Mobile P2Ping: A Super-Peer based Structured P2P System Using a Fleet of City Buses

Boon-Chong Seet
Singapore-MIT Alliance

***Abstract*— Recently, researchers have introduced the notion of super-peers to improve signaling efficiency as well as lookup performance of peer-to-peer (P2P) systems. In a separate development, recent works on applications of mobile ad hoc networks (MANET) have seen several proposals on utilizing mobile fleets such as city buses to deploy a mobile backbone infrastructure for communication and Internet access in a metropolitan environment. This paper further explores the possibility of deploying P2P applications such as content sharing and distributed computing, over this mobile backbone infrastructure. Specifically, we study how city buses may be deployed as a mobile system of super-peers. We discuss the main motivations behind our proposal, and outline in detail the design of a super-peer based structured P2P system using a fleet of city buses.**

Index Terms **—Mobile Peer-to-Peer, Super-Peers, Mobile Ad Hoc Networks, City Buses.**

I. Introduction

P2P computing has gained significant attention from both industry and research communities in recent years. A key attraction of P2P systems is their ability to scale without requiring expensive and powerful servers. The reason is because P2P systems work by distributing the functionality and harnessing the resources across a large number of independent peers. In addition to having high scalability, such systems are also inherently robust and fault tolerance since there is no centralized server, and the network is inherently self-organized. Today, the P2P technology has been widely embraced by the Internet users, and has seen successful applications in areas of digital content sharing, distributed computing and collaboration, distributed storage, and many more. With the advent of wireless technology, the number of mobile users has increased tremendously over the years. Greater attention has therefore been paid to tackling the challenges of P2P computing in the mobile environment. Among the key challenges include: relatively unstable and variable mobile connectivity; heterogeneity and limited resources of mobile devices, such as in operating power, storage, and processing speed.

Boon-Chong Seet is with the Singapore-MIT Alliance, N4-B3C-14, Nanyang Avenue, Singapore 639798 (phone: 65-6790-6953; fax: 65-6792-6559; e-mail: smasbc@ nus.edu.sg).

The recent availability of affordable wireless network devices such as handhelds with Wireless LAN/Bluetooth connectivity has further fueled the pace of innovations in wireless networking research. A particular development of interest to this paper is the recent works on applications of MANET – an all-wireless and decentralized multi-hop network, which has seen several proposals, e.g. [1-3], on utilizing mobile fleets such as city buses to deploy a mobile backbone infrastructure for communication and Internet access in a metropolitan environment. This paper further explores the possibility of deploying P2P applications such as content sharing and distributed computing, over this mobile backbone infrastructure. Specifically, we study how city buses may be deployed as a mobile system of super-peers. We discuss the main motivations behind our proposal, and outline in detail the design of a super-peer based structured P2P system using a fleet of city buses.

The remainder of this paper is organized as follows. Section II presents the background and overview of our work in this paper. We focus our description on two most relevant areas: i) Super-peer networks; and ii) MANET based on mobile fleets such as city buses. For more general introduction to P2P computing and MANET, the reader may wish to refer to [22] and [23], respectively. In Section III, the motivations for our work are discussed, and a detailed design of our proposed system is presented in Section IV. Finally, Section V concludes the paper with a summary and some directions for future work.

II. Background and Overview

Until recently, Internet P2P systems assumed all peers are equal and uniform in resources. Functionality is thus distributed without considering real-world heterogeneity of peer capabilities. For example, some peers may have smaller disk and slower processor speed than others. However, they perform the same role and responsibility as other peers with greater capabilities. This results in instances of inefficiency and bottlenecks in performance due to very limited capabilities of these peers. To account for and even exploit the existence of such heterogeneity of peer capabilities, the notion of super-peers, or nodes which are more well-provisioned in terms of resource capacity, have recently been introduced [5].

Super-peers take on a greater role and responsibility among participating peers. A super-peer often plays the role of a server that manages the queries and responses for a subset of ordinary peers. This super-peer in turn, is connected to other super-peers in a pure P2P way. As a result of the clustering of heterogeneous devices and elevating certain well-provisioned nodes to the role of super-peers, the impact of inefficiency and performance bottlenecks presented by some ordinary peers can be minimized. Kazaa [6] and newer releases of Gnutella [7] are popular industry P2P systems that have adopted the super-peer architecture into their designs for improved efficiency and performance.

Mobile fleets such as buses can be wireless-enabled (retrofitted with radio transceivers), and along with an on-board computer running an appropriate routing scheme, these buses are able to inter-communicate and serve as mobile routers for other nodes connected to them. Unlike private vehicles whose direction of travel may not be known a priori and may travel at high speed, buses travel regularly along their pre-determined routes, leave and return to depot in predictable times, and travel at lower speed. The resulting high regularity and predictability of their traveling pattern, coupled with low connectivity outage by virtue of their lower mobility and possibly longer transmission range are characteristics that can be used to establish a mobile routing backbone that exhibits both high stability and reliability.

This paper proposes to exploit this underlying stable routing infrastructure for deployment of P2P applications in a mobile environment. We further propose that buses should be a natural choice for super-peers since they are less constrained by size than other mobile devices and thus can serve as a host with higher system capacity.

III. MOTIVATIONS

A. *Extensive Bus Network and High Ridership*

Public buses are one of the dominant modes of public road transport in many parts in the world. Especially in metropolises such as Tokyo and Singapore, where the land is relatively scarce and the city is densely populated, public buses offer a means to reduce the use of private transportation and the need for more land for roadway construction. In most of these cities, an extensive bus network with services linking to virtually every corner of the city is already in place.

We can take the case of Singapore as an example. In this city-state, the main bus operator operates around 190 bus services (150 trunk and 40 feeder services) with more than 2500 buses daily [8]. This gives about 13 buses on the road for each bus service. The buses serve a total of 35 bus terminals/interchanges located in major residential, commercial and industrial estates in the city. Virtually all households are within 400m, or a 5min walk away from a bus stop. A ridership of 2.4 million passenger trips a day has been recorded [8], representing more than half the city's population of 4 million. These statistics show that the deployment of a mobile backbone for communication using city buses is generally feasible, given the relatively dense network of buses, and also their easy accessibility by the mass population.

B. *Useful Bus Characteristics*

As mentioned in Section II, the buses exhibit unique characteristics that can be exploited in particular for mobility management. These characteristics include high regularity and predictability of traveling pattern as well as lower mobility relative to other road vehicles such as cars and taxis. In addition, recent developments in Intelligent Transport Systems (ITS), especially in areas of positioning systems such as GPS-based Vehicle Location System [9] for bus fleet management, are potential solutions that can be harnessed to augment the routing scalability[1] of the underlying mobile backbone.

Buses are also inherently less constrained than mobile end devices in terms of battery power and computational resources. Therefore if P2P applications are deployed over the mobile backbone, we can also exploit this difference in capabilities between end-devices and buses to task the latter with a greater role as super-peers.

C. *Support for New Services*

1) "P2Ping": Apart from providing good transport service, it is in the bus operator's interest to also explore ways to make commuters feel enjoyable during their bus journey. One possible way is to offer value-added services such as peer-to-peer chat and music sharing[2] between bus commuters. The buses when acting as super-peers can provide, for instance, an efficient lookup service for commuters to lookup available chat peers (like a name service), or some music files of interest shared by other commuters.

2) Gaming: Mobile gaming is another potential service. It is long known that a significant challenge to game development for mobile terminals is their limited capacities, which ultimately restricts the games to, e.g. simple text-based games with little or no interactivity. To allow for more sophisticated media-rich gaming with possibly multiple players, a much higher computation power is necessary. This may be achieved by combining resources of many mobile devices to perform a *common* computation-intensive task such as mobile gaming. This is the fundamental concept of *wireless grid computing*, which may be adapted for mobile P2P. The super-peer architecture in particular, is well-suited to implement a proxy cluster-based grid service interface for peer-resource discovery (lookup) and job distribution [12]. The buses being super-peers for example, can maintain an index of commuters' device capacity (such as memory, processor speed, battery level) and

[1] Position-based routing [10] is known to be scalable due to its only requirement for local communication between neighbors.

[2] Clearly this has to be legitimate and copyright-compliance [11].

compose a processing grid in which these devices or peers, work together as a single virtual high-capacity computer to support more powerful gaming on the move.

3) Value-added search: With a high ridership, the bus network can also be a potential distribution channel for content producers. For example, the buses can use their role as super-peers to profile the interests of commuters based on their search queries, and provide in addition to their search results, some value-added information such as the availability of similar new content that they may be interested in, or even let them sample the new content like excerpts of a new song, music clips, movie teasers, fully functional trial-versions of new software, games, e-books, etc. from selected content producers and maybe hosted by the buses. In turn, content producers can receive feedback such as popularity of a specific content for their market research based on its frequency of download by the bus commuters.

IV. SYSTEM DESIGN

A. *Model and Assumptions*

Our system is composed of three network entities: the i) commuters (ordinary peers); ii) buses (super-peers); and iii) bus depots (forwarding gateways). We adopt the single term “depot” to refer to both bus terminals and interchanges for ease of reference. Only commuters and buses may originate or receive traffic. The depots in our system are simply transit gateways to facilitate forwarding of messages between buses/commuters. In the following, we elaborate on the assumptions of our bus service and communication models.

1) Bus Service Model: Buses are grouped according to their bus service numbers. A bus service is operated by a fleet of buses plying along a specific route. Each bus is uniquely identified by a bus node ID, and a bus service number. Buses of the same fleet leave and return to depot one after another. Multiple buses of the same fleet may be on the road at the same time at different locations along the bus route. Multiple depots are located across the metropolitan area. Each depot is also uniquely identified by a depot ID. A bus service may start and terminate at the same or different depots. Two most common types of bus services are considered. The first is a trunk service, in which a bus starts from one depot and ends at another depot. The second is a feeder (or loop) service, in which a bus leaves and returns to the same depot. An illustration of the bus service model is shown in Fig. 1.

2) Communication Model: With the exception of depot-to-depot communication, in which we assume the depots can communicate with each other over a wired infrastructure, all other communications between entities such as between bus-commuter, bus-bus, and bus-depot are assumed by means of radio transmission. However, we use two different

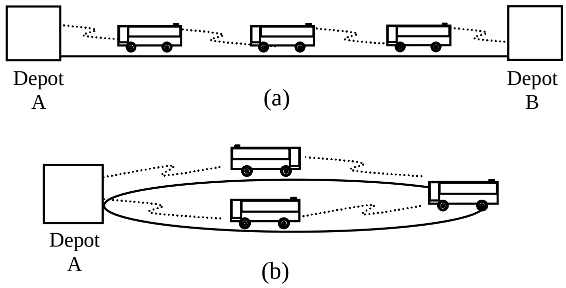

Fig. 1. Bus service model: (a) Trunk; and (b) Feeder (or loop); service.

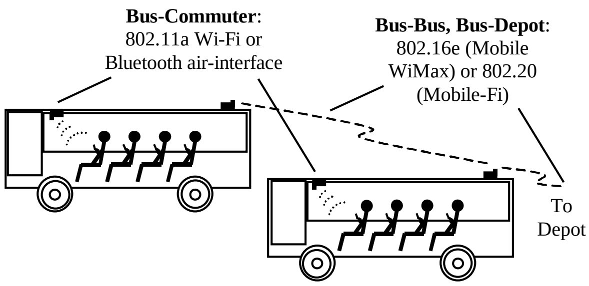

Fig. 2. Air interfaces for bus-commuter, bus-bus, and bus-depot communication.

transmission ranges for our system as shown in Fig. 2. A short-range radio based on open standards such as IEEE 802.11 Wi-Fi [13] or Bluetooth [14] is used for internal communication between the bus and its on-board commuters. On other hand, the buses can communicate mutually and with depots using longer-range radios as they are less constrained by size and power than the commuter devices. Candidate air interface for bus-bus, bus-depot communication are IEEE 802.16e and 802.20. Note that each bus therefore has two interfaces (short and long range). 802.16*e* is a mobility extension of 802.16 WiMax standard for fixed wireless access in a metro-area (with mesh ad-hoc networking option). Transmission range is between 2-5 km, non-line of sight, and supports up to 15 Mbps for vehicular speeds in access of 100 km/hr. Further details including that of 802.11a for Wireless LAN are given as shown in Table I.

If a bus-bus or bus-depot pair is in the range of each other, they are considered *neighbors*. On the other hand, communication between distant entities may require some intermediate nodes to serve as relayers, which thus can occur over multiple wireless hops, or through depots over the wired infrastructure. Given the relatively long range of the radio used for bus-bus, bus-depot communication, we also make the assumption that buses of the same fleet (or bus service) are connected in a way resembling a node string between their starting and ending depot (Fig.1). For the purpose of message forwarding using an underlying geographic forwarding scheme, each bus and depot is also assume to know its own physical location. For depots, their (stationary) location may be pre-determined. For the buses, however, they may track their own real-time location using GPS or by other means such as RF signpost transmitters.

TABLE I
FEATURES OF 802.11A AND 802.11E

	802.11a	802.11e
Frequency band	5 GHz	2-6 GHz
Access type	Local area access	Metropolitan area access
Range	Sub-100 m	2-5 km
MAC protocol	CSMA/CA	Dynamic TDMA
Bit rate	Up to 54 Mbps in 20 MHz channel bandwidth	Up to 15 Mbps in 5 MHz channel bandwidth
Non-line of sight	No	Yes
Mobility support	Pedestrian to low vehicular speed	Higher vehicular speed

B. Super-Peers in Structured P2P

In our system, the buses function as super-peers and commuters connect to the buses on which they ride as ordinary peers. As in a normal super-peer operation, the commuters query the bus to locate objects of their interest and receive the results from it. The objects can be a list of files (e.g. for music sharing), profile of a person (e.g. for P2P chat), or a system description of commuter's device (e.g. for looking up resources for computing-intensive task such as mobile gaming). Objects are typically described by file name, keywords, or some metadata.

Existing super-peer systems such as Kazaa are mostly based on unstructured search for object lookup. Flooding is typically used to broadcast queries among the super-peers, which is known to be expensive, and the search may be prematurely terminated after exhausting the allowable number of hops, leading to unguaranteed lookups. Newer structured P2P systems based on distributed hash tables (DHT) including Chord [17], CAN [18], and Pastry [19], provide a more deterministic way of lookup and bounds on the lookup costs. In DHT, every object is mapped to some peer. A description (key) containing a link (value) to where the object can be found is then stored at the peer to which the object is mapped. Any object can easily be found by directing query to the peer responsible for storing the object location. Flooding can therefore be avoided.

However, current structured P2P systems have mostly not consider the heterogeneity of peer capabilities and may still suffer from bottlenecks due to very limited capabilities of some peers. Thus, it makes technical sense to adopt the super-peer concept for further improving the structured lookup performance by taking into account the real-world heterogeneity of P2P systems. The next section describes features of our object lookup algorithm in a super-peer based structured P2P system.

C. Object Publish, Withdraw, and Lookup

Here we assume each object is associated with a unique ID. Each bus maintains an index of its own commuters' objects. The index is composed of tuples of the form *<ObjectID, OwnerID>*, uniquely identifying each object and the commuter who owns the object. Each tuple is then published by the bus (on which the owner resides) to the super-peer network in the form of a quartet *<ObjectID, OwnerID, BusServiceNo, BusNodeID>*, where *BusServiceNo* and *BusNodeID* are the service number and node ID of the publishing bus, respectively. The super-peers responsible for storing this quartet are a group of buses operating the same bus service number. We map the *ObjectID* to a *bus service number* through a hash function, and all buses of the mapped bus service shall be responsible for storing the quartet. Note that unlike in normal DHT where objects are often mapped to individual nodes that can be very dynamic (especially in a mobile environment), we map to bus services, which can minimize the mapping disruption as a bus service can persist even as existing buses leave or new buses join the system or when they leave/return to their depot. In the following, we explain the object publish and withdraw operation:

- When a commuter node joins, e.g. when commuter gets on the bus or turn on his/her device during the journey, it uploads its object tuples to the bus. The bus in turn adds the tuples into its index and publishes the corresponding quartets to the super-peer network. If the commuter inserts/deletes an object, the object tuple is sent to the bus, which then adds/removes this tuple to/from its index and publishes/withdraws the corresponding quartet from the super-peer network. When a commuter node leaves, e.g. when commuter gets off the bus, or turn off his/her device during the journey, the bus removes the tuples owned by the commuter from its index and similarly withdraws the corresponding quartets from the super-peer network.
- In our system, we will also need to address the issue of bus joining and leaving the system. When a bus joins, e.g. when bus leaves its depot, it requests from an operating "sibling bus" (a bus from the same bus service), object quartets that it is required to maintain. Similarly, when a bus leaves, e.g. when bus returns to its depot, it withdraws the quartets owned by all its commuters from the super-peer network, as they are all leaving the bus at the depot. Alternatively, we can define a *TimeOut* field for the quartet, with value being the difference between the time at which the quartet is created and the expected arrival time of the bus at the ending depot. This allows the quartets to self-expire, saving the bus from issuing an explicit withdrawal.

Object lookup: When a commuter is looking for an object with id *ObjectID*, it sends its request to the bus. The bus then first searches its own index for the object. If a tuple with a matching *ObjectID* is found, i.e. the object is available from a fellow commuter on the bus, a reply is instantly returned to the querying commuter. Otherwise, the bus forwards the query to the bus service responsible for holding the quartet of this object. Recall that this can be known by simply hashing the *ObjectID*. When a bus of the responsible bus service receives the query, it finds the quartet and forwards the query to the bus node (identified by the *BusServiceNo* and *BusNodeID* of the quartet) in which the owner of the object is located. Upon receiving, the bus contacts the commuter matching the *OwnerID* of the quartet and returns the query result.

D. Message Forwarding

In our approach for message forwarding, we assume an underlying multi-hop routing protocol based on nodal position information. We shall also use depots as gateways to facilitate routing between the buses. We first describe the mechanism for beaconing and maintaining information for our forwarding tables. We then explain the procedure by which our messages are forwarded.

1) Beaconing: As commuters do not participate in message forwarding over the super-peer network, the description for beaconing here only involve the buses and depots. We assume each bus and depot beacons regularly its presence. The information in each beacon for the buses includes: *BusNodeID*, *BusServiceNo*, and *BusNode Location* (in terms of latitude and longitude), On the other hand, beacon from a depot includes *DepotID* and its physical location.

2) Maintaining Tables: When a beacon packet from a bus is received, the information is stored into a Neighbor Table (Table II) and later removed after a system-defined timeout value. Similarly, when a bus receives a beacon from a depot (either its own operating depot or a depot that the bus passes by), the information is stored into a Depot Table. However, a difference is that after timeout, the depot entry is not removed from the table, but simply indicated as NULL in its neighbor field as shown in Table III. If a new beacon from the same depot is later received, this field is reset to one. This is because unlike buses, depots are stationary, and therefore the buses may expire them only after a much longer duration, such as at the end of their day's operation. In addition, at each depot, an information database detailing which bus service operates at which depot is assumed available. For example in Table IV, the first entry shows that bus service 51 (a trunk service) operates between depots 8 and 15. On the other hand, bus service 179 only operates at depot 10 because it is a feeder (or loop) service.

TABLE II
NEIGHBOR TABLE TO STORE BUS NODE INFORMATION

BusService No.	BusNode ID	BusNode Location	
		Latitude	Longitude
179	2	N01 16.477	E103 42.315
199	3	N01 26.449	E103 57.398

TABLE III
DEPOT TABLE TO STORE DEPOT INFORMATION

DepotID	Depot Location		Neighbor Flag
	Latitude	Longitude	
10	N01 20.369	E103 42.348	1
12	N01 19.942	E103 44.544	0

TABLE IV
BUS DEPOT INFORMATION DATABASE

Bus Service No. (Trunk/Feeder)	DepotID	
51	8	15
179	10	--

3) Forwarding: We next describe how a message, e.g. an object query, object publish or withdraw request, can be forwarded to its intended destination. Recall from Section IV.C that, for an object query, the destination is not a specific bus or commuter node, but a bus service. In other words, the query can be forwarded to any bus of the destination bus service (i.e. anycast). Likewise, for object publish and withdraw request, we first forward the message to a bus of the destination bus service, after which the message is multicast to its "sibling" buses (or buses of same bus service). Multicasting between sibling buses is straightforward since the group membership, i.e. grouping of buses into bus services, is typically not dynamic and can be known by buses in advance. Current multicast schemes based on geographic information such as PBM [20] can be employed. Similarly, existing geographic-based unicast schemes like GPSR [21] can be used or adapted for subsequent forwarding of the object query and also the object itself to/from a specific bus node in which the requested object is located.

We now describe the process of how a message can be forwarded to a destination bus service, the pseudo-code of which is shown in Fig. 3. When a bus receives a message destined for a bus service that is different from its own, it looks up its Neighbor Table to first see if it has a neighbor, which is a bus of the destination bus service. If so, the message is forwarded immediately to this bus and our forwarding objective is accomplished. Otherwise, it looks up its Depot Table to see if it has a depot as its immediate neighbor (by looking at the neighbor flag in each table entry). If a neighboring depot exists, the bus forwards the message to it. Recall that in our system, the depot maintains a connection to at least a bus of all bus services that operate from it, as well as to other depots over a wired infrastructure. Therefore, when this depot receives the message, it can forward either immediately to the destination bus service if this service is operating from its depot, or forward to another depot (using the wired infrastructure) where the destination bus service is operating after looking up the depot information database

(Table IV). As soon as the message arrives at a bus of the destination bus service, the message can be forwarded as unicast or multicast (depending on message type) to its sibling bus node(s) as previously described.

However, if there is neither a neighboring bus of the destination bus service nor a neighboring depot, the bus shall select the closest known depot from its Depot Table (note that each bus will know about at least one depot: its own depot), to which it forwards the message using the underlying geographic forwarding scheme.

Bus-side:
Let r be a *bus* node receiving a message m for destination service s
Let N be the set of entries in r's neighbor table
Let P be the set of entries in r's depot table

Checks neighbor table of r
If ($\exists\, n_1 \in N$: n_1 is a bus node of service s AND closest to r)
  Forwards m to n_1 *// done*
  Return
Else *// if no matching bus service*
  Checks depot table of r
  If ($\exists\, p_1 \in P$: p_1 is a depot with flag set to 1 AND closest to r)
    Forwards m to p_1 *// m to be processed by depot-side*
    Return
  Else *// if no matching bus service or neighboring depot*
    Selects p_2 such that $\text{dist}(p_2, r) = \min\{\text{dist}(u, r) \mid u \in P\}$
    Forwards m to p_2 *// using underlying geographic*
  End If *// forwarding scheme*
End If

Depot-side:
Let v be a *depot* node receiving a message m for destination service s
Let G be the set of bus services operating in v

If ($s \in G$) *// if s is operating in v*
  Forwards m to a bus node of service s *// done*
  Return
Else *// otherwise*
  Forwards m to a depot node where s operates
  // using database and wired infrastructure
  Return
End If

Fig. 3. Pseudo-code for message forwarding to a destination bus service.

V. CONCLUSIONS AND FUTURE WORK

In brief, we summarize the main contribution of this work as follows: (i) proposal of a mobile system of super-peers using city buses; (ii) design of a structured lookup service for bus commuters to locate their objects of interest; (iii) design of a geographic-based message forwarding protocol optimized for the bus system environment. This is an initial step towards a new conceptual framework for a Mobile P2P system. As future work, a specification of the system will be formalized, and shall be evaluated through analysis, simulation and prototype implementation.

ACKNOWLEDGMENT

The author would like to thank the Centre for Advanced Information Systems in NTU, for their hospitality and use of their computing facilities to carry out this work.

REFERENCES

[1] K.-J. Wong, B.-S. Lee, B.-C. Seet, G. Liu, and L. Zhu, “BUSNet: Model and usage of regular traffic patterns in mobile ad hoc networks for inter-vehicular communications”, in *Proc. ICT 2003*, Thailand, Apr. 2003.

[2] J. G. Jetcheva, Y.-C. Hu, S. PalChaudhuