)\leq\alpha\bullet(U - V)\,s + \beta\bullet(U - V)\,t \leq 1- \gamma\bullet(U - V)$

inequalities we first eliminate t to get the possible range for s values, then solve for the range

These inequalities (a) - (c) are linear in s and t and are of the general form

$$m \leq a\,s + b\,t \leq n$$

To solve these for possible t values as t(s). If there is no solution, then the algorithm returns false. If there is a solution, then it return s, t values as

$$s_m \leq s \leq s_M\,,\ t_m(s) \leq t \leq t_M(s).$$

This discussion is summarized and the intersection points are classified as follows:

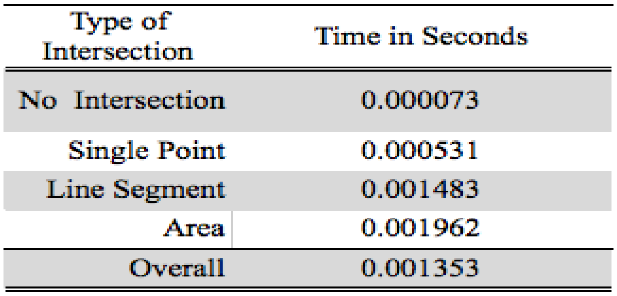
```
if the algorithm returns false,
    No Intersection
elseif ($s_m = s_M$ and ($t_m(s) = t_M(s)$ for $s_m \leq s \leq s_M$)
    Single Point Intersection
elseif ($s_m$=$s_M$or ($t_m(s) = t_M(s)$ for $s_m \leq s \leq s_M$)
    Line segment intersection common to two triangles
else
    Area Intersection common to two triangles
```

This completes the discussion of our algorithm for intersection between triangles.

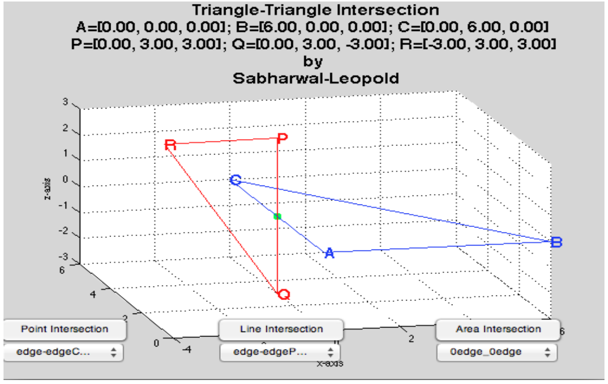
4. EXPERIMENTAL RESULTS

The algorithm is implemented in MacPython 3.3.3. The test time results are obtained via Python time it utility. Time tests were performed on Apple Macintosh OS X processor (2.2 GHz intel Core i7). The average time for no intersection, single point intersection (0D), line intersection (1D) and area intersection (2D) are measured in seconds. The program is executed 100 times on each of 22 sample triangle pairs. The intersection times shown in Table 1 are neither optimized nor hardware embedded, they include classification of intersections also. Times are shown as proof of concept that this single integrated algorithm works reliably on all triangle pairs. The test data domain consists of synthetic triangles that have every possible type of intersection. For example, the times for no intersection are averaged over 100 runs with three samples. Similarly for single point intersection six sample pairs are averaged over 100 runs, then four samples are used for line segment and averaged over 100 runs, and finally nine sample triangle pairs are used and averaged. The composite average intersection time is computed. The test time statistics are displayed in Table 1. We also give three sample run output figures for examples of single point intersection, a line intersection and an area intersection. The Matlab software is used to draw the figures, see Fig. 1-3. The user interface allows user to select any type of the possible triangle-triangle pair for intersections and it displays the corresponding triangle and the intersection. For the sake of space consideration, one of the 0D (single point) intersections, see Fig. 1, one of the 1D line segment intersections, see Fig. 2; and one of 2D area intersections, see Fig. 3, are displayed here.

TABLE 1. EXECUTION AVERAGE TIMES OF ALGORITHM IN SECONDS



Type of Intersection	Time in Seconds
No Intersection	0.000073
Single Point	0.000531
Line Segment	0.001483
Area	0.001962
Overall	0.001353



Triangle-Triangle Intersection
A=[0.00, 0.00, 0.00]; B=[6.00, 0.00, 0.00]; C=[0.00, 6.00, 0.00]
P=[0.00, 3.00, 3.00]; Q=[0.00, 3.00, -3.00]; R=[-3.00, 3.00, 3.00]
by
Sabharwal-Leopold

Fig. 1. Single Point Intersection

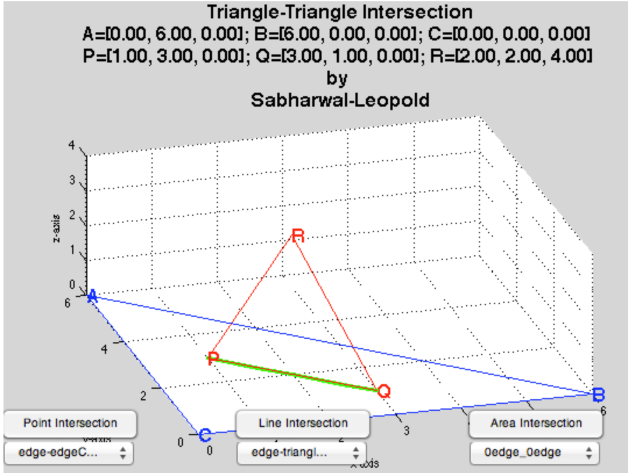

Triangle-Triangle Intersection
A=[0.00, 6.00, 0.00]; B=[6.00, 0.00, 0.00]; C=[0.00, 0.00, 0.00]
P=[1.00, 3.00, 0.00]; Q=[3.00, 1.00, 0.00]; R=[2.00, 2.00, 4.00]
by
Sabharwal-Leopold
z-axis
y-axis
x-axis
A
B
C
P
Q
R
Point Intersection
edge-edgeC...
Line Intersection
edge-triangl...
Area Intersection
0edge_0edge

Fig. 2. Line Segment Intersection

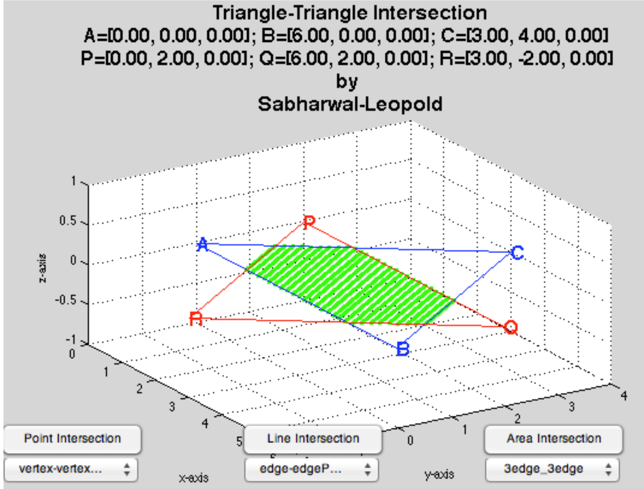

Triangle-Triangle Intersection
A=[0.00, 0.00, 0.00]; B=[6.00, 0.00, 0.00]; C=[3.00, 4.00, 0.00]
P=[0.00, 2.00, 0.00]; Q=[6.00, 2.00, 0.00]; R=[3.00, -2.00, 0.00]
by
Sabharwal-Leopold
z-axis
x-axis
y-axis
A
B
C
P
Q
R
Point Intersection
vertex-vertex...
Line Intersection
edge-edgeP...
Area Intersection
3edge_3edge

Fig. 3. Area Intersection

5. Applications of Triangle-Triangle Intersection

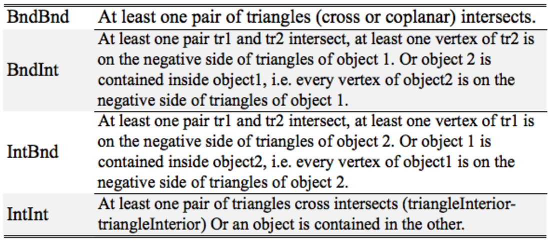
Here we describe two of the applications where triangle-triangle intersection is directly applicable and is used extensively: qualitative spatial reasoning and geometric modeling. It is not limited to these two applications, other applications include virtual reality, and computer vision.

5.1 Qualitative Spatial Reasoning

In Qualitative Spatial Reasoning, the spatial relations are determined by the 9-Intersection/4-Intersection model[7, 8]. That is, for any pair of objects A and B, the interior-interior intersection predicate, IntInt(A, B), has true or false value depending on whether the interior of A and the interior of B intersect without regard to precise intersection. Similarly IntBnd(A, B) represents the truth value for the intersection of the interior of A and the boundary of B, and BndBnd(A,B) represents the predicate for the intersection of the boundaries of A and B. These four qualitative spatial reasoning predicates are sufficient to define RCC8 spatial relations[7].

In the application VRCC-3D+[13], the boundary of an object is already triangulated; that is, we will need to intersect pairs of only triangles. To reduce the computational complexity, the algorithm uses axis aligned bounding boxes (AABB) to determine the closest triangles which may possibly intersect[13]. Table 2 is a characterization of the intersection predicates, which subsequently can be used to resolve the eight RCC8 relations. Here we assume all normals are oriented towards the outside of the object. Each characterization in Table 2 describes when the associated predicate is true. If the truth test fails, then other triangles need to be tested. If no pair of triangles results in a true value, then the result is false.

TABLE 2. CHARACTERIZATION OF INTERSECTION PREDICATES



BndBnd	At least one pair of triangles (cross or coplanar) intersects.
BndInt	At least one pair tr1 and tr2 intersect, at least one vertex of tr2 is on the negative side of triangles of object 1. Or object 2 is contained inside object1, i.e. every vertex of object2 is on the negative side of triangles of object 1.
IntBnd	At least one pair tr1 and tr2 intersect, at least one vertex of tr1 is on the negative side of triangles of object 2. Or object 1 is contained inside object2, i.e. every vertex of object1 is on the negative side of triangles of object 2.
IntInt	At least one pair of triangles cross intersects (triangleInterior-triangleInterior) Or an object is contained in the other.

This characterizes the intersection predicates that help in resolving the RCC8 relations.

5.2 Geometric Modeling

In geometric modeling, the surface-surface intersection problem occurs frequently. In CAD/CAM, the objects are represented with surface boundaries using the ANSI (Brep) model. Intersection between 3D surfaces amounts to detecting and computing intersection between 3D triangles. Briefly, a pair of surfaces is subdivided using axis aligned bounding boxes (AABB) until the surfaces are reasonably planar and bounding boxes intersect. Then the plane surfaces are triangulated and the triangles are tested for cross-intersection and the intersection computed. The intersection segments are linked together to form curves of surface-surface intersection. The curves may be open curves, closed curves, or even a combination of both [4]. The algorithm presented here is extremely useful for geometric modeling where intersection between objects occurs thousands of times. For geometric applications cross intersection is most often used to obtain the line segment of intersection. Detailed analysis and implementation of the most comprehensive surface/surface intersection algorithm may be found in [4].

6. CONCLUSION

Herein we presented a single algorithm for the complete design and a robust implementation of a complete framework for determining and characterizing the triangle-triangle intersections. In contrast to other single track algorithms, this approach is a generic technique to detect any type cross or coplanar intersection using only one algorithm. The classification of intersections is based on logical tests on parametric coordinates rather than computational arithmetic tests in Cartesian coordinates. Thus our algorithm not only detects the existence but also classifies and computes the intersection as a single point, a line segment, or an area whichever the case may be. The algorithm provides more information than required by spatial reasoning systems. Consequently, we hope the additional information including classification of 3D intersection presented herein will be useful in other related applications.

REFERENCES

[1] Max J. Egenhofer, R. Franzosa, Point-Set topological Relations, International Journal of Geographical Information Systems 5(2), pp. 161 - 174, 1991.

[2] Oren Tropp, Ayellet Tal, IlanShimshoni, A fast triangle to triangle intersection test for collision detection, Computer Animation and Virtual Worlds, Vol17 (50), pp.527 - 535, 2006.

[3] G. Caumon, Collon - Drouaillet P, Le Carlier de Veslud C, Viseur S, Sausse J (2009) Surface - based 3D modeling of geological structures. Math Geosci 41:927–945, 2009.

[4] E.G. Houghton, Emnett R.F., Factor J.D. and Sabharwal C.L., Implementation of A Divide and Conquer Method For Surface Intersections; Computer Aided Geometric Design Vol.2, pp. 173 - 183, 1985.

[5] Ahmed H. Elsheikh, Mustafa Elsheikh, A reliable triangular mesh intersection algorithm and its application in geological modeling, Engineering with Computers, pp.1 - 15, 2012.

[6] D. A. Randell, Z. Cui, and A.G. Cohn, A Spatial Logic Based on Regions and Connection., KR, 92, pp. 165–176, 1992.

[7] C. Sabharwal and J. Leopold, “Reducing 9-Intersection to 4-Intersection for Identifying Relations in Region Connection Calculus”, Proceedings of the 24th International Conference on Computer Applications in Industry and Engineering (CAINE 2011), Honolulu, Hawaii, Nov. 16-18, 2011, pp. 118-123, 2011.

[8] Mo¨ller T. A fast triangle - triangle intersection test. Journal of Graphics Tools, 1997; 2(2): 25–30.

[9] Held M. ERIT a collection of efficient and reliable intersection tests. Journal of Graphics Tools 1997; 2(4): pp. 25–44, 1997.

[10] Badouel Didier, An Efficient Ray - Polygon Intersection, Graphics Gems (Andrew S. Glassner, ed.), Academic Press, pp. 390 - 393, 1990.

[11] Guigue P, Devillers O. Fast and robust triangle - triangle overlap test using orientation predicates. Journal of GraphicsTools 2003; 8 (1): pp. 25–42, 2003.

[12] Chaman Sabharwal, Jennifer Leopold, and Douglas McGeehan,:Triangle-Triangle Intersection Determination and Classification to Support Qualitative Spatial Reasoning, Polibits, Research Journal of Computer Science and Computer Engineering with Applications Issue 48 (July–December 2013), pp. 13–22, 2013.

[13] N.Eloe, J. Leopold, C. Sabharwal, and Z. Yin, “Efficient Computation of Boundary Intersection and Error Tolerance in VRCC-3D+ ”, Proceedings of the 18h International Conference on Distributed Multimedia Systems (DMS’12), Miami, FL, Aug. 9 - 11, 2012, pp. 67 – 70, 2012.

INTENTIONAL BLANK

DESIGN AND IMPLEMENTATION OF INTEL-SPONSORED REAL-TIME MULTI-VIEW FACE DETECTION SYSTEM

Bohua Gan[1], Vincent Chang[1], Guanying Wang[1], Xiuli Pan[1], Guan Wang[1],
Naihai Zou[2], and Felming Feng[2]

[1]University of Michigan—Shanghai Jiao Tong University Joint Institute,
Shanghai, China
ganbohua@gmail.com
[2]Intel Asia-Pacific Research and Development Ltd, Shanghai, China
nanhai.zou@intel.com

ABSTRACT

The paper introduces a case study of design and implementation of Intel-sponsored real-time face detection system conducted in University of Michigan—Shanghai Jiao Tong University Joint Institute (JI). This work is teamed up totally by 15 JI students and developed in three phases during 2013 and 2014. The system design of face detection is based on Intel High Definition (HD) 4000 graphics and OpenCL. With numerous techniques including the accelerated pipeline over CPU and GPU, image decomposition, two-dimensional (2D) task allocation, and the combination of Viola-Jones algorithm and continuously adaptive mean-shift (Camshift) algorithm, the speed reaches 32 fps for real-time multi-view face detection. Plus, the frontal view detection accuracy obtains 81% in Phase I and reaches 95% for multi-view detection, in Phase III. Furthermore, an innovative application called face-detection game controller (FDGC) is developed. At the time of this writing, the technology has been implemented in wearable devices and mobile with Intel cores.

KEYWORDS

Multi-view face detection, real-time, graphic processing unit (GPU), OpenCL, face-detection game controller (FDGC).

1. INTRODUCTION

Mobile commerce has reached unprecedented height: global smart phone shipments have over 1.16 billion in 2014. Meanwhile the competition of mobile processor chip market is fierce between Qualcomm, MediaTek (MTK), Samsung, and Intel, etc. Qualcomm and MTK achieve above 30 percent of market share each, following by Samsung's 20 percent of the share. Unlike its predominant advantage in the PC fields, Intel gains only 2.81 percent share in 2014 which nearly missed the mobile revolution [1].

David C. Wyld et al. (Eds) : WiMo, ITCSE, ICAIT, NC - 2015
pp. 37–47, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.51004

Intel’s core competitiveness hinges up advanced semiconductor fabrication process and technology. In addition, effective and powerful software solutions make Intel the leading position in the semiconductor industry. Furthermore, Intel has various System on Chip (SoC) for different platforms and billions expenditures of Research & Development every year. Advanced technologies such as SoC fabrication, Bluetooth, and wireless charging etc. are widely used in Intel-inside wearable and mobile devices.

RealSense technology which incorporates face recognition and three-dimensional (3D) cameras is used on the robots, unmanned aerial vehicles, as well as a new interactive mode of 3D immersive experience. This innovative mode could eventually make user free from the constraint of keyboards and passwords.

As the most important part of RealSense technology, face detection has been in-depth studied by Intel. This corporation has sponsored three-phase real-time face detection capstone design project (CDP) on University of Michigan－Shanghai Jiao-Tong University (Joint Institute) including 15 students during 2013 and 2014 [2]. It aims at innovating in the face detection technology and improving the performance of detecting speed, and accuracy etc.

The speed of original version which at start of Phase I is 3 fps for 480p videos frontal view face detection and it reaches 9 fps at the end of Phase I. With the combination of image decomposition and two-dimensional (2D) task allocation in Phase II, the speed is increased by 3 times and reaching 30 fps. Combining the Camshift tracking in Phase III, multi-view face detection speed is 32 fps finally. Plus, the detection accuracy obtains 81% in Phase I and reaches 95%, in Phase III. In addition, an interesting interactive application called Face-Detection Game Controller (FDGC) is developed. The program’s conception, design details and testing results are presented in Section 2. Conclusions are given in Section 3.

2. DESIGN AND IMPLEMENTATION OF THE REAL-TIME MULTI-VIEW FACE DETECTION SYSTEM

As Table 1 shows, there are three phases in the face detection CDP, including two phases in 2013 and one phase in 2014. Every phase lasts about 12 weeks and involves 5 students.

2.1. OpenCL Accelerated Face Detection (Phase I, summer, 2013)

The first step is a literature survey and an objective analysis. According to the capability of Intel’s HD 4000 graphics and OpenCL, the customer’s requirements are definite; they are performance improvement, working prototype, and deliverable product separately. To meet the customer requirements, the detailed engineering specifications are generated in Week 1. There are mainly three aspects which are accuracy, efficiency, and innovative application [3]. As Table 2 shows, the objective of efficiency is reaching over 10fps for 480p videos, detecting 85% frontal view faces correctly, and implementing an innovative application with bulged faces detection.

In Week 2, the entire video processing flows are definite as shown in Figure 1.The entire processing function is decomposed into three workflows: video pre-processing, face detection and accelerated process &innovative apps.

- Video Processing comprises of a series of preprocessing on the input video. It changes the video stream into a different type of data which will make subsequent computation easier.
- Face detection deals with the computation of images to locate human faces and find the accurate sizes of them.
- Accelerated process will improve the speed exponentially and innovative applications are concerned with how to implement application programs after face detection is done.

Through brainstorming, in which every member expressed their ideas of possible solution concept related to the projects and Quality Function Deployment according to the discussion, the combination of using Segment into Shots, Viola-Jones Algorithm, and GPU acceleration was selected as the final design.

Table 1. Overview of all the tasks involved in developing this 3-phase work during 2013 and 2014. The task marked with * represents the primary contribution for acceleration.

Week(nth)	Phase I (Summer, 2013)	Phase II (Fall, 2013)	Phase III (Summer, 2014)
1	Literature Survey, Objective Analysis		
2	Function Decomposition &Algorithm Choosing	Previous Work Rebuilding	
3	Image Face Detection		
4	Video Stream Face Detection	Analysis & Design	**Skin-Color-Based Image Segmentation Implementation***
5			
6	Performance Improvement	**Pipeline of Accelerated Concept over CPU and GPU & Image Decomposition and Task Allocation***	
7	Innovative Applications Design		**Viola-Jones detection + Camshift tracking Pipeline Building***
8			
9	Debug & Test		**Tuning Face Recognition using Principal Components Analysis***
10			
11	Final Delivery	UI Design, User Feedback and Update	**Detection Box Merge Optimization & Image Preprocessing Pipeline Tuning***
12		Final Delivery	

Table 2. Engineering specifications of Phase I project, OpenCL acceleration face detection

Engineering Specifications	**Description**
Accuracy	Detect 85% faces correctly
Efficiency	10-15 fps for 480p videos with frontal view face detection
Innovative Application	Adding bulged effect on the detected faces

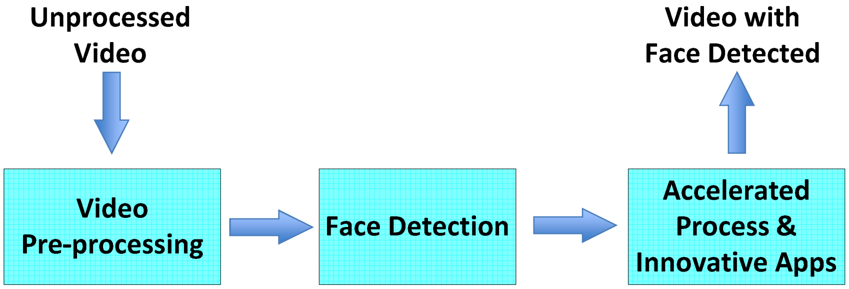


Fig. 1. Video processing flow in Phase I

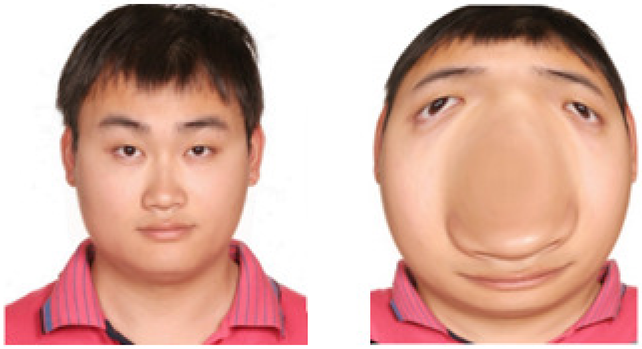

(a)
(b)

Fig. 2. A special effect on the detected face in Phase I: (a) Original image (b) The image added with the bulged effect on the detected face (Courtesy of Ref. [3]).

- Segment into shots, images are extracted on a certain rate and the number of extracted images is reduced.
- Viola-Jones algorithm, it provides fast and efficient face detection framework [4].
- GPU acceleration, images separated from video streams can be dealt in parallel with GPU. Therefore the speed is increased

From Week 3 to Week 5, the image and video stream face detection are completed. Due to slow data exchange between CPU and GPU and the many sub-images that needed to be processed, the performance of the initial version running in Intel HD 4000 graphics did not meet the expectations. In order to improve the speed and accuracy, processing all sub-images in parallel, exchanging data only when necessary, and buffering small exchanges are applied. Therefore, the frontal view face detection speed has changed from 3 fps to 9 fps for 480p videos which is little less than expectation. The accuracy is 81% in average.

Innovative application design is an important part of this project. And according to the mentor from Intel, students are encouraged to think outside of the rules. The team implements bulge effects on the detected faces as shown in Figure 2, and faces in the video are all bulged smoothly. In Week 9 and 10, the team focused on testing and debugging. They found several bugs when running with Intel hardware. This helps Intel improve the performance of the hardware, and it is also beneficial to the following capstone design teams.

2.2. OpenCL Accelerated Face Detection (Phase II, fall, 2013)

Intel requires the team of Phase II to bring the Phase I face detection program to a new level with the help of Intel Beignet OpenCL implementation. Specifically, two specific requirements were mentioned by Intel:

- Optimize the existing parallel computing algorithm.
- Come up with innovative ideas on applications and realize some of them [5].

Based on first three weeks' literature survey, objective analysis, and previous work rebuilding, the engineering specifications are definite as shown in Table III. The goal of this phase is faster, more accurate and innovative.

The pipeline of accelerated concept over CPU and GPU has displayed in Figure 3 and was decided particular by the team in Weeks 4 and 5. The general process is to first turn video streams into still frame which will divert into gray images in CPU. Classifier will get the image faces data from gray images, which include information of the size and location, and then add special effects on the faces in GPU. Thus making GPU and CPU work simultaneously, the speed increases obviously and reaches 20 fps for 480p videos.

In order to further accelerate the process, some ideas were considered and two of them were used. They proved to be available in Phase I. In this phase, the extensive improved solutions, including image decomposition and task allocation, are designed in detail. Image decomposition is to make pre-processing and classifiers more efficient, and task allocation is to make the acceleration over CPU and GPU more efficient [6]. Due to the effort in these two factors, it reaches 30fps for 480p videos which is higher than Phase I and the accuracy achieves 94% in average.

(1) Image decomposition. Since a GPU has the property of parallel calculation, the images were separated into small parts and then transformed together into the GPU. With the little parts calculated at the same time, the calculated speed is raised remarkably.

(2) Task allocation. The tasks are sorted in a chain before, but the cores of GPUs are distributed like a 2D matrix. It takes quite a lot of time to match up each task with each core of a GPU. In this phase, tasks are put into a 2D task matrix where each task is matched up with the cores in a GPU automatically, and the matching time is greatly reduced.

Innovative application is the vital engineering specification in Phase II, so the team pays much more attention on user interface and creative application design. In the final version of the program, it supports two types of inputs, which are still images and video streams. As shown in Figure 4, users can click the "Video" button or "Image" button to choose different input forms. For choosing video as input, users need to make sure that the computer has been connected to a web camera firstly. The program will do real-time face detection on the videos captured by the camera. For choosing image as input, users need to click the “Browse” button as the input document.

Table 3. Engineering Specifications in Phase II.

Engineering Specifications	**Description**
Accuracy	Detect 90%-95% faces correctly
Efficiency	20-30 fps for 480p videos with frontal view faces detection
Innovative Application	Adding different effects on the detected faces

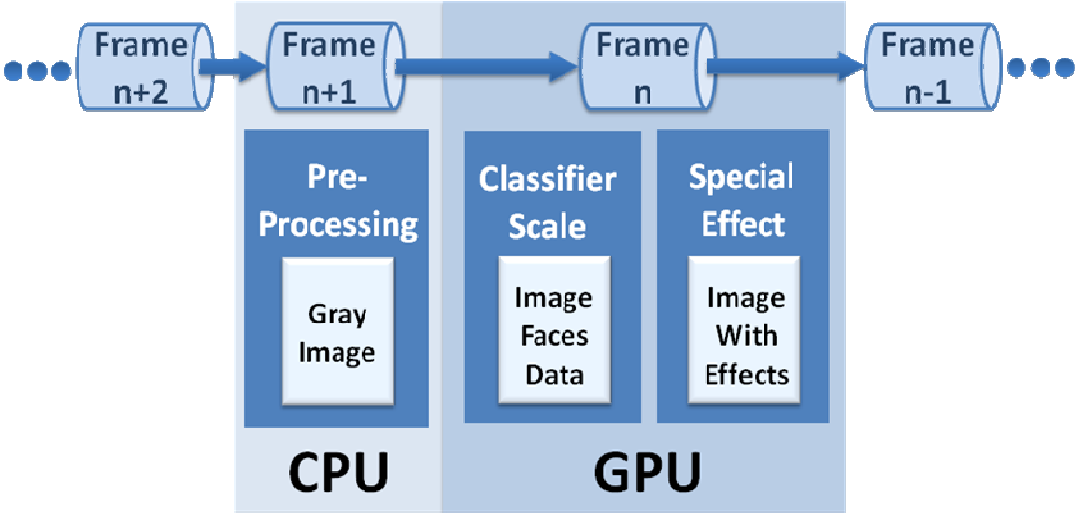
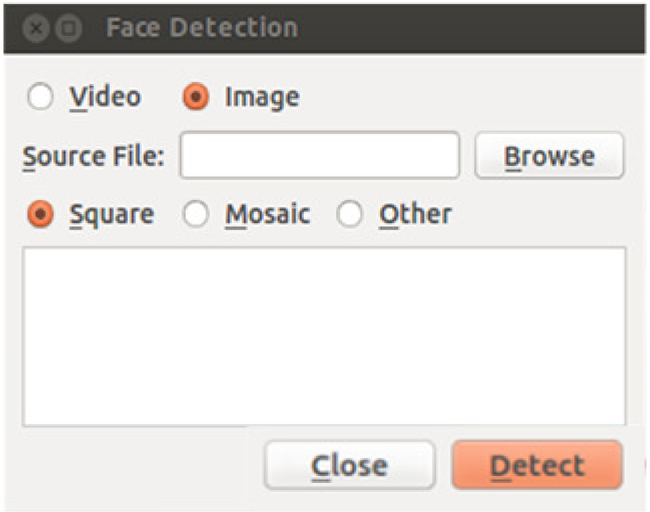


Fig. 3. Design of the pipeline for face detection acceleration over CPU and GPU in Phase II.



○ Video ● Image

Source File:

● Square ○ Mosaic ○ Other

Fig. 4. User Interface of the accelerated face detection system in Phase II (Courtesy of Ref. [6]).

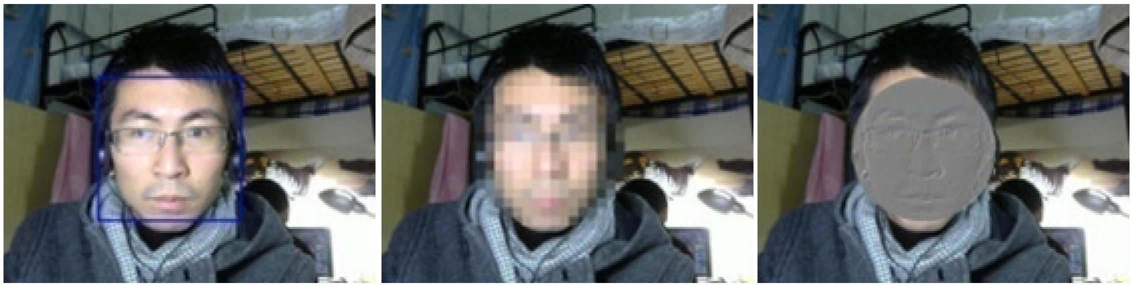


Fig. 5. Specialeffects on the detected faces in Phase II: (a) Rectangle (b) Mosaic (c) Comic relief (Courtesy of Ref. [6]).

In addition, users can click three options for face displaying as shown in Figure 5. They are the rectangle, mosaic and comic relief effects.

2.3 OpenCL based Face Detection and Recognition Algorithms Tuning (Phase III, summer, 2014)

Different from previous phases, Phase III is asked to realize the tilted face detection, and detecting multi-view faces makes the product satisfactory with the commercial requirements as shown in Table 4 [7].

Three image pre-processing steps which have shown in Figure 6 are used to remove background. Detection area is significantly reduced, and the speed is boosted with these techniques.

- Skin Segmentation is used to identify skin region. Firstly, RGB image is converted to YCrCb image which ignores illumination, then skin model is applied to obtain segmented binary image. Meanwhile, some topological methods, like dilation, erosion and Gaussian smooth, are used to denoise and smooth the binary image [8].
- Bounding Boxes Finding is used to find the located detection area. Label the connected white components in the binary image at first and find contours of each component. Then rectangle windows are drawn to bounding each contour.
- Merge the overlapped boxes into larger joint boxes. The merge operation can reduce duplication caused by the overlap which will reduce the execution time.

Table 4. Engineering Specifications in Phase III.

Engineering Specifications	**Description**
Accuracy	Detect 90%-95% faces correctly
Efficiency	Over 30 fps for 480p videos with multi-views detection
Functionality	Tilted face detection
Innovative Application	Face-Detection Game Controller

From Week 7 to Week 10, the team focused on increasing the accuracy, solving the problem of detecting tilted faces and video face detection which based on the image face detection. Camshift (Continuously Adaptive Mean-shift) [9] can detect tilted face in a large scale of angle and the inner core of it involves more float and matrix calculation than Viola-Jones. The solution which builds a pipeline that leverages the consistency of video with Viola-Jones and Camshift algorithm. The new pipeline for the video multi-view face detection shows in Fig. 7 and there are three steps as follows:

- Using Viola-Jones Algorithm to detect faces every 10 frame.
- Using Camshift Algorithm to track those faces for 9 continuous frames.
- Iterate previous two steps.

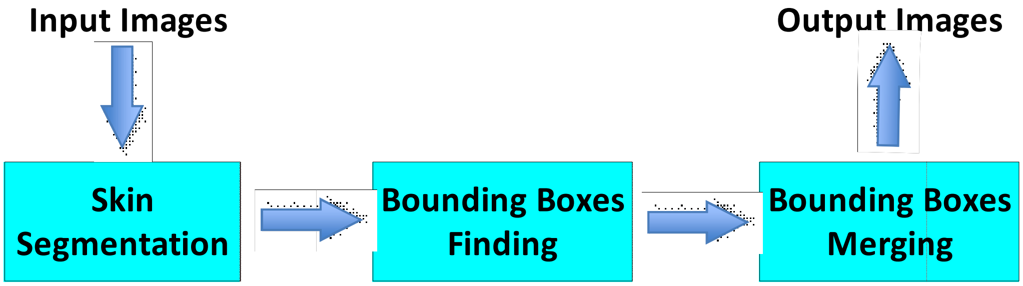


Fig. 6. Flow chart for image pre-processing in Phase III (Courtesy of Ref. [7]).

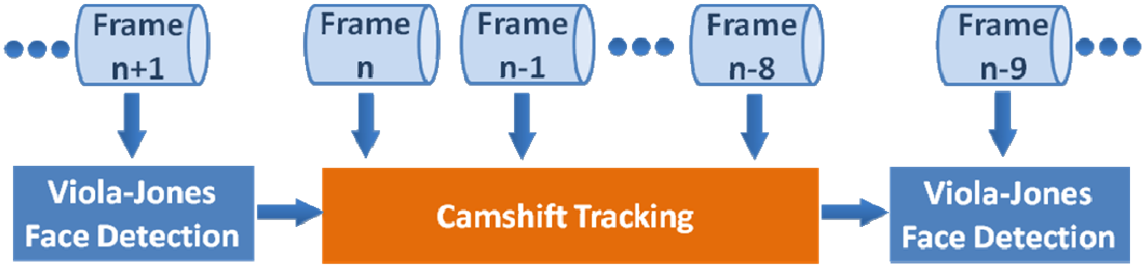


Fig 7. New pipeline combining Viola-Jones Face Detection with Camshift Tracking (Courtesy of Ref. [7]).

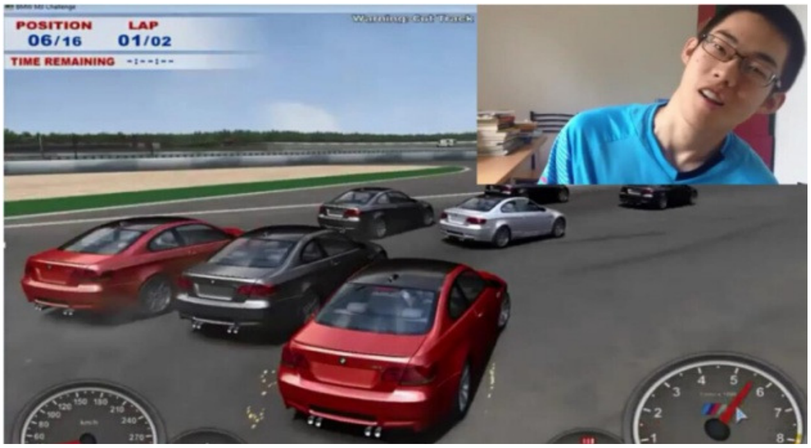


Fig. 8. An innovative application in Phase III: A face-detection game controller (FDGC) (Courtesy of Ref. [7]).

According this new pipeline, the speed breaks through to 32 fps for multi-views face detection and the accuracy is over 95%. In order to show the improvement of Phase III and get real value with new pipeline technic, the team applies it in Face-Detection Game Controller, in which everyone can throw the keyboard, mouse away and play the video game by the face as shown in Figure 8. Four more steps are needed for this extended pipeline:

- Detect & Track Faces, using pipeline to detect and track faces.
- Get the Angle of Faces, using geometrical principle to calculate the angle.
- Socket Communication between Windows & Unix, using socket communication to control signals (i.e. Left/Right).
- Simulate Keyboard Signal, using windows inner function to simulate hardware signal and operate the game.

3. CONCLUSIONS

Intel sponsored 3-phase real-time face detection capstone design projects executed at JI during 2013—2014 in order to strengthen its RealSence technology. The ultimate goal of this work is to catch up the market demands for mobiles and wearables as well as compete with predominant industry giants. Prior to Phase I, the speed of the original version is 3 fps for 480p videos frontal view face detection.

The platform development and system design of this work are based on Intel's HD 4000 graphics and OpenCL. It reaches 9 fps and 81% accuracy after the platform setup and system validation in Phase I. With the techniques of image decomposition and 2D task allocation, the speed and accuracy increase to 30 fps and 94%, respectively, for frontal view detection in Phase II. Combining Viola-Jones algorithm with the Camshift tracking method, the speed reaches 32 fps and the accuracy achieves 95% for real-time multi-view face detection in Phase III. Furthermore, an interesting interactive application called face-detection game controller (FDGC) is developed. The technology has been implemented in wearable devices and mobile with Intel cores at the time of this writing.

ACKNOWLEDGEMENTS

The authors first thank Jackson He, Evelyn Yan, and Michael Fu from Intel for their financial and technical support. Second, the authors thank Haoyang Chen, Guoxing Li, Zichao Fu, Xinyi Guo, Sen Cao, Zhang Chen, Jifan Zhu, Xingmo Liu, Lili Su, Bochun Zhang, Shaoqing Zhu, and Bowen Hong for their efforts in each phase project. Finally, the authors thank ChenyanFeng, Taoxi Li, Jiayi Wu, and Shuo Zhang for the valuable discussion and assistance, Jason Chang for editing the manuscript and Mian Li for helpful advice.

REFERENCES

[1] R. James(2014), Brainstorm Tech conference in Aspen, Colorado. Available: http://fortune.com.

[2] V. Chang (2014), "MNC-Sponsored Multidisciplinary Industrial-Strength Capstone Design Projects in China," International Journal for Innovation Education and Research, Vol.2-10, pp 54-65.

[3] G. Wang, H. Chen, G. Li, Z. Fu, and X. Guo (2013), "OpenCL Accelerated Face Detection (Phase I)," undergraduate thesis, University of Michigan—Shanghai JiaoTong University Joint Institute, summer.

[4] P. Viola, and Michael J. Jones (2004), “Robust Real-Time Face Detection,” International Journal of Computer Vision, Vol. 57, pp 137-154.

[5] X. Pan, S. Cao, Z. Chen, J. Zhu, and X. Liu (2013), “OpenCL Accelerated Face Detection (Phase II),” undergraduate thesis, University of Michigan—Shanghai Jiao Tong University Joint Institute, fall.

[6] C. Fernandez, J. R. Saeta, X. Martorell, and J. Hernando (2011), “Real-time GPU-based Face Detection in HD Video Sequences,” IEEE International Conference on Computer Vision Workshops, November.

[7] G. Wang, L. Su, B. Zhang, S. Zhu, and B. Hong (2014), “OpenCL Based Face Detection and Recognition Algorithm Tuning (Phase III),” undergraduate thesis, University of Michigan—Shanghai Jiao Tong University Joint Institute, summer.

[8] H. Han, S. Shan, X. Chena, and W. Gao (2013), “A comparative study on illumination preprocessing in face recognition,” Pattern Recognition, Vol. 46, pp 1691–1699.

[9] C. Du, H. Zhu, L. Luo, J. Liu, and X. Huang (2013), “Face detection in video based on AdaBoost algorithm and skin model,” The Journal of China Universities of Posts and Telecommunications, Volume 20, Supplement 1, pp 6–9, 24.

AUTHORS

Bohua Gan is a postgraduate student of Shanghai Jiao Tong University majors in Electrical and Communication Engineering. He received B. E in Communication Engineering from Dalian Maritime University in 2013 and LL. B in Law from Dalian Maritime University in 2012. He is interested in all kinds of techniques, such as open source GPGPU, antenna, and telemedicine etc.

He is also a research assistant of Prof. Vincent Chang since September, 2014. He made effort in Intel-sponsored Capstone Design Projects focusing on OpenCL related tasks on Intel GPU. He wishes to make our lives better through the technology.

Vincent Chang, Ph.D. & MBA is Teaching Professor & Faculty Director of Corporate Relations at University of Michigan—Shanghai Jiao Tong University Joint Institute (JI). He is also the founder of Knowledge Master, Inc. (KMI) in the US and the cofounder of Semiconductor Leadership programs at UC Berkeley Extension.

Dr. Chang has been working in academia and industrial positions for more than 20 years in total with experiences in China, US and Taiwan. Dr. Chang pioneered Corporate Partners Programs at JI, covering entrepreneurial innovation and industry research. He strategically initiated and developed 96 MNC-sponsored Capstone Design Projects focusing on healthcare, energy, electronics, and mobile internet, creating an innovative platform collaborative with Covidien, GE, HP, Intel, Philips, and Siemens.

From 2004 to 2010, Dr. Chang was the President and CEO of KMI, a California-based e-learning company specializing in semiconductor IC design and innovation leadership. The company was recognized as an outstanding on-demand technology and content provider by UC Berkeley and awarded 9-year business partner contract from 2007 to 2016. Prior to moving to the US in 2003, Dr. Chang was Associate Professor of Electrical Engineering at Tamkang University in Taiwan. In 1995, he cofounded Calvin Engineering specializing in structural design along with Calvin Chang in Taiwan in 1995. His research has appeared in Electronics Letters, IEEE Journal of Lightwave Technology, IEEE Photonics Technology Letters, and International Journal for Innovation Education and Research. He also authored 14 textbooks in microelectronics published in Taiwan.

Guanying Wang is a software engineer at Google. She received B. E in Electrical and Computer Engineering from Shanghai Jiao Tong University, Shanghai, China and B.E in Computer Software Engineering from University of Michigan, Ann Arbor, in 2013. She received M. E in Information Technology from Carnegie Mellon University in 2014. She is interested in all kinds of novel technologies. She had participated in Mobile Advertisement System Design, Dynamic Keyboard IOS Application, Contextual Design, Inquires and User Interface Prototyping, Intel Innovation Project: OpenCL Accelerated Face Detection and Carnegie Financial Service Web App Development. And she is also a good manager of the project.

Xiuli Pan is a student of Shanghai Jiao Tong University majors in Electrical and Computer Science Engineering. He is interested in machines as well as programs for the logic inside them is attractive.

He is interested in all kinds of techniques such as network, microprocessor, automatic control and programming. With all of the skills I have learned I have made a PRP system based on microprocessor and Bluetooth to detect the barrier and guide the road on smart phones for blind people.

He always dream of making life better by making the techniques more easy to use and more ways to be used. And this idea makes me to join this project and want to make the face detection more easily to use and have a wild use in all areas.

Guan Wang was born in Beijing, China, in1992. He received B.S.E. degree in electrical and computer science engineering from Shanghai Jiao Tong University, Shanghai, China and B.S.E. degree in computer science engineering from University of Michigan, Ann Arbor, in 2014.

His research interests are are mainly in the deep learning, data mining, and information retrieval. His is doing research study with Prof. Honglak Lee. Try to build a system that detecting vehicle as well as road conditions by analyzing real-time video and pipelining multiple detection and tracking algorithms in machine learning and computer vision, such as Convolution Neural Network (CNN), and Discriminative Part Models (DPM). He is also a research assistant of Prof. Michael Cafarella. He made effort in Senbazuru, which is a prototype spreadsheet database management system (SSDBMS), and also helped developing a Feature Engineering Infrastructure that could dramatically ease the Explore-Extract-Evaluate interaction loop that characterizes many trained system projects. As for his future plan, he wishes to continue digging into deep learning and information retrieval area and leveraging those knowledge to make something meaningful.

Nanhai Zou is a senior staff engineer in Intel Asia-Pacific research and developing center. He works for Intel Open source technology center. He is leading a team to work on open source GPGPU and OpenCL related tasks on Intel GPU. Nanhai Zou graduated from Shanghai Jiao Tong university. He holds a master degree of thermal engineering.

Fleming Feng graduated from Tsinghua University at 1990, and then went to Shanghai Jiao Tong University for a master’s degree. He started to work for Intel from year 1997 after 4 year experience in a software company. From 2000, Fleming started to work on Linux and open source project development. Ander from year 2002, Fleming began to work with TSP(the predecessor of OTC), managing the development team for Carrier Grace Linux, and from 2005, took the role for managing the OTC Shanghai team. From 2009, Fleming worked as the PRC chief OSS scientist majorly focusing on Linux and open source collaboration with China government, industry and university.

INTENTIONAL BLANK

ENHANCING PERFORMANCE OF AN HPC CLUSTER BY ADOPTING NON-DEDICATED NODES

Pil Seong Park

Department of Computer Science, University of Suwon, Hwasung, Korea
pspark@suwon.ac.kr

ABSTRACT

Persona-sized HPC clusters are widely used in many small labs, because they are cost-effective and easy to build. Instead of adding costly new nodes to old clusters, we may try to make use of some servers' idle times by including them working independently on the same LAN, especially during the night. However such extension across a firewall raises not only some security problem with NFS but also a load balancing problem caused by heterogeneity. In this paper, we propose a method to solve such problems using only old techniques applicable to old systems as is, without requiring any upgrade for hardware or software. Some experimental results dealing with heterogeneity and load balancing are presented using a two-queue overflow queuing network problem.

KEYWORDS

HPC clusters, Security, NFS, SSH tunnelling, load balancing

1. INTRODUCTION

The desire to get more computing power and higher reliability by orchestrating a number of low cost commercial off-the-shelf computers has given rise to a variety of architectures and configurations. A computer cluster is composed of a set of loosely or tightly coupled computers that are usually connected to each other through a fast LAN and work together so that they can be viewed as a single system in many respects.

Computer clusters may be configured for different purposes. Load-balancing (LB) clusters are configurations in which nodes share computational workload like a web server cluster. High-performance computing (HPC) clusters are used for computation-intensive purposes, rather than handling IO-oriented operations. High-availability (HA) clusters improve the availability of the cluster, by having redundant nodes, which are then used to provide service when system components fail.

Many kinds of commercial clusters are on the market. However the technologies to build similar systems using off-the-shelf components are widely known (e.g., see [1]), and it is easy to build a

David C. Wyld et al. (Eds) : WiMo, ITCSE, ICAIT, NC - 2015
pp. 49–59, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.51005

low-cost personal-sized cluster [2]. Many small labs first build their own personal-sized cluster, and gradually grow it by adding some more dedicated nodes later. Instead, if there are some other nodes that are being used for other purposes on the same LAN, we may try to utilize their idle times as long as they are not always busy enough, especially during the night. The Berkeley NOW (Network of Workstations) project is one of early attempts to make use of the power of clustered machines on a building-wide scale [3]. However such extension gives rise to difficulties in security, administering the cluster, and load forecasting for optimal performance.

In this paper, we deal with some technical issues in extending a personal-sized Linux cluster across a LAN. We do not use state-of-the-art technologies, since the sole purpose is to achieve our purpose using our old HPC clusters as is, with no hardware or software upgrade. Some results of the experiments to evaluate the system are given at the end.

2. BACKGROUND

2.1. HPC Cluster Middlewares

An HPC cluster normally consists of the dedicated nodes that reside on a secure isolated private network behind a firewall. Users normally access the master/login node (master, for short) only, which sits in front of compute nodes (slave nodes, or slaves for short).

The activities of all compute nodes are orchestrated by a cluster middleware that allows treating the whole cluster system via a single system image concept. Well-known HPC middlewares based on message passing are the Message Passing Interface (MPI) [4] and the Parallel Virtual Machine (PVM) [5], the former being the de facto standard. According to the standard, many commercial or non-commercial implementation libraries have been developed. LAM/MPI, FT-MPI, and LA-MPI are some of widely used non-commercial libraries, and their technologies and resources have been combined into the on-going Open MPI project [6].

All the nodes in an HPC cluster share not only executable codes but also data via NFS (Network File System), which is not encrypted in general. It is perfectly all right as long as the whole cluster nodes are on a secure local area network behind a firewall.

To extend the cluster to include other nodes outside of the firewall, we will be confronted with some problems with security and data sharing among nodes. Moreover, such growth results in a heterogeneous cluster with nodes of different power, possibly running different Linux versions.

2.2. File Sharing Protocols

NFS, created by Sun Microsystems in 1984, is a protocol to allow file sharing between UNIX systems residing on a LAN. Linux NFS clients support three versions of the NFS protocol: NFSv2 (1989), NFSv3 (1995), and NFSv4 (2000). However the NFS protocol as is has many problems to use in extending the cluster, since its packets are not encrypted and due to other shortcomings to be discussed later.

Other alternatives to NFS include AFS (Andrew File System), DFS (Distributed File System), RFS (Remote File System), Netware, etc. [7]. There are also various clustered file systems shared by multiple servers [8]. However we do not adopt such new technologies since they are not supported by our old cluster.

2.3. SSH Tunnelling

Encryption of NFS traffic is necessary for secure extension across a firewall. One of the common techniques that are ordinarily used is known as cryptographically protected tunnelling. An IP-level or TCP-level stream of packets is used to tunnel application-layer segments [9]. A TCP tunnel is a technology that aggregates and transfers packets between two hosts as a single TCP connection. By using a TCP tunnel, several protocols can be transparently transmitted through a firewall. Under certain conditions, it is known that the use of a TCP tunnel severely degrades the end-to-end TCP performance, which is called TCP meltdown problem [10].

The SSH protocol allows any client and server programs to communicate securely over an insecure network. Furthermore, it allows tunnelling (port forwarding) of any TCP connection on top of SSH, so as to cryptographically protect any application that uses clear-text protocols.

3. EXTENSION OF AN HPC CLUSTER

We would like to extend our old cluster to include some other nodes across the firewall, as shown in Figure 1. In general, the master node has attached storage that is accessible by diskless slave nodes using insecure NFS. Since NFS relies on the inherently insecure unencrypted UDP protocol (up to NFSv3), IP spoofing is possible. Moreover, firewall configuration is difficult because of the way NFS daemons work.

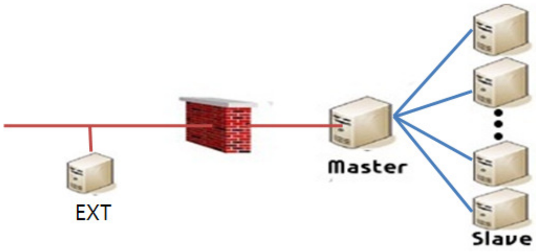


Figure 1. Cluster extension to include the non-dedicated node EXT across a firewall.

3.1. Fixing NFS Ports for SSH Tunnelling

SSH tunnelling, which makes use of SSH port forwarding, is widely used to encrypt some unencrypted packets or to bypass firewalls, e.g., see [9,11]. SSH tunnels support only TCP protocols of fixed ports, but NFS uses UDP protocols by default, and the ports of some daemons essential for the operation of NFS are variable.

Fortunately NFS over TCP protocols are also supported from the Linux kernel 2.4 and later on the NFS client side, and from the kernel 2.4.19 on the server side [12]. Since all the nodes of our cluster satisfy this condition, we can make NFS work over TCP by specifying the option "-o tcp" in the mounting command. The following is an example of mounting *server*'s */nfs_dir* directory on the client's *mount_pt*.

```
# mount  -t nfs  -o tcp  server:/nfs_dir   mount_pt
```

The daemons essential for NFS operation are 2 kinds. Portmapper (port 111) and rpc.nfsd (port 2049) use fixed ports, but rpc.statd, rpc.lockd, rpc.mountd, and rpc.rquotad use ports that are randomly assigned by the operating system. However the ports of the latter can be fixed by specifying port numbers in appropriate configuration files [13].

The ports of the first three can be fixed by adding the following lines in the NFS configuration file /etc/sysconfig/nfs, for example,

```
STATD_PORT=4001
LOCKD_TCPPORT=4002
LOCKD_UDPPORT=4002
MOUNTD_PORT=4003
```

The rest ports can be fixed by defining new port numbers in /etc/services, for example

```
rquotad 4004/tcp
rquotad 4004/u에
```

3.2. Setting up an SSH Tunnel

For tunnelling of NFS, it is necessary for the server to mount its own NFS directory to be exported to clients. Hence on the server side, the configuration file /etc/exports has to be modified, for example, for exporting */nfs_dir*,

```
/nfs_dir    localhost (sync,rw,insecure,root_squash)
```

where “insecure” means it allows connection from ports higher than 1023, and “root_squash” means it squashes the root permissions for the client and denies root access to access/create files on the NFS server for security.

On the client side, to forward the ports 10000 and 20000, for example, to the fixed ports 2049 (rpc.nfsd) and 4003 (rpc.mountd), respectively, we can use the command

```
# ssh nfssvr -L 10000:localhost:2049  -L 20000:localhost:4003 -f sleep 600m
```

where "*nfssvr*" is the IP or the name of the NFS server registered in /etc/hosts, and "-f sleep 600m" means that port forwarding is to last for 600 minutes in the background.

Once connected to the NFS server, an SSH tunnel will be open if the correct password is entered. Then the NFS server’s export directory can be mounted on the client’s side. The following command is an example to mount the */nfs_dir* directory of the NFS server on the */client_dir* directory of the client.

```
# mount -t nfs -o tcp,hard,intr,port=10000, mountport=20000 localhost:/client_dir
```

4. Dealing With Heterogeneity

Data partitioning and load balancing are important components in parallel computation. Since earlier works (e.g., see [14]), many authors have studied load balancing using different strategies on dedicated/non-dedicated heterogeneous systems [15,16,17,18], but it is nearly impossible to find works on the security problems arising in cluster expansion, which is more technical rather than academic.

Even though the original tightly-coupled HPC cluster may be homogeneous, the extended cluster inevitably will behave like a heterogeneous system. First, the power of the newly added node EXT may be different and its workload may change continually since it is not a dedicate node. In addition, the communication speed between the node EXT and the original cluster is slower than that among the original cluster nodes. Hence we need to use a dynamic run-time load balancing strategy [19], while assigning equal amount of work to the nodes of the original cluster.

4.1. The Sample Problem

Strategies of dynamic load balancing depend on problems, and we need introduce the problem we consider first. We deal with the old famous two-queue overflow queuing problem given by the Kolmogorov balance equation

$$\begin{aligned}&[\lambda_1\left(1-\delta_{i,n_1-1}\delta_{j,n_2-1}\right)+\lambda_2\left(1-\delta_{j,n_2-1}\right)+\mu_1\min(i,s_1)+\mu_2\min(j,s_2)]\,p_{i,j}\\&=\lambda_1\left(1-\delta_{i,0}\right)p_{i-1,j}+\mu_1\left(1-\delta_{i,n_1-1}\right)\min(i+1,s_1)\,p_{i+1,j},\quad i,\ j=1,2,\end{aligned}$$

where $p_{i,j}$ is the steady-state probability distribution giving the probability that there are i_j customers in the j-th queue. It is assumed that, in the i-th queue, there are s_i parallel servers and n_i-s_i-1 waiting spaces, and customers enter the i-th queue with mean arrival rate λ_i, and depart at the mean rate μ_i. The model allows overflow from the first queue to the second queue if the first queue is full. The total number of possible states is $n_1 \times n_2$.

It is known that no analytic solution exists, and the balance equation has to be solved explicitly. We can write the discrete equation as a singular linear system, say $A\mathbf{x} = 0$ where $\mathbf{x}$ is the vector consisting of all states $p_{i,j}$'s. Even for systems with relatively small numbers of queues, waiting spaces, and servers per queue, the size of the resulting matrix is huge. The numbers of waiting spaces in our problem are 200 and 100 in each queue, respectively, and the resulting matrix size is 20,000 x 20,000.

4.2. Job Assignment by Domain Decomposition

The graph of the resulting matrix of a k-queue problem is the same as the graph of the matrix corresponding to the k-dimensional Laplacian operator. Thus the graph of the matrix A is a two-dimensional rectangular grid.

As an example, Figure 1.a) shows the grid of a two-queue overflow model with 9 x 5 = 45 states, where each dot corresponds to some state $p_{i,j}$, i=1,2,…9 and j=1,2….5. If we use 3 computation nodes, we may assign 15 grid points to each node. However to compute the values at boundary points (in vertical dotted boxes), each node needs the values at the boundary points computed by adjacent nodes. As a result, node 0 and node 2 should have enough memory to store 20 grid points (or 4 vertical grid lines) and node 2 needs to store 25 grid points (or 5 vertical grid lines), even though each node updates only 15 grid points (or 3 vertical grid lines). The extra grid points are to store the values to be received from adjacent nodes.

Figure 1.b) shows partitioning of the corresponding matrix A of size 45×45, each submatrix A_i being of size 5×45. For better load balancing, workload will be assigned in units of one vertical line (that corresponds to the computation with one submatrix A_i). Since we deal with a huge matrix problem of size $20{,}000 \times 20{,}000$, one unit workload is computation with $100 \times 20{,}000$ submatrix.

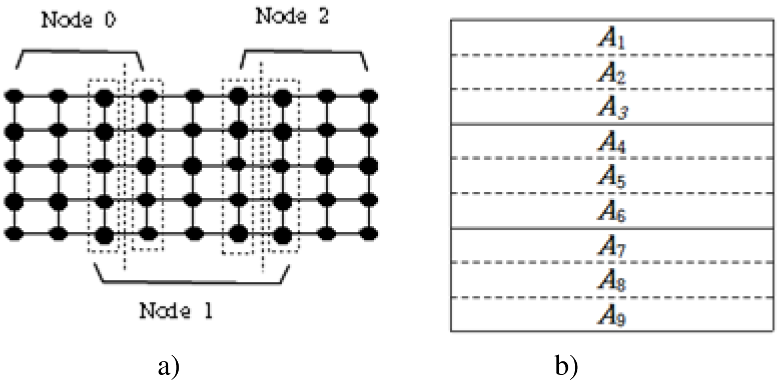


Figure 2. Examples showing a) job assignment to 3 nodes, and b) matrix partitioning for a 9 x 5 = 45 state two-queue overflow queuing model.

To solve the singular system $A\mathbf{x} = 0$, we use a splitting method $A=D-C$ where D is the diagonal matrix that consists of all diagonal entries of A, and $C=D-A$. Then we can rewrite $D\mathbf{x} = C\mathbf{x}$ which leads to a convergent eigenvalue problem $D^{-1}C\mathbf{x} = \mathbf{x}$. For convergence test, we use the size of the residual defined by

$$\| \mathrm{r} \|_2 = \| (I - D^{-1}C)\mathbf{x} \|_2 \qquad \text{where } \| \mathbf{x} \|_2 = 1.$$

4.3. Communication for Asynchronous Iteration

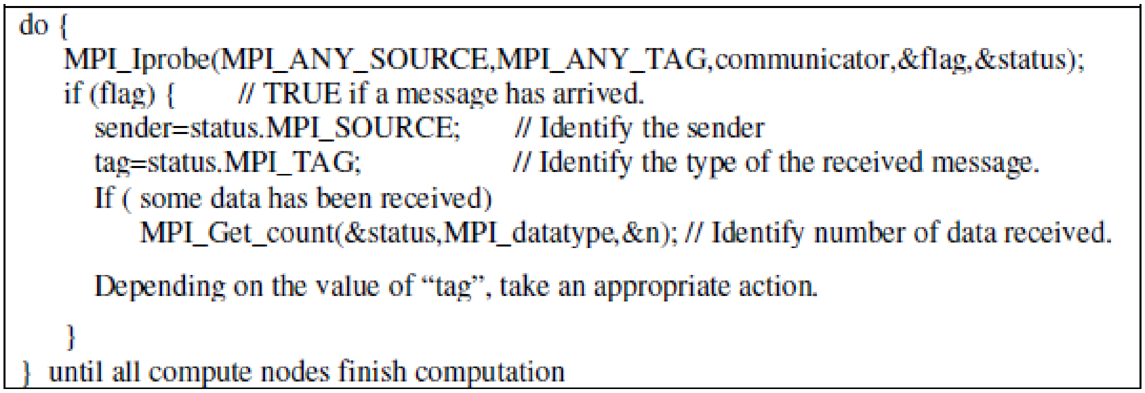
In normal synchronous parallel computation, each node updates the values of the grid points assigned to it in lockstep fashion, so that it can give the same solution as that of serial computation. During computation, each node should communicate with its adjacent nodes to exchange boundary point values, and all nodes should synchronize computation.

The workload of the added node EXT may change dynamically all the time, and it is unpredictable and uncontrollable. If the node EXT is busy doing some other jobs, it cannot communicate with its adjacent nodes in time, degrading overall performance by making all other nodes wait.

Asynchronous algorithms appear naturally in parallel systems and are heavily exploited applications.Allowing nodes to operate in an asynchronous manner simplifies the implementation of parallel algorithms and eliminates the overhead associated with synchronization. Since 1990's, some theories on asynchronous iteration has been developed, e.g., see [20]. Fortunately, the matrix A in our problem satisfies the necessary condition for a matrix to satisfy for convergence. Hence we adopt asynchronous iteration freely, but we need some more work for implementation details.

We assume the master node is always alive and is not busy so that it can immediately handle communication with other nodes. Data exchange between two adjacent nodes should be done through the master node, and the master node should keep the most current values of all grid points. But all communication should be initiated by compute nodes (including the added non-dedicate nodes), and the master node should just respond according to compute nodes' request.

Figure 3 is the master's algorithm using MPI functions to handle other nodes' request. The master continually checks if any message has arrived from any other nodes by calling the MPI_Iprobe( ) function. If there is any message arrived, it then identifies the sender, tag, and sometimes the number of data received. Based on the tag received, the master takes some appropriate action. For example, if the received tag is 1000, it means "Receive the boundary data I send, and send the boundary data I need.", etc. Then any compute node can continue its work without delay.



```
do {
    MPI_Iprobe(MPI_ANY_SOURCE,MPI_ANY_TAG,communicator,&flag,&status);
    if (flag) {       // TRUE if a message has arrived.
        sender=status.MPI_SOURCE;       // Identify the sender
        tag=status.MPI_TAG;             // Identify the type of the received message.
        If ( some data has been received)
            MPI_Get_count(&status,MPI_datatype,&n); // Identify number of data received.

        Depending on the value of "tag", take an appropriate action.

    }
} until all compute nodes finish computation
```

Figure 3. A master's algorithm to handle compute nodes' request.

We also adopt a dynamic load balancing strategy by measuring the time interval between two consecutive data requests by the same node. It is necessary only for non-dedicated nodes, since the others are dedicated and have the same computational power. The workload of the non-dedicated node is reduced if

$$n < \alpha \frac{N t}{T}$$

where N is the average workload (in units of the number of vertical grid lines), T is the average time interval between two consecutive data requests by the same dedicated node, t is the time interval between two requests of EXT, and α is some tolerance parameter less than 1 (we used 0.8). Similarly, the workload of the non-dedicate node is increased if

$$n > \beta \frac{N t}{T}$$

for some β (we used 1.2).

In adjusting workloads, we used a simpler strategy. That is, when one of these conditions is satisfied, we just decrease/increase two boundary grid lines of EXT and assigned them to/from adjacent nodes. For this, to each non-dedicated node, we assigned the grid region which is between the two regions assigned to dedicated nodes. It takes very little time to re-assign grid lines to some other node, since our matrix is very sparse (just 5 nonzero entries in each row) and can be generated by each node instantly whenever necessary.

5. PERFORMANCE TESTS

Our old cluster we want to extend has 2 nodes each with dual 1GHz Pentium 3 Xeon processors running Fedora Core 4, with LAM/MPI v.7.1.2 installed. The nodes are interconnected via a Gigabit LAN, and NFS is used for file sharing. The non-dedicated node EXT which sits on the outside fast Ethernet has a 2.4 GHz Intel® Core™ i5 CPU with 8 GB memory, and runs on Fedora 13.

The performance of the NFS through an SSH tunnel will inevitably drop due to encryption overhead. Tests were performed between the master node and the node EXT, using UDP or TCP, and with or without an SSH tunnel across the firewall. The times it took to read or write a file of size 1GB from EXT were measured, at varying NFS block sizes. Since NFSSVC_MAXBLKSIZE (maximum block size) of the NFS in Fedora Core 4 is 32*1024 (see /usr/src/kernels/2.6.11-1.1369_FC4-i686/include/linux/nfsd/const.h), tests were performed at block sizes of 4k, 8k, 16k, and 32k, respectively, 3 times for each and they are averaged. In addition, to delete any remaining data in cache, NFS file system was manually unmounted and mounted again between tests.

The following are example commands that first measure the time taken to create the file /home/testfile of size 1GB on the NFS server and read it using block size 16KB.

```
# time dd if=/dev/zero of=/home/testfile bs=16k count=65536
# time dd if=/home/testfile of=/dev/null bs=16k
```

The results of the NFS performance test, with or without SSH tunnelling are given in Table 1. For the common NFS without tunnelling, the figures in parentheses are the times it took when TCP is used, and others are when UDP is used.

As we see, the larger the NFS block size, the faster in all cases. For the common NFS without SSH tunnelling, write operation using UDP is slightly faster than TCP, but it is not the case for read operation. Moreover the NFS with SSH tunnelling takes 3.9%-10.1% more time for write and 1.4-4.3% more for read, than the common NFS using UDP.

As long as the NFS block size is taken as large as possible, the tunnelling overhead may not be large even though NFS service is done through SSH tunnelling, since the non-dedicated nodes outside of the firewall need not read or write so often through NFS, which is common in high performance parallel computing.

Table 1. Performance of NFS, with or without SSH tunnelling (sec)

Block size	Common NFS		SSH tunnelled	
	Read	Write	Read	Write
4K	96.12 (95.31)	119.37 (120.95)	100.29	131.45
8K	95.07 (94.43)	115.74 (120.31)	98.12	122.13
16K	93.85 (92.21)	114.70 (118.32)	95.31	121.72
32K	92.51 (91.45)	112.35 (115.87)	93.89	116.83

Figure 4 shows the comparison results. The solid line marked with 1 is the synchronous computation result using only the dedicated nodes of our old cluster which has 4 cores in total. The dotted line marked with 2 is the result when we add the node EXT (we created just 1 process on EXT). The residual and elapsed time were checked every time the first slave reports the intermediate result. The running times to converge to the result with residual size less than 10^{-5} were 956.75 sec for curve 1 and 854.14 sec for curve 2, respectively. Hence the speedup by the added EXT node is 1.12.

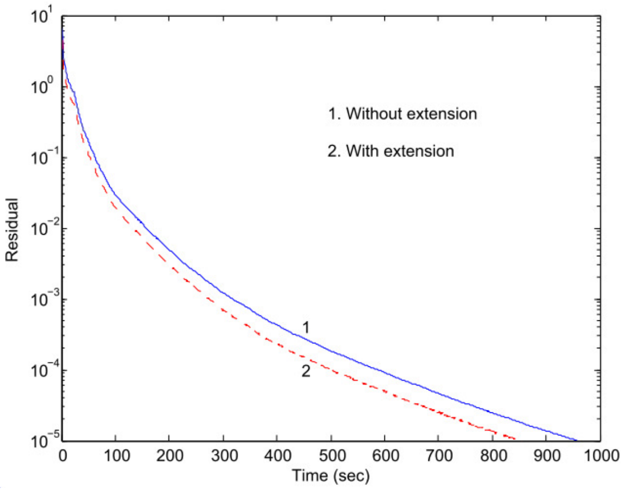


Figure 4. Comparison results.

The result is somewhat disappointing, since the speed of the CPU core on EXT is at least twice as fast as that of the old cluster. Probably it is mainly due to much slower network speed between the cluster and the node EXT and the extra workload for some other jobs.

6. CONCLUSIONS

Personal-sized HPC clusters are widely used in many small labs, because they are easy to build, cost-effective, and easy to grow. Instead of adding costly new nodes, we can extend clusters to include some other servers on the same LAN, so that we can make use of their idle times.

However, unlike a tightly-coupled HPC cluster behind a firewall, the resulting system suffers a security problem with NFS which is vital for HPC clusters.

Of course there are many new good solutions using recent technologies. However we do adopt such strategy, because they require upgrade of hardwares and/or softwares including an operating system. Instead we devise a solution using SSH tunnelling, which can be applied to the old system as is. Probably this approach may be helpful to many of small home-made cluster systems. We also devised a dynamic load balancing method which is based on domain decomposition and asynchronous iteration, using a two-queue overflow queuing network problem of matrix size $20{,}000 \times 20{,}000$. The speedup obtained by using one extra process on a much faster non-dedicated node is somewhat disappointing, mainly because of the slow fast Ethernet speed between the cluster and the extra node. We expect much higher speedup if the outer network is upgraded.

ACKNOWLEDGEMENTS

This work was supported by the GRRC program of Gyeonggi province [GRRC SUWON2014-B1, Cooperative Recognition and Response System based on Tension Sensing].

REFERENCES

[1] G. W. Burris, "Setting Up an HPC Cluster", ADMIN Magazine, 2015, http://www.admin-magazine.com/HPC/Articles/real_world_hpc_setting_up_an_hpc_cluster

[2] M. Baker and R. Buyya, "Cluster Computing: the commodity supercomputer", Software-Practice and Experience, vol.29(6), 1999, pp.551-576.

[3] Berkeley NOW Project, http://now.cs.berkeley.edu/

[4] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, and J. J. Dongarra, MPI: The Complete Reference. MIT Press, Cambridge, MA, USA, 1996.

[5] G. A. Geist, A. Beguelin, J. J. Dongarra, W. Jiang, R. Manchek, and V. S. Sunderam, PVM: Parallel Virtual Machine: A Users' Guide and Tutorial for Networked Parallel Computing. MIT Press, Cambridge, MA, USA 1994.

[6] Open MPI, http://www.open-mpi.org

[7] Linux NFS-HOWTO, http://nfs.sourceforge.net/nfs-howto/

[8] Clustered file system, http://en.wikipedia.org/wiki/Clustered_file_system

[9] M. Dusi, F. Gringoli, and L. Salgarelli, "A Preliminary Look at the Privacy of SSH Tunnels", in Proc. 17th International Conference on Computer Communications and Networks(ICCCN '08), 2008.

[10] O. Honda, H. Ohsaki, M. Imase, M. Ishizuka, and J. Murayama, "Understanding TCP over TCP: Effects of TCP tunneling on end-to-end throughput and latency," in Proc. 2005 OpticsEast/ITCom, Oct. 2005.

[11] Port Forwarding Using SSH Tunnel, http://www.fclose.com/b/linux/818/port-forwarding-using-ssh-tunnel/

[12] Linux NFS Overview, FAQ and HOWTO, http://nfs.sourceforge.net/

[13] http://www.linuxquestions.org/questions/linux -security-4/firewall-blocking-nfs-even-though-ports-are-open-294069/

[14] M. Zaki, W. Li, and S. Parthasarathy, "Customized Dynamic Load Balancing in a Heterogeneous Network of Workstations", in 1996 Proc. 5th IEEE Int. Symposium on High Performance Distributed Computing.

[15] M. Eggen, N. Franklin, and R. Eggen, "Load Balancing on a Non-dedicated Heterogeneous Network of Workstations." in International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2002), June 2002.

[16] J. Faik, L. G. Gervasio, J. E. Flaherty, J. Chang, J. D. Teresco, E.G. Boman, and K. D. Devine, "A model for resource-aware load balancing on heterogeneous clusters", Tech. Rep. CS-03-03, Williams College Dept. of Computer Science, http://www.cs.williams.edu/drum/, 2003.

[17] I. Galindo, F. Almeida, and J. M. Badia-Contelles, "Dynamic Load Balancing on Dedicated Heterogeneous Systems", Recent Advances in Parallel Virtual Machine and Message Passing Interface, Lecture Notes in Computer Science, vol.5205, 2008, pp 64-74

[18] K. Lu, R. Subrata, and A. Y. Zomaya, "On the performance-driven load distribution for heterogeneous computational grids", J. of Computer and System Sciences vol.73, 2007, pp.1191-1206.

[19] L. V. Kale, M. Bhandarkar, and R. Brunner, "Run-time Support for Adaptive Load Balancing", in Lecture Notes in Computer Science, Proc. 4th Workshop on Runtime Systems for Parallel Programming (RTSPP) Cancun - Mexico, vol.1800, 2000, pp.1152-1159.

[20] M. P. Raju and S. Khaitan, "Domain Decomposition Based High Performance Computing", International J. of Computer Science Issues, vol.5, 2009,pp.27-32.

AUTHOR

Prof. Pil Seong Park received his B.S. degree in Physical Oceanography from Seoul National University, Korea in 1977, and M.S. degree in Applied Mathematics from Old Dominion University, U.S.A. in 1984. He received his Ph.D. degree in Interdisciplinary Applied Mathematics (with emphasis on computer science) from University of Maryland at College Park, U.S.A in 1991. He worked as the head of Computer Center at Korea Ocean Research & Development Institute from 1991 to 1995. Currently, he is a professor of the Department of Computer Science, University of Suwon in Korea. His research interest includes high performance computing, Linux clusters, digital image processing, and knowledge-based information systems. He is a member of several academic societies in Korea.

INTENTIONAL BLANK

ADAPTIVE AUTHENTICATION: A CASE STUDY FOR UNIFIED AUTHENTICATION PLATFORM

Khairul Azmi Abu Bakar[1] , Nor Izyani Daud[2]
and Mohd Shafeq Md Hasan[3]

[1]Information Security Lab, MIMOS Berhad, Malaysia
khairul.azmi@gmail.com
[2]Information Security Lab, MIMOS Berhad, Malaysia
izyani.daud@mimos.my
[3]Information Security Lab, MIMOS Berhad, Malaysia
shafeq.hasan@mimos.my

ABSTRACT

Adaptive authentication is a risk-based authentication that identifies high-risk and suspicious illegitimate login attempts. User past login records which implicitly contains attribute factors context information are used to establish user behavior profile. Later if the user logins under different environmental context from that established profile, the identity of the user may be questioned. The system may challenge the user to present additional authentication method to get authenticated. We implemented such adaptive authentication system in our production server and collected user login records for more than six months. In this paper, we presents the analysis of the user login profile with regards to attribute factors such as geographical location and time of login. We also developed testbed system that uses the collected real data to evaluate the system for different ratio threshold values.

KEYWORDS

Adaptive Authentication, Web Application, Testbed Analysis

1. INTRODUCTION

Authentication is a process to confirm that someone or something is, in fact, who or what it is claimed to be. The process involves obtaining identification credentials or authentication method such as username/password from a user and validating the credential against some authority. Users who can present valid credential are considered authenticated identities. In general, there are three categories of credentials: something you know (password), something you have (ATM card) or what you are (fingerprint). To make it difficult for unauthorized person to gain access, the system may implement multi-factor authentication where the user needs to successfully present additional credentials from at least two of those three categories. If one factor is compromised or broken, the attacker still has at least one more barrier to breach to break into the system.

David C. Wyld et al. (Eds) : WiMo, ITCSE, ICAIT, NC - 2015
pp. 61–72, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.51006

In traditional authentication system, the decision on the level of authentication credential required solely depends on the application that the user trying to access. High sensitive applications such as internet banking would demand the user to present stronger authentication credential than what insensitive applications would. The required authentication methods could also be a combination of two or more credentials, increasing the authentication security even more. However, in such system, the security requirement is static because it only depends on the application security requirement.

Adaptive authentication system uses login environmental characteristics and user behavioral profiling to identify high-risk login and dynamically customizes the authentication requirement accordingly. The system studies common behavior pattern of all users based on their past history login access. If a user follows the same patterns when logging into the system, the login experience may probably be a username/password indicating a low risk attempt. However, if a user tries to login under different behavior or environment, the identity of the user is questioned. The system may adaptively challenge the user to provide stronger or additional authentication credentials to get authenticated.

We built such system in our lab and put it on our production server for couple of months. The system had been collecting and storing users' login information into a database. We also developed a testbed system that uses those data as the input parameters. The testbed enables us to evaluate the behavior of our adaptive authentication system with different set of configurations are used. In this paper, we present the result.

The remainder of the paper is organized as follows. Section 2 presents our current Unified Authentication Platform (UAP) system that contains adaptive authentication component. Section 3 describes the formula to calculate User Attribute Score with regards to the relevant attribute factors. Section 4 explains the algorithm used for the testbed environment that have been developed to evaluate the system. Section 5 present the results from user login records analysis and different ratio threshold values. Finally, Section 6 draws the conclusions.

2. BACKGROUND

In MIMOS Berhad, we have developed an authentication system called Unified Authentication Platform (UAP). UAP is a centralized multi-factor authentication system with web-based single sign-on (SSO) capability to manage user authentication profiles. It is designed to manage front-end application authentication using an established protocol, Secure Assertion Markup Language (SAML), which provides a centralized authentication framework and aims to reduce significant application changes at the backend. The objectives of UAP are as the following:

1. provide an infrastructure that offers authentication service to applications
2. provide information technology that de-couples authentication function from application and authorization
3. grow indigenous authentication mechanism industry throughout the country
4. a unified authentication platform initiative for enabling government e-services application

UAP is derived from Shibboleth [7] which is a standard based, open source package for web single sign-on across or within organizational boundaries. In addition, UAP supports multiple authentication methods. Users can choose from a list of authentication methods to get authenticated and be allowed to access various applications without having to go through the same authentication process again. The overall architecture of UAP is depicted in Figure 1[5].

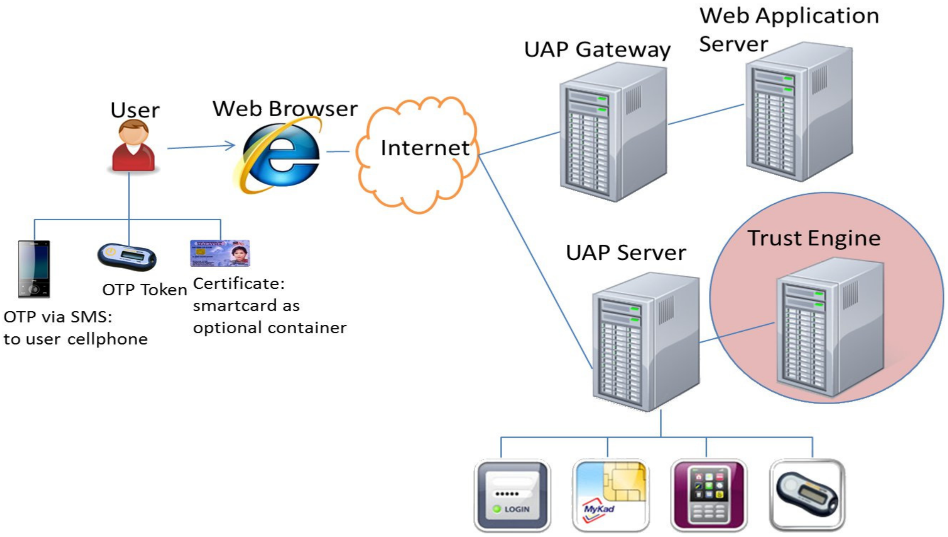


Figure 1. Adaptive UAP Architecture Diagram

For the new generation of UAP, called adaptive UAP, we introduce an additional component called Trust Engine that incorporates adaptive control based on security risk and level of assurance. To make informed authentication decision, Trust Engine takes into account attribute factors from the user behavior profiles which had been previously analyzed and stored.

Adaptive UAP consists of two processes: Pattern Generator and Trust Evaluator. Pattern Generator analyses context information from user past login records which had been stored at table data_log. Only login records from the last predefined period of time are processed. Context information from the records such as login time and IP Addresses are converted into a meaningful data format before the results along with the number of occurrence are stored at table common_attr as the user behavior profile. Pattern Generator is currently configured to get executed on every midnight.

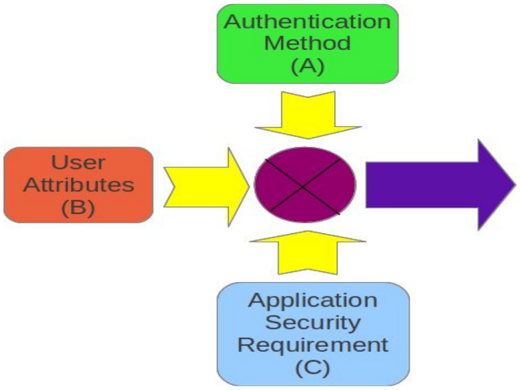

Authentication Method (A)
User Attributes (B)
Application Security Requirement (C)

Figure 2. Decision Factors

Trust Evaluator is responsible to analyze, decide and act upon every login request from users. Trust Evaluator makes the final decision regarding on user authentication based on three factors as shown in Figure 2: Authentication Method Score, User Attribute Score and Application Required Trust Level. Authentication Method Score derives from the accumulated trust value of all user successfully presented credentials. User Attribute Score is the result value from the comparison process between the user current login contexts against the established user behavior profile. Finally, Application Required Trust Level is the minimum trust level set by the application that the user need to acquire to get authenticated. At the end of the process, Trust Evaluator stores the login records into table data_log as depicted in Table 1 next to be processed by component Pattern Generator.

Table 1. User Login Information.

No	Entry	Description
1.	uuid	User unique identity
2.	time_login	Data and time of the login
3.	browser_osname	User's browser and OS information
4.	ip_int	User's terminal IP Address
5.	sp_id	UAP Gateway Server ID string
6.	auth_method	Authentication method ID
7.	Tr	Application required trust level

Basically the requirement as in Formula 1 should be satisfied before the user is considered an authenticated entity by the system.

$$A - B \geq C \qquad (1)$$

Where
A: Authentication Method Strength
B: User Attribute Score
C: Application Security Requirement

If the established trust level does not meet the application required trust level, Trust Engine returns FALSE to Authentication Server. User needs to present additional authentication method to increase the established trust level. More explanation with some scenario examples can be read at [3] and [4].

3. USER ATTRIBUTE SCORE

In this section, the second component of the formula that contributes for the final authentication decision is discussed. User Attribute Score represents the uncertainty level of current login attempt with respect to the established user behavior profile. In other words, if the user logins under different behavior and environment from what the user normally had experienced, the score value would be high. In this case study, attribute factors used are user geographical location (city name), time login, type of browser and operating and targeted application.

There are two steps to calculate User Attribute Score. In the first step, the system needs to find out the user common context for each attribute factor. Common context should meet the following conditions:

1. the number of records for the last 14 days is more than 10
2. the frequency of occurrence of any particular context is more than a ratio threshold of the overall records

Trust Evaluator should have gathered reasonably enough number of user records to establish credible user attribute profile. In this case study, the minimum number of records required is set to 10 from which only login records for the past 14 days are been considered .

To get qualified as a common context, the occurrence of the context should also exceeds a ratio threshold. Presumably the threshold is set to 30%, if the user has 100 login records within the period of time, any context that has at least 30 records is labeled as the user common context. For example, if the user had been login from city of Kuala Lumpur for more than 30 times, Kuala Lumpur is considered as the user common context under geolocation factor. If none of the geolocation information meets both of the two conditions, the user is considered as not having common context for that attribute factor and geolocation element is omitted from the formula. Other attribute factors may still be applied subject to the same conditions.

If there exist common profile within the attribute factor, Trust Evaluator compares each of the common profiles with the user current login context. If there is a match, the value of that attribute factor is set to zero which gives no effect to the final score. Otherwise, the attribute factor is activated. As explained later in this paper, the number of event where the attribute factor is activated is recorded for analysis. The number reflects the rate of occurrence where the user logins under different environment from the user behavior profile for every attribute factor.

Each of the attribute factor is assigned with a weightage value. The weightage value is in fraction numbering format and represents the significance of the factor in user behavior profile. Higher significant factor is assigned with a higher value. User attribute factor is calculated by adding the weightage values for all activated factors as shown in (2) and multiply the result with a variable max_user_score to limit the maximum possible scoring number.

$$\begin{aligned}\text{attribute\_score} = (&(\,\text{time} * \text{weight}_{\text{time}}) \\ &+ (\text{geolocation} * \text{weight}_{\text{location}}) \\ &+ (\text{browserOS} * \text{weight}_{\text{browser}}) \\ &+ (\text{application} * \text{weight}_{\text{application}})) \\ * (\text{max\_user\_score}) &\end{aligned} \tag{2}$$

4. TESTBED SETUP

We have developed a testbed component for the Trust Engine that would able to take input parameter from the stored records in table data_log instead of from the UAP server. Both processes in Trust Engine take place as it is a real input data. The testbed allows us to evaluate the performance of the Trust Engine when different set of configuration is used. Algorithm 1 shows the pseudo code for the testbed. Trust Evaluator component processes each of the login record while Pattern Generator component is executed at the end of each day.

Algorithm 1. General Steps for TestBed

```
Generate list of days of table data_log
for Each day in the list do
    for Every login records on day day do
        process TrustEvaluator
    end for
    process PatternGenerator
end for
```

5. EXPERIMENT RESULT

We collected login information from our production server from 6 May 2014 until 15 January 2015. Total number of 171,045 login information from 1244 unique users have been recorded during those 254 days period.

5.1. Attribute Factor

In this section, we present the overall analysis of the collected user login information based on the attribute factor.

5.1.1. Geographical Location

The information about user geographical location is extracted from the IP Address of the user terminal. We use solution from a third party company ip2location [6] which provides database records that contain geographical information such as name of the city, region and country of origin for IP Addresses. Special IP Addresses (10.0.0.0 to 10.255.255.255, 172.16.0.0 to 172.31.255.255, or 192.168.0 0 to 192.168.255.255) are originated from internal network and are categorized as private IP Addresses.

From the total number of records collected, 67.7% (115,816) have IP Address information. 97.8% (113,254) of those login records that contains IP Address information are from internal network. From the other 2,562 login records that comes from external network, 2,515 (98.1%) are from Malaysia where city of Kuala Lumpur is the most originating login access location with 2,116 records (84.1%). The remaining login records are from USA (37), Sweden (3), Thailand (3), Netherland (2) and Philippines (2).

5.1.2. Time Login

Time login entries stored in table data_log are the time of the server machine when the login requests were received from UAP server. In other words, if the users login at a location that has a different time zone, the server only stores the time of the server into the database. Time period is divided into three time blocks, each with a different ID based on a standard working hours as shown in Table 2.

The login records show that 95.4% of the login requests were received during the configured working hours which is the second time block in the table. 2.9% of the total records are from the first time block (7 pm- 12 am). Only 1.7% of the login access were recorded during the third time block (12 am - 8 am). Figure 3 shows the number of login events based on the time hours.

Table 2. Time Block Duration

No	Block ID	Duration
1	A	12am – 8am
2	B	8am - 7pm
3	C	7pm – 12am

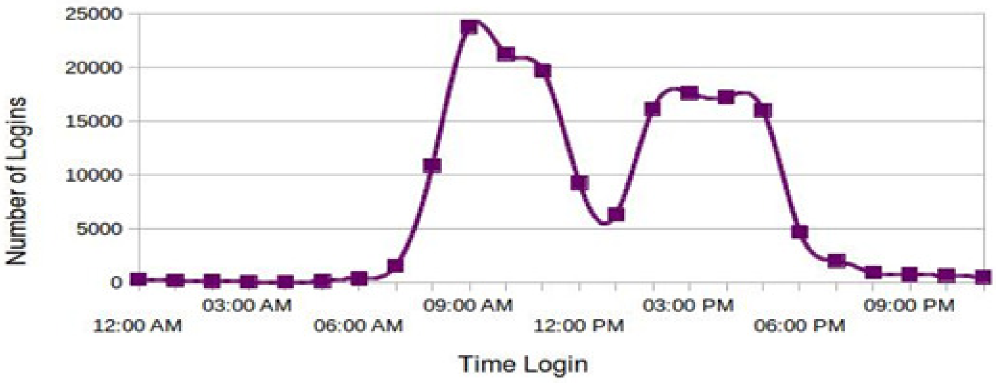


Figure 3. Number of Login with Respect to Time Hour.

The graph in Figure 3 also shows that the number of users login increases from 8 am onwards. After 6:00 pm, the number of login starts to decrease. This is normal since the standard working hours is between 8:30 am to 5:30 pm. In the afternoon, the number of login decreases for around 2 hours before increases back. It is a standard time for users to have their lunch break and stop operation.

5.1.3. Type of Browser and Operating System

Adaptive UAP is able to extract information about the user terminal by looking at the user agent string header send by the browser. There are ways to modify the string such as by using browser extensions (User Agent Switcher [1] and User Agent Selector [2]). However, in this case study, we assume that all user agent strings received by the system are original and unmodified.

Information such as type of browser and operating system running at the user terminal can be determined based on the user agent string. Figure 4 shows an example of a user agent string. In this example, the type of browser and operating system are Safari and iOS 8.1 respectively. It is also clear that the user was using an iPhone smartphone.

Mozilla/5.0 (iPhone; CPU iPhone OS 8_1 like Mac OS X) AppleWebKit/600.1.4
(KHTML, like Gecko) CriOS/39.0.2171.50 Mobile/12B411 Safari/600.1.4

Figure 4. Example of User Agent String Header.

Figure 5 and 6 show the distribution of type of browser and operating system the users used during the data collection period. In term of type of browser usage as shown in Figure 5, majority of login records are from browser Chrome (42.2%), followed by Firefox (33.7%) and Mozilla (16.7%). From the operating system stand point, as shown in Figure 6, 92.2% of the login records are from Windows operating system. The second most popular operating system is Mac OS (4.3%), followed by Linux OS (2.2%).

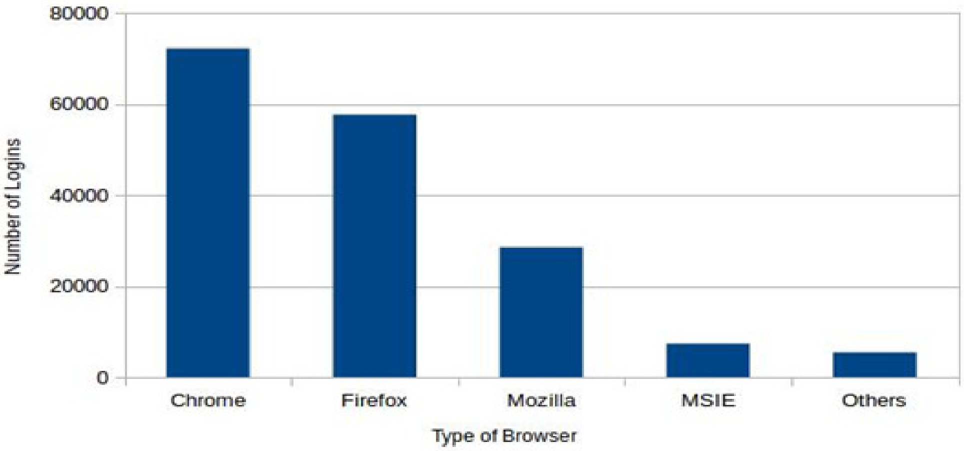


Figure 5. Number of Logins with Respect to Type of Browser

Trust Engine regards both entries of browser and operating system as one attribute factor which carry one weightage value. If one of them is different, Trust Engine consider it as another entry. Figure 7 shows the number of the paired value of browser and operating system derived from the collected login records. The top four most favorite operating system is all Windows 7. In those top list, the most popular browser is Chrome (36.4%), followed by Firefox (31.4%), Mozilla (16.5%) and Internet Explorer (3.9%). The fifth place is browser Chrome with Windows 8 operating system which is 1.5% from the total number of collected login records.

5.1.4. Application

Six UAP gateway servers have been used with only one of them was intended for high trust application. The threshold trust level for low trust and high trust application were set to 10 and 30 respectively. One UAP gateway server was purposely set to 0 so that the final authentication decision by the Trust Engine is not affected by the attribute score value. Table 3 shows the list of all UAP gateway servers used.

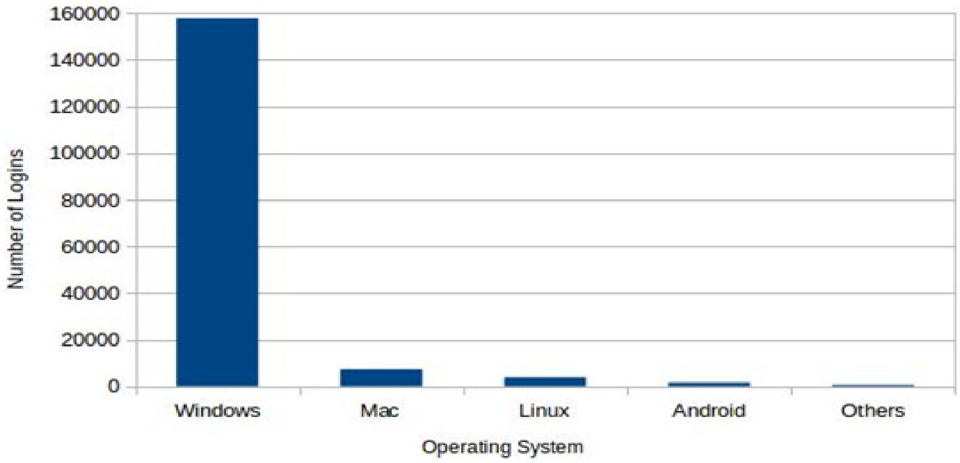


Figure 6. Number of Login with respect to Operating System

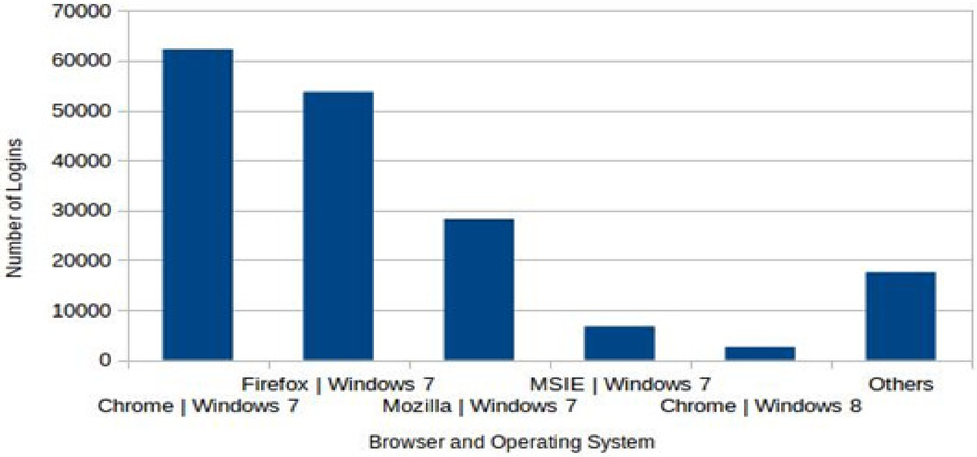


Figure 7. Number of Logins with Respect to paired Browser and OS

Table 3. List of all UAP Gateway Servers.

No	UAP Gateway Server ID	Required Trust
1.	https://uap.mimos.my/shibboleth	0
2.	https://sp.essweb.mimos.my/shibboleth	10
3.	https://bmyhdw.mimos.com/8647196938	10
4.	https://ipms-uap-gw-prod/vfdst35hj3120	10
5.	https://misso.mimos.my/shibboleth	10
6.	https://HT-MIESS1.mimos.my/shibboleth	30

There could be more than one application associated with one UAP gateway. If a user accesses two different applications which are served by the same UAP gateway, Trust Engine gets the same UAP gateway information for both login attempts.

Based on the collected login records, we found that majority (87%) of the logins were for accessing UAP gateway labelled as https://sp.essweb.mimos.my/shibboleth. The second most popular UAP gateway is https://HT-MIESS1.mimos.my/shibboleth which is assigned for high trust application. High trust applications are used to view high confidential documents. Examples of high trust applications are e-payslip to view the salary details online and e-PCB that shows employer's monthly income tax payment to the government tax return agency.

5.2. Ratio Threshold

In this section, we present the effect of User Attribute Score when different configuration of ratio thresholds are used. As explained in Section 3, the value assigned to the ratio threshold determines the qualification of the common context for every attribute factor. The higher the assigned value is, the more strict for any entry to be qualified as a common context. Each individual user may have different pattern of behavioral profile.

The same collected data that had been stored in table data_log was used as the input parameters to our testbed. The experiment setup allows us to see the effect of ratio threshold value to the number of activated attribute factor for every login access. The configuration of the testbed is shown in Table 4.

Table 4. Testbed Configuration

Entry		**Value**
Attribute		**Weightage**
1.	Geographical location	8
2.	Time	6
3.	browserOS	4
4.	Application	2
Credential		**Weightage**
1.	password	13
2.	smsPin	20
3.	otp	20
4.	certificate	40
5.	tck	20
6.	tckbar	20
Time Interval		14 days
Threshold Ratio Percentage		10,30 and 50

The number of events when the attribute factors are activated is recorded and analyzed. In each of those events, the respected attribute factors affect the outcome of the Attribute Score which then reduces the final result of the user established trust. The summary of the results from the experiment is depicted as in Figure 8. The last column labelled 'none' represents the login events where no single attribute factor was activated. In this experiment we assume that there is no adversary that wish to attack our system. In additional, all of the participating users were not aware of the profile tracking and login under their own normal behavior.

We used different values (10, 30 and 50) for ratio threshold percentage to see their affect to each Attribute Factor. From the Figure 8, we can see that in general for every attribute factor, the higher the ratio threshold is, the higher number of events where the attribute factors are activated. The number of events where no attribute factor is activated decreases when higher threshold ratio percentage is used.

Type of browser contributes the highest number of events followed by application, operating system, time login and location. We can conclude that users as individual have more tendency to use different type of browser compared to other factors. Location is the least factor. One main reason is because most of the users are the employers of the company and most of the applications can only be accessed from the company private network.

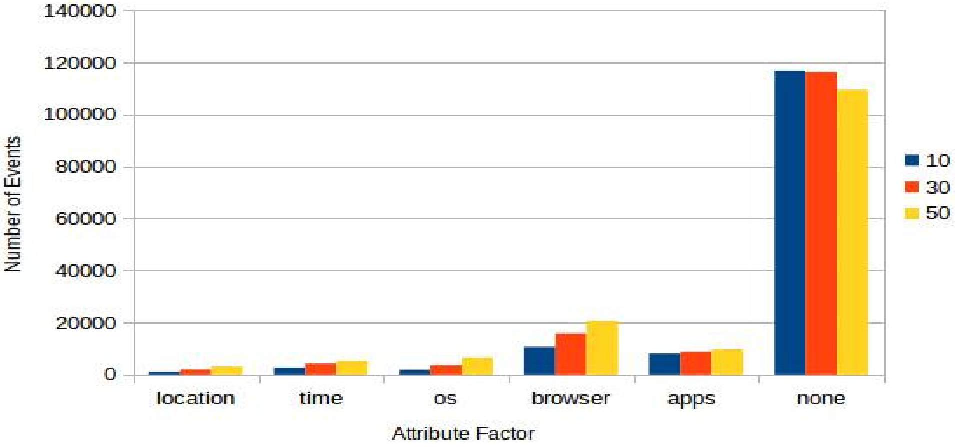


Figure 8. Total Number of Events for every Attribute Factor.

6. CONCLUSION

Adaptive authentication is an additional security layer that uses risk factor analysis to make authentication decision. Users' past login records are used to form user behavior profile. Context information in login records such as user geographical location and time login are important parameters to reflect user common behavior profile. If the user logins from a different environment from the established behavior profile, adaptive authentication calculates the risk associated with the deviation and may request the user to present additional authentication method to be authenticated. We collects actual login records from the production environment for more than 6 months. The login records profiling with regards to relevant attribute factors is analyzed. We also develop a testbed environment that uses those records to evaluate our system when different percentage of threshold ratios are used. In this paper, we present the results of the experiment. For the future works, we plan to evaluate our testbed system for other parameters setup.

ACKNOWLEDGEMENTS

We acknowledge the support provided by Ministry of Science, Technology and Innovation (MOSTI) in funding the MIMOS Unified Authentication Platform (UAP) project through the Tenth Malaysia Plan (10MP). The completion of the project allows the delivery of a centralized authentication infrastructural platform for web applications.

REFERENCES

[1] http://chrispederick.com/work/user-agent-switcher/. [Retrieved 30 January 2015].

[2] https://chromeuseragentselector.wordpress.com/. [Retrieved 30 January 2015].

[3] Khairul Azmi Abu Bakar and Galoh Rashidah Haron. Adaptive authentication: Issues and challenges. In World Congress on Computer and Information Technology (WCCIT), pages 1–6, June 2013.

[4] Khairul Azmi Abu Bakar and Galoh Rashidah Haron. Adaptive authentication based on analysis of user behavior. In Science and Information Conference (SAI), pages 601–606, August 2014.

[5] Galoh Rashidah Haron, Dharmadharshni Maniam, Vijayakumari Sadavisam, and Wong Hon Loon. Re-engineering of web reverse proxy with shibboleth authentication. In The 7th International Conference for Internet Technology and Secured Transactions (ICITST-2012), pages 325–330, 2012.

[6] ip2location. ip2location home page. http://www.ip2location.com. [Retrieved April 2012].

[7] Shibboleth. Shibboleth documentation. https://wiki.shibboleth.net. [Retrieved January 2014].

AUTHORS

Khairul Azmi Abu Bakar received the degree in Computer Engineering from Iowa State University, USA in 1995 and master degree in Communication and Computer from National University of Malaysia in 2002. He was awarded Ph.D. degree in Electrical Engineering from University of Strathclyde, United Kingdom in 2012 for the study on free-riding nodes in an open MANET. He is currently a staff researcher at MIMOS Berhad where he has been since 1996. He has been involved in several R&D projects in the field of micro-controller, smartcard, security systems under open source platform. His primary research interests include wireless ad hoc security, authentication system and computer network.

Nor Izyani Daud was born in Kuala Lumpur, Malaysia. She received the B.A Hons Degree in Information Technology, Artificial Intelligence majoring from the Universiti Utara Malaysia in 2000; and Master Degree in Real Time Software Engineering from Universiti Teknologi Malaysia in 2006. In 2006, she joined MIMOS Berhad as a Senior Engineer. She has been working in Information Security areas; for example smart card programming, security scanning and analyzing. She also involved with CMMI-Capability Maturity Model Integration implementation in the organization.

Mohd Shafeq Hasan was born in Kuala Lumpur, Malaysia in 1987. He received the B.A Hons Degree in Information Technology, Java majoring from the Kuala Lumpur Metropolitan University College in 2011, He joined MIMOS Berhad as a Junior Engineer. He has been working in Information Security areas; for example adaptive authentication, J2EE and Spring Security

SENSOR SELECTION SCHEME IN WIRELESS SENSOR NETWORKS: A NEW ROUTING APPROACH

Mohammad Alwadi[1] and Girija Chetty[2]

[1]Department of Information Sciences and Engineering,
The University of Canberra, Canberra, Australia
u3019769@uni.canberra.edu.au
[2]Faculty of ESTM, The University of Canberra, Canberra, Australia
girija.chetty@canberra.edu.au

ABSTRACT

In this paper, we propose a novel energy efficient environment monitoring scheme for wireless sensor networks, based on data mining formulation. The proposed adapting routing scheme for sensors for achieving energy efficiency. The experimental validation of the proposed approach using publicly available Intel Berkeley lab Wireless Sensor Network dataset shows that it is possible to achieve energy efficient environment monitoring for wireless sensor networks, with a trade-off between accuracy and life time extension factor of sensors, using the proposed approach.

KEYWORDS

Wireless sensor Networks, Physical environment Monitoring , machine learning, data mining, feature selection, adaptive routing.

1. INTRODUCTION

Wireless and wired sensor networks (WSNs/SNs) has become a focus of intensive research today, especially for monitoring and characterizing of large physical environments, and for tracking environmental or physical conditions such as temperature, pressure, wind and humidity. A wireless or a wired sensor network (WSN/SN) consists of a number of sensor nodes (few tens to thousands) storing, processing and re-laying the sensed data, often to a base station for further computation [1,2]. Sensor networks can be used in many applications, such as wildlife monitoring [3], military target tracking and surveillance [4], hazardous environment exploration [5], and natural disaster relief [6]. Many of these applications are expected to run unattended for months or years. Sensor nodes are however constrained by limited resources, particularly in terms of energy. Since communication is one order of magnitude more energy consuming than processing, the design of data collection schemes that limit the amount of transmitted data is therefore recognized as a central issue for wireless sensor networks. An efficient way to address this challenge is to approximate, by means of mathematical models, the evolution of the

David C. Wyld et al. (Eds) : WiMo, ITCSE, ICAIT, NC - 2015
pp. 73–79, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.51007

measurements taken by sensors over space and /or time. Indeed, whenever a mathematical model may be used in place of the true measurements, significant gains in communications may be obtained by only transmitting the parameters of the model instead of the set of real measurements.

Since in most cases there is little or no a priori information about the variations taken by sensor measurements, the models must be identified in an automated manner. This calls for the use of machine learning and data mining techniques, which allow modelling the variations of future measurements on the basis of past measurements.

In this paper, we introduce a novel data mining based formulation for energy efficient WSN (Wireless Sensor Network) monitoring. The proposed approach involves an adaptive routing scheme to be used for energy efficiency and is based on selecting most significant sensors for the accurate modelling of the WSN environment. The experimental validation of the proposed scheme for publicly available Intel Berkeley lab Wireless Sensor Network dataset shows it is indeed possible to achieve energy efficiency without degradation in accurate characterization and understanding of WSN environment. The proposed machine learning and data mining formulation for achieving energy efficiency, provides better implementation mechanism in terms of trade-off between accuracy and energy efficiency, due to an optimal combination of feature selection and classifier techniques used in machine learning chain. By approaching the complexity of WSN/SN with a data mining formulation, where each sensor node is equivalent to an attribute or a feature of a data set, and all the sensor nodes together forming the WSN/SN set up - equivalent to a multiple features or attributes of the data set, it is possible to use powerful feature selection, dimensionality reduction and learning classifier algorithms from machine learning/data mining field, and come up with an energy efficient environment monitoring system [7]. In other words, by employing a good feature selection algorithm along with a good classification algorithm, for example, it is possible to obtain an energy efficient solution with acceptable characterization or classification accuracy (where the WSN/SN set up is emulated with a data set acquired from the physical environment). Here, minimizing the number of sensors for energy efficiency is very similar to minimizing the number of features with an optimal feature selection scheme for the data mining problem. Further, the number of sensors chosen by the feature selection scheme, leads to a routing scheme for collecting the data from sensors, transmitting them until it reaches the base station node. As accuracy of data mining schemes rely on amount of previous data available for predicting the future state of the environment, it is possible to obtain an adaptive routing scheme, through the life of WSN, as more and more historical data becomes available, allowing trade-off between energy efficiency and prediction accuracy. We have shown that it is possible to do this, with an experimental validation of our proposed scheme with a publicly available WSN dataset acquired from real physical environment, the Intel Berkeley Lab [8]. Rest of the paper is organized as follows. Next Section describes the details of the dataset used, and Section 3 de-scribes the proposed machine learning - data mining approach. The details of experimental results obtained are presented in Section 4, and the paper concludes in Section 5, with conclusions and plan for further research.

2. DATA SET DESCRIPTION

The publicly available data set used for experimental validation consists of temperature and humidity measurements, which come from a deployment of 54 sensors in the Intel research laboratory at Berkeley [8]. The deployment took place between February 28th and April 5th,

2004. A picture of the deployment is provided in Figure 1 below, where sensor nodes are identified by numbers ranging from 1 to 54.

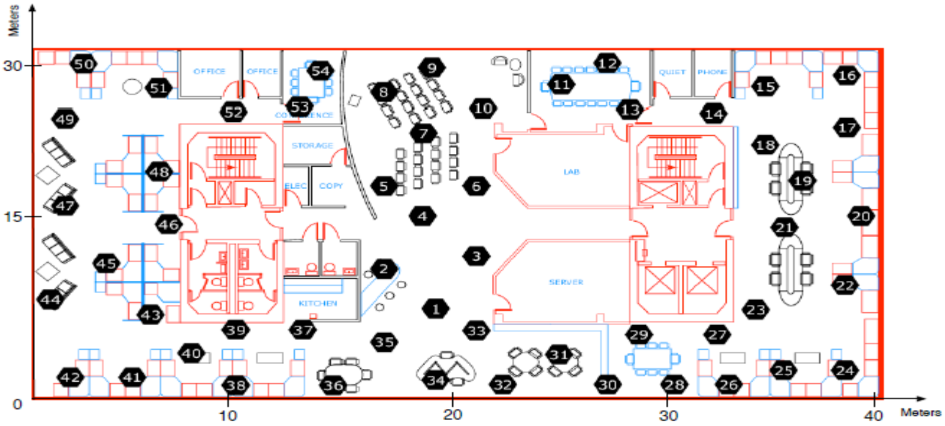


Figure 1. Intel Berkeley Wireless Sensor Network Data set: location of 54 sensors in an area of 1200m^2

Many sensor readings from WSN test bed were missing, due to this being a simple prototype. We selected from this data set few subsets of measurements. The readings were originally sampled every thirty-one seconds. A pre-processing stage where data was partitioned was applied to the data set. After pre-processing, we prepared several subsets of data. The approach we used for this WSN test bed, involves, an assumption that data collection and transmission is done by some of the sensors (source nodes), that purely sense the environment and transmit their measurement to collector nodes (sink nodes/base station), and based on the relative distance between source nodes and sink nodes, the route or the path taken for sensor data to be transmitted from one node to other is predetermined at sink/base station node. Those nodes who actively participate in sensing the environment, and transmit the data, consume the power and those who do not participate in this activity do not consume any power. This is how the WSN can be made energy efficient; by involving optimum number of sensors to participate in environment sensing and transmission task, and leaving non-participating sensors in sleep mode (no energy consumption). This can however, impact on the accuracy of sensing the environment, if number of sensors participating in routing scheme is not properly chosen. To ensure a trade-off between accuracy and energy efficiency is achieved, it is essential that a dynamic or adaptive routing scheme is used, where, the machine learning/data mining technique can use larger training data from previous/historical data sets to predict the future environment accurately, and continuously adapt the routing scheme for nodes based on a threshold error measure for prediction accuracy and energy efficiency. Next Section describes the proposed machine learning data mining scheme used for sensor selection and routing in this approach.

3. Sensor Selection And Routing Approach

The sensor selection and routing approach is based on a feature selection technique that selects the attributes (sensors), by evaluating the worth of a subset of attributes by considering the individual predictive ability of each feature/sensor along with the degree of redundancy between them. Subsets of features that are highly correlated with the class while having low inter correlation are preferred [9, 10]. Further, this feature/sensor selection algorithm identifies locally predictive attributes, and iteratively adds the attributes with the highest correlation with the class as long as there is not already an attribute in the subset that has a higher correlation with the attribute in question. Once the appropriate group of sensors are selected, the prediction of sensor output at sink node or base station is done by linear regression algorithm, using the Akaike criterion [10, 11], which involves stepping through the attributes, removing the one with smallest standardised coefficient until no improvement is observed in the estimate of the error given by Akaike information metric. Figure 2 and the table below shows how the sensor selection evolves as the training data (historic data) used for predicting the sink sensor output is increased, and ensures the prediction accuracy/error is maintained at a particular threshold value.

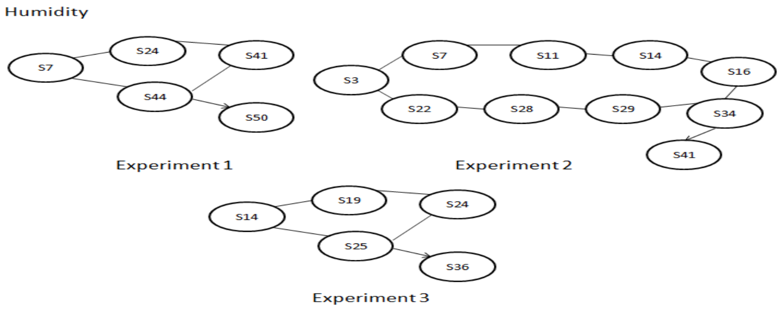


Figure 2. Sensor selection map for Humidity Experiment 1,2 and 3

4. EXPERIMENTAL RESULTS AND DISCUSSION

Different sets of experiments were performed to examine the relative performance of sensor selection and routing approach proposed here. We used k-fold stratified cross validation technique for performing experiments, with k=2, 5 and 10, based on the training data available (using larger folds for larger training data). Further, to estimate the relative energy efficiency achieved, we performed experiments with all sensors (without feature selection/sensor selection) algorithm, and with sensors selected by feature selection algorithm. As mentioned before, the feature selection algorithm allows selection of an optimal number of features or sensor nodes needed to characterize or to classify the environment (which in turn leads to an energy efficient scheme). Further, time taken to build the model is also an important parameter, particularly for adaptive sensor routine scheme to be used for real time environment monitoring.

Table 1. Results for Three experiments scenarios

Exp number	**Number of Sensors**	**Number of Training samples**	**Features/sensors selected**	
1	52	35	7,24,41,44,50	
2	52	2700	3,7,11,14,16,22,28,29,34,41	
3	52	5400	14,19,24,25,36	
Exp number	**Time (No F selection)**	**Time(F Selection)**	**RMSE No F selection**	**RMSE with F selection**
1	0.01	0.01	3.82	0.04
2	0.14	0.01	0.96	2.11
3	0.17	0.03	1.91	4.56

For the first set of experiments, we used first 52 sensors and a small set of training samples (35 Humidity measurements). As can be seen from the sensor locations shown in Figure 2, sensor 50 is the sink node (emulating base station node), and sensors 1 to 49 participate in measuring and transmitting the environment around them to the sink node, where the machine learning prediction task is to estimate the measurement at sink node (sensor 50). The RMS error (root mean squared error) at the sink node (node 50) provides a measure of prediction For all source sensor nodes (1-49) in WSN participating in measuring the temperature in the environment and sending it to sink node, the RMS error is 3.82%, and with sensor selection scheme used with only 5 sensors participating in routing scheme, the RMS error is 0.04%. As can be seen in Table 1, with a moderate degradation in accuracy (3.82% to 0.04%), energy efficiency achieved is of the order of 52 (52/5). We used a new measure for energy efficiency, the life time extension factor (LTEF), which can be defined as:

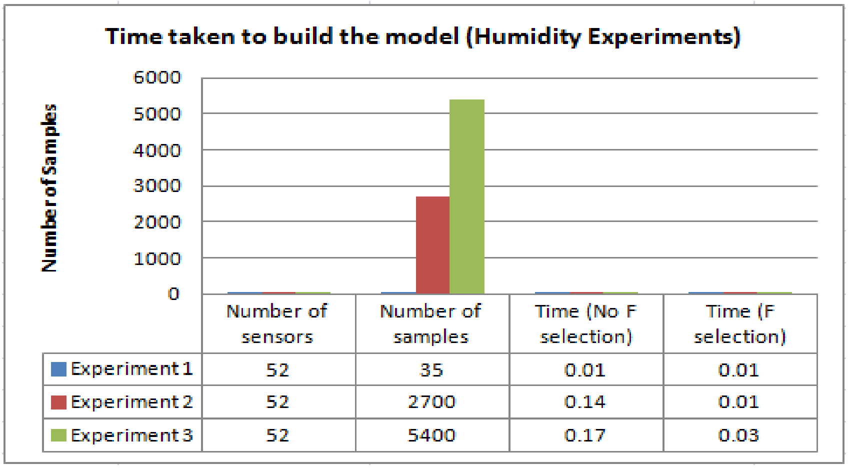
$$\mathrm{LTEF} = \frac{\text{Total number of sensors}}{\text{Sensors participating in the routing scheme}}$$

With 5 sensors out of 52 sensor nodes in active mode, the LTEF achieved is around 10 times, and 52 sensor nodes are in sleep mode. The trade off is a slight reduction in accuracy. This could be due to less training data used. We used only 35 temperature samples for prediction scheme. With more data samples used in the prediction scheme, performance could be better. To test this hypothesis, we performed next set of experiments.

For second set of experiments, we used 2700 training samples collected on different days. As can be seen in Table 1, with larger training data size, we found that the participating sensors in the routing scheme are different, as the proposed feature selection algorithm chooses different set of sensors (3,7,11,14,16,22,28,29,34,41). We used 52 sensors for this set of experiments, as two of the sensors (sensor 5 did not have more than 35 measurements). With all 52 sensors in the routing scheme, the RMS errors is 0.96%, and with 10 sensor nodes (3,7,11,14,16,22,28,29,34,41), the error is 2.11%. This was not a significant improvement in prediction accuracy (from 0.96% to 2.11%), with life time extension of 5.2 (52/10). As is evident here, by using larger training data (2700 temperature measurements), it was possible to achieve an improvement in prediction accuracy and energy efficiency as well.

To examine the influence of increasing training data size, we performed third set of experiments with 5200 samples. The performance achieved for this set of experiments is shown in Table 1. Here the adaptive routing scheme based on proposed feature selection technique selects 5 sensors (14,19,24,25,36). For this set of experiments, the RMS error varies from 1.91% for all sensors participating in the scheme to 4.56 % with LTEF of 6.6 (52/5). Though there is no degradation in prediction accuracy, there is not much improvement in energy efficiency, with doubling of training data size for the building the model This could be due to overtraining that has happened, with the network losing its generalization ability. So by increasing training data size, it may not be just possible to achieve performance improvement, for pre-diction accuracy (RMS error) and energy efficiency (LTEF), and a trade off may be needed. An optimal combination of training data size, and number of sensors actively participating in routing scheme can result in energy efficient WSN, without compromising the prediction accuracy.

Further, another important parameter is model building time, as for adaptive sensor routing scheme to be implemented in real time WSN environment, routing scheme has to dynamically compute the sensors that are in active mode and in sleep mode. Out of 3 experimental scenarios considered here, as can be seen from Table 1, the model building time from 0.01seconds to 0.01 seconds for experiment 1, from 0.14 seconds to 0.01 seconds for experiment 2, and from 0.17 seconds to 0.03 seconds for experiment 3. So, the proposed adaptive routing scheme for sensor selection provides an added benefit of reduced model building times, suitable for real time deployment. figure below shows the time taken to build the model.



	Number of sensors	Number of samples	Time (No F selection)	Time (F selection)
🟦 Experiment 1	52	35	0.01	0.01
🟥 Experiment 2	52	2700	0.14	0.01
🟩 Experiment 3	52	5400	0.17	0.03

Figure 3. Time taken to build the model (Humidity Exp1,2 and 3)

5. Conclusions

Energy sources are very limited in sensor networks, in particular wireless sensor net-works. For monitoring large physical environments using SNs and WSNs, it is important that appropriate intelligent monitoring protocols and adaptive routing schemes are used to achieve energy efficiency and increase in lifetime of sensor nodes, with-out compromising the accuracy of characterizing the WSN environment. In this paper, we proposed an adaptive routing scheme for sensor nodes in WSN, based on machine learning data mining formulation with a feature selection algorithm that selects few most significant sensors to be active at a time, and adapts them continuously as time evolves. The experimental validation for a real world publicly available WSN dataset, proves our hypothesis, and allows energy efficiency to be achieved without compromising the prediction accuracy, with an added benefit in terms of reduced model building times. Further work involves, developing new algorithms for sensor selection and environment characterization with WSNs and their experimental validation with other similar datasets, that can lead to better energy efficiency. Also, our further re-search involves extending this work with adapting these classifiers for big data stream data mining schemes, for real time dynamic monitoring of complex and large physical environments in an energy efficient manner.

References

[1] Ping, S., Delay measurement time synchronization for wireless sensor networks. Intel Research Berkeley Lab, 2003.

[2] Hall, M., et al., The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter, 2009. 11(1): p. 10-18.

[3] Csirik, J., P. Bertholet, and H. Bunke. Pattern recognition in wireless sensor networks in presence of sensor failures. 2011.

[4] Nakamura, E.F. and A.A.F. Loureiro, Information fusion in wireless sensor networks, in Proceedings of the 2008 ACM SIGMOD international conference on Management of data2008, ACM: Vancouver, Canada. p. 1365-1372.

[5] Bashyal, S. and G.K. Venayagamoorthy. Collaborative routing algorithm for wireless sensor network longevity. 2007. IEEE.

[6] Richter, R., Distributed Pattern Recognition in Wireless Sensor Networks.

[7] Alwadi, M. and G. Chetty, Energy Efficient Data Mining Scheme for Big Data Biodiversity Environment. 2014.

[8] Bodik, P., et al., Intel lab data. Online dataset, 2004.

[9] Hall, M.A., Correlation-based feature selection for machine learning, 1999, The University of Waikato.

[10] Akaike, H., Information theory and an extension of the maximum likelihood principle, in Selected Papers of Hirotugu Akaike1998, Springer. p. 199-213.

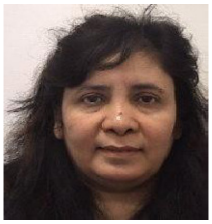
[11] Ashraf, M., et al., A New Approach for Constructing Missing Features Values. International Journal of Intelligent Information Processing, 2012. 3(1).

AUTHORS

Mohammad Alwadi

Mohammad Alwadi has bachelors degree in Computer Science from Jordan, Masters degree from the University of Canberra, Australia and Currently a PhD candidate in the Faculty of Information Science and Engineering at The University of Canberra, Australia. Mohammad research interests are in the area of computer networking, Data Mining and wireless sensor networks This author has 8 publications and working towards the submission of his PhD Thesis at the University of Canberra. Australia. He has more than 6 years of experience in the research field of Data mining , sensors networks and wireless sensor networks.

Dr.Girija Chetty



Girija has a Bachelors and Masters degree in Electrical Engineering and Computer Science from India, and PhD in Information Sciences and Engineering from Australia. She has more than 25 years of experience in Industry, Research and Teaching from Universities and Research and Development Organisations from India and Australia, and has held several leadership positions including Head of Software Engineering and Computer Science, and Course Director for Master of Computing (Mainframe) Course. Currently, she is an Associate Professor, and Head of the Multimodal Systems and Information Fusion Group in University of Canberra, Australia, and leads a research group with several PhD students, Post Docs, research assistants and regular International and National visiting researchers. She is a Senior Member of IEEE, USA, and senior member of Australian Computer Society, and her research interests are in the area of multimodal systems, computer vision, pattern recognition, data mining, and medical image computing. She has published extensively with more than 120 fully refereed publications in several invited book chapters, edited books, high quality conference and journals, and she is in the editorial boards, technical review committees and regular reviewer for several IEEE, Elsevier and IET journals in the area related to her research interests. She is highly interested in seeking wide and interdisciplinary collaborations, research scholars and visitors in her research group.

INTENTIONAL BLANK

AN GROUP BEHAVIOR MOBILITY MODEL FOR OPPORTUNISTIC NETWORKS

GuoDong KANG[1] and GuoLiang KANG[2]

[1]DFH Satellite Co., Ltd., 100094, Beijing, China
kongton584@163.com
[2]University of Technology, Sydney 15 Broadway, Ultimo NSW 2007
kgl.prml@gmail.com

ABSTRACT

Mobility is regarded as a network transport mechanism for distributing data in many networks. However, many mobility models ignore the fact that peer nodes often carried by people and thus move in group pattern according to some kind of social relation. In this paper, we propose one mobility model based on group behavior character which derives from real movement scenario in daily life. This paper also gives the character analysis of this mobility model and compares with the classic Random Waypoint Mobility model.

KEYWORDS

Mobility model , Group behavior , Opportunistic network

1. INTRODUCTION

Nowadays mobility is playing an important role more and more in many wireless networks such as mobile ad-hoc networking (MANET) and delay-tolerant networking (DTN) [2, 3]. One common important fact for these opportunistic networks is that mobile devices among them are often carried by people. From this point of view, the mobility of mobile devices is exactly the same as the movement of persons who are taking them. The idea of mobility structure in this paper comes from people's movement custom or character in everyday life.

Usually we can observe one familiar scenario: persons in the same group often go to some place together then to another place together meantime everyone has his own walking or moving style. How to construct this kind of mobility is the main work of this paper.

To construct this kind of mobility model, we have to face two questions: how to allocate the given nodes into different groups according to certain method; how to describe and construct nodes' common group mobility behavior and every node's individual mobility behavior. The mobility model proposed in this paper thoroughly considers nodes' individual realistic movement character combing common group character based group division method.

David C. Wyld et al. (Eds) : WiMo, ITCSE, ICAIT, NC - 2015
pp. 81–89, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.51008

The rest of this paper is organized as follows: Section II describes the allocation method of mobile nodes in opportunistic network. Section III proposes the setup method of the mobility models based on common group-behavior character and individual self-behavior character. Section IV gives analysis of the proposed new mobility model. Conclusion is given in Section V.

2. GROUP ALLOCATION METHOD

The group allocation method can be undertaken as following two steps: Firstly, the network should find and construct the relations among whole nodes in the form of mathematic matrix which will be regarded as the basis of second step. Then whole nodes are allocated into different groups according to the relations founded in first step.

2.1. Node Relation Setup

Normally, there is some kind of social or biological relations among those nodes since they are carried by people. To set up this kind of mutual relation, we adopt the classical method of representing social or biological network, weighted graphs. The strength of mutual relation between any node pair is represented using a value in the range [0, 1]. As a consequence, the network internal relation can be described as a relation matrix with a dimension of N*N where N is the total number of nodes in the network. At present, there are several models that describe the key properties of real-life network, such as random graph, small world, etc. Some research work show that the properties of these random graphs, such as path lengths, clustering coefficients, can't be regarded as accurate models of realistic networks [5, 6, 7]. Here, we choose the geometric random graph to be the network model. In this kind of model, the geometry relations of nodes have strong association with the social relation of nodes. That means when any two nodes are in the radio range of each other, the social relation exists. On the contrary, since they even can't communicate with each other, we think that there is no any social interaction between them. So when the Euclidean distance between any two nodes is smaller than radio range R, the corresponding element of the social relation matrix M is set to be 1 or else set to be 0 as follows.

$$m_{i,j} = \begin{cases} 1 \ \ if \ i \neq j \ and \ \|P_i - P_j\| \leq R, \\ 0 \ \ otherwise \end{cases}$$

It should be emphasized that the relations value of one node and itself is regarded to be zero in the matrix. In [6] it is shown that in two or three dimensional area using Euclidean norm can supply surprisingly accurate reproductions of many features of real biological networks.

For example, 100 nodes are uniformly distributed as shown figure 1. Lines are used to represent the mutual relations among these nodes whose radio range is assumed to be 10 meters here.

2.2. Nodes Group Allocation

Once the relation matrix M is obtained, groups can be detected. Group structure is one of the common characters in many real networks. However, finding group structures within an arbitrary network is an acknowledged difficult task. A lot of work has been done on that. Currently, there are several methods that can achieve that goal, such as Minimum-cut method, Hierarchical clustering, Girvan-Newman algorithm, etc.

Minimum-cut method is one of the oldest algorithms for dividing networks into parts. This method uses in load balancing for parallel computing in order to minimize communication between processor nodes. However, this method always finds communities regardless of whether they are implicit in the structure, and it can only find a fixed number of them. So it is less than ideal for finding community structure in general networks [4].

Hierarchical clustering is another method for finding community structures in network. This method detects the community by defining a similarity measure quantifying some (usually topological) type of similarity between node pairs.

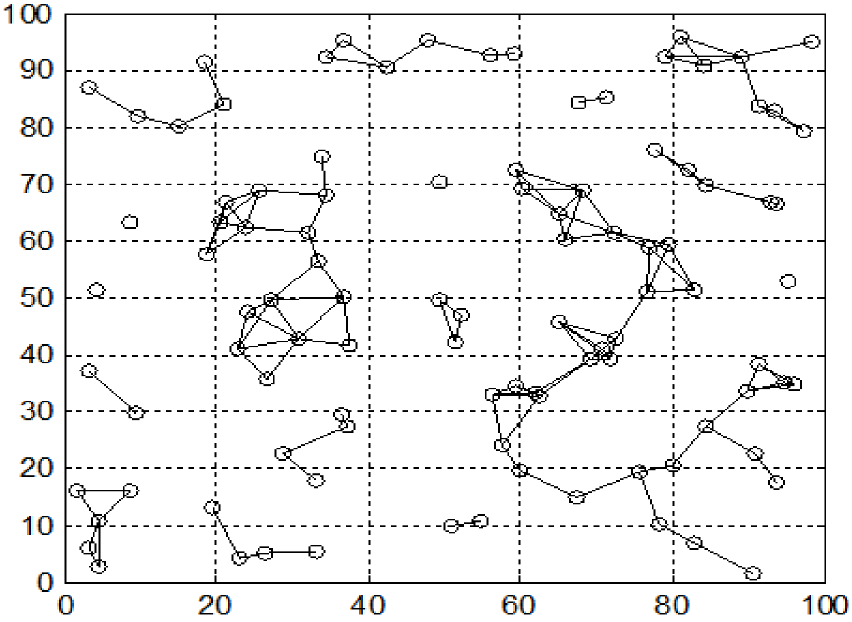


Figure 1 An example of nodes relations. Small circles represent nodes. Short lines represent the relation among them.

The Girvan–Newman algorithm is one commonly used algorithm for finding communities [12]. It identifies edges in one network that lie between communities and then removes them, leaving behind just the communities themselves. But this method runs slowly which makes it impractical for networks of more than a few thousand nodes [11].

Modularity maximization is one of the most widely used methods for community detection [11]. Modularity is a benefit function that measures the quality of a particular division of a network into communities. This method detects the community structure of high modularity value by exhaustively searching over all possible divisions [8].

In this paper we adopt modularity maximization as the social group detection method. Modularity maximization is one of the most widely used methods for group detection [11].

Modularity is a benefit function that measures the quality of a particular division of a network into groups. This method detects the group structure of high modularity value by exhaustively searching over all possible divisions [8].

In real networks, the modularity value is usually in the range [0.3, 0.7]; 1 means a very strong group structure and 0 means no better than random.

3. MOBILITY MODEL SETUP

3.1. Mobility Scenario Description

Usually peer nodes in one group act in concert, for example, these nodes may go from office to restaurant together for lunch, maybe after lunch they will have a rest and then go from restaurant to cinema for entertainment.

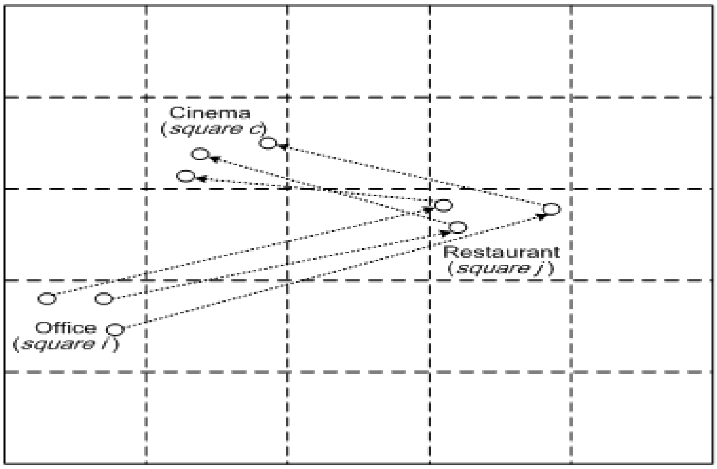


Figure 2 Illustration of mobility scenario

In terms of this realistic mobility scenario, this paper here models the office, restaurant and cinema to be small squares as shown in figure 2 in simulation scenario.

When the group moves from office (square i) to restaurant(square j), the restaurant (square j) will be regarded as group-destination for all the peer nodes.

When the group continues to move from restaurant the cinema, one new square (square c) in the simulation will be chosen as the new group-destination corresponding to the cinema.

It should be noted that peer nodes in the same group also have their own self-destination. This means that these nodes self-destinations may be different positions corresponding to different points in the group-destination square.

In realistic scenario, another important phenomenon is as follows: when nodes move towards the destinations, some of them can move fast and some of them perhaps move slowly. This results that these faster nodes can reach their self-destinations earlier than the slower nodes. After arriving, these faster nodes would wait other slower nodes for a while before departure for next new group destination.

3.2. Mobility Model Abstraction

Before moving, each group will randomly choose one small square in the simulation area to be its group-destination. And then each peer node in the group will choose one position in that small square to be its self-destination individually. Once they know their self-destinations, each node will move straightly towards its goal at some speed value. Because of the speed and destination position difference of nodes in one group, they can't ensure that all of them will reach their self-destinations at the same time. So the former arriver will pause for a period of time to wait until all his peer partners arrive. Upon all the members of the group achieve arrival, they will choose a new small square to be their next common destination and move towards it.

3.3. Mobility Mathematical Model

Assume there are total N nodes in one group, the n^{th} node is denoted as node $n(1 \le n \le N)$. The beginning square is denoted as S_j, so the beginning position of node n can be denoted as $P_n(t|t=0) \in S_i$. The group-destination square is denoted as S_j, the self-destination of node n in S_j is denoted as $P_n(t|t=0) \in S_i$.

At time $t(t \ge 0)$ the position of node n is a function of coordinate $X_{dn}(t)$ and $Y_{dn}(t)$ which can be expressed as follows:

$$P_n(t+1) = \begin{bmatrix} X_n(t+1) \\ Y_n(t+1) \end{bmatrix} = \begin{bmatrix} X_n(t) + V_n(t) \cdot \cos\Phi_n(t) \\ Y_n(t) + V_n(t) \cdot \sin\Phi_n(t) \end{bmatrix}$$

Where

t is the current moment;

$t+1$ is the next moment;

$\Phi_n(t)$ is the direction function of node n which can be defined as

$$\Phi_n(t) = arctg(\frac{|Y_{dn} - Y_n(t)|}{|X_{dn} - X_n(t)|})$$

$V_n(t)$ is the velocity function which obeys normal distribution.

3.4. Speed Choice

The research in [13] has shown that the walking speed of a pedestrian obeys normal distribution. The measurement in [14] shows that the mean speed of a walking pedestrian is a range which is from 1.16 m/s to 1.58 m/s representing walking normally or walking fast.

In [15], the Manual of Uniform Traffic Control Devices (MUTCD) shows that pedestrians walk with a normal speed of 1.2 m/s (4 ft/sec). [10] indicates a statistics that walking speed for younger pedestrians (ages 14 to 64) was 1.25 m/sec (4.09 ft/sec); for older pedestrians (ages 65 and over) it was 0.97 m/sec (3.19 ft/sec). For designing purposes values of 1.22 m/sec (4 ft/sec) for younger pedestrians and 0.91 m/sec (3 ft/sec) for older pedestrians are appropriate[10].

In this paper, the mean value of walking speed of the node n, μ_{n_v}, equals to 1.22 m/s [10] and its standard deviation σ_{n_v} is uniformly chosen in the range of $[0 \quad 0.26m/s]$ [13]. So the speed of node n at time t is

$$V_n(t) \sim N(\mu_{n_v}, \sigma_{n_v})$$

The probability density function of $V_n(t)$ can be expressed as:

$$f(v) = \frac{\exp(-\frac{(v-\mu_{n_v})^2}{2\sigma_{n_v}^2})}{\sigma_{n_v}\sqrt{2\pi}}$$

4. MODEL CHARACTER ANALYSIS

4.1. Classical Individual Mobility Model-RWP

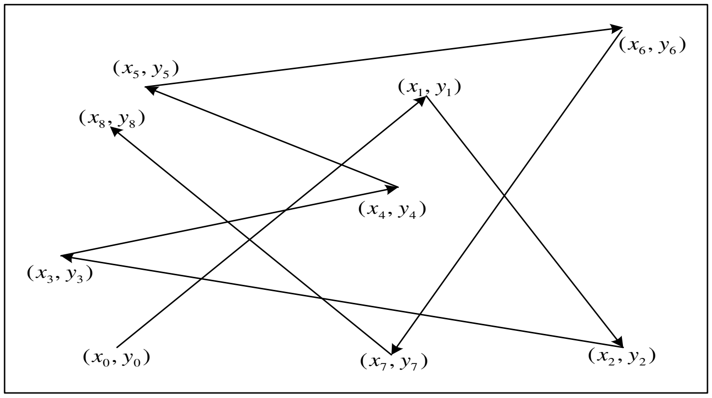


Figure 3 Illustration of RWP mobility

Josh et al. [8] present a Random Waypoint model. In this model a mobile node is initially placed at random location in the simulation area with a destination and speed according to the random distribution. Then the mobile node moves from its current location to it’s destinations via a straight line. After arrival at the destination, the mobile node determines a pause time according to a random distribution, and after that chooses a new destination.

4.2. Analysis and Comparison

The proposed group model in this paper can be regarded as one improvement version of RWP. To analyze the character of these two mobility models, two important parameters inter-contact time and contact time are adopted. These two parameters describe the characteristics of connection opportunities of the network, i.e. how many and when do they occur, how often and how long. Contact time is defined as the time interval during which the two nodes can keep contact. The time interval from this contact to next one is defined as inter-contact time during which nodes can’t communicate. [1] These two parameters are very important for opportunistic network. Contact time can help us determining the capacity of opportunistic networks. Inter-contact time is a parameter which strongly affects the feasibility of the opportunistic networking.

The new mobility model in this paper is named as Peer Node Group Mobility Model (PNGM) here. We let each mobility model run 1000 seconds for one experiment and made 50 times similar experiments for each mobility model. Figure 4 and Figure 5 give the contact time distribution and inter-contact time distribution in different coordinate systems. In Fig. 4 (a), we can see that the new mobility model' inter-contact time distribution behave an exponential distribution plot using log-log coordinate system. In Fig. 4 (b), the contact time distributions show more evident difference. The RWP mobility models' curves are still exponential-like curves. In contrast, the shape of PNGM model' curves are changed. The PNGM model's curve is one kind of transition from exponential-like curve to power- law-like curve. This shows PNGM model behaves stronger group character than RWP model. In Fig. 5 we can see the difference between the curves more clearly using semi-log coordinate system. The exponential nature of the inter-contact time shown in Fig. 4 (a) becomes straight line form in Fig. 5 (a).

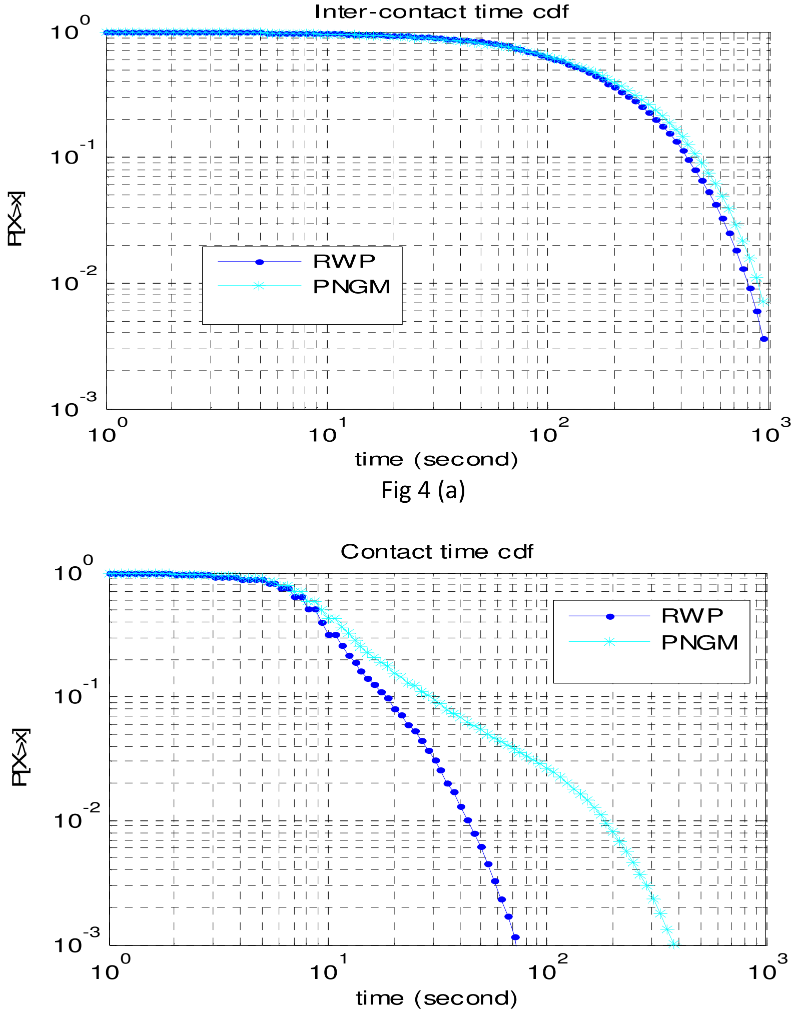


Fig 4 (a)

Fig4 (b)

Figure 4. Inter-contact time cdf and contact time cdf in log-log coordinates.

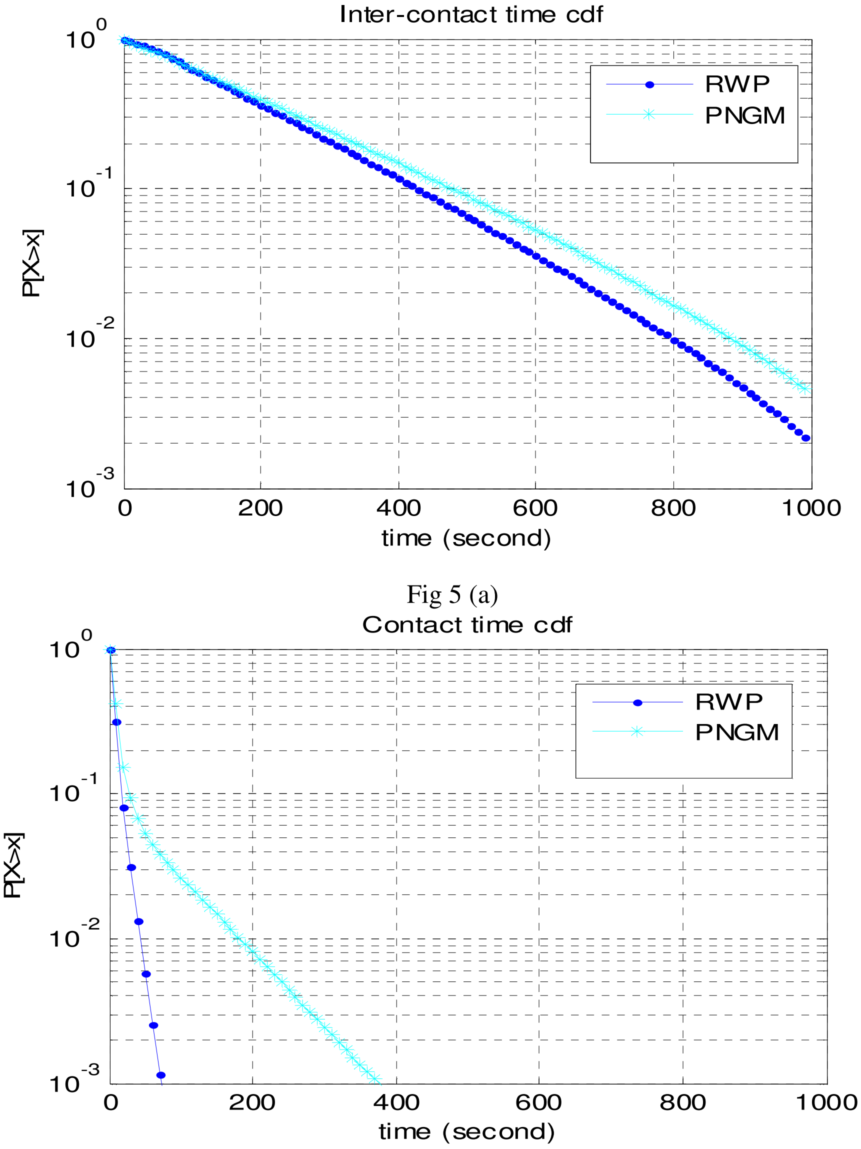


Fig 5 (a)

Fig5 (b)

Figure 5. Inter-contact time cdf and contact time cdf in semi-log coordinates.

5. Conclusions

In this paper, we proposed one new group mobility model with group character. We analyze its curves about contact time, inter-contact time. The curves show different shapes with the classical individual model RWP. By comparison, PNGM model shows obvious group character than RWP model.

Acknowledgements

This author would like to thank DFH satellite Co., Ltd. for the support to this paper.

REFERENCES

[1] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Pocket Switched Networks: Real-world mobility and its consequences for opportunistic forwarding," Technical Report UCAM-CL-TR-617, University of Cambridge, Computer Laboratory, February 2005.

[2] J Broch, DA Maltz, DB Johnson, YC Hu, J Jetcheva," Multi-hop wireless ad hoc network routing protocols", in Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 1998), pages 85-97, 1998.

[3] Y. Ko and N.H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," in Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom), pages 66-75, 1998.

[4] M. E. J. Newman, Detecting community structure in networks. Eur. Phys. J. B, 38, 321–330 (2004)..

[5] E. de Silva and M. Stumpf, "Complex networks and simple models in Biology", J. R. Soc. Interface, 2 (2005), pp. 419-430..

[6] N. Przulj, D. G. Corneil, and I. Jurisica, "Modeling interactome:Scale- free or geometric?", Bioinformatics, 20 (2004), pp. 3508-3515..

[7] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks", Nature, 393 (1998), pp. 440-442.

[8] M.E.J. Newman and M. Girvan. "Finding and evaluaing community structure in networks". Physical Review E,68,2003.

[9] Mirco Musolesi, Cecilia Mascolo, "A community based mobility model for ad hoc network research", in Proceedings of the 2nd international Workshop on Multi-Hop Ad Hoc Networks: From theory To Reality (REALMAN '06), Florence, Italy, May 2006, pp 31-38.

[10] R.L. Knoblauch, M.T. Pietrucha, M. Nitzburg. Field studies of pedestrian walking speed and start-up time. Transportation Research Board Records No. 1538, 1996.

[11] Newman, M. E. J. Fast algorithm for detecting community structure in networks. Phys. Rev. E 69,066133 (2004)

[12] Girvan, M. and Newman, M. E. J., Community structure in social and biological networks, Proc. Natl. Acad.Sci. USA 99, 8271–8276 (2002).

[13] L. Henderson. The statistics of crowd fluids. Nature, Vol. no. 229, 381-383, 1971.

[14] Finnis, K.K. and Walton, D. Field observations of factors influencing walking speeds. Ergonomics, 2006.

[15] LaPlante, John and Kaeser, Thomas P. "A History of Pedestrian Signal Walking Speed Assumptions", in Proceedings of the 3rd Urban Street Symposium: Uptown, Downtown, or Small Town: Designing Urban Streets That Work, 2007.

AUTHORS

GuoDong KANG is an telecommunication engineer of DFH Satellite Co., Ltd., 100094, Beijing, China.

Guoliang KANG is an PhD student of University of Technology, Sydney 15 Broadway, Ultimo NSW 2007

AUTHOR INDEX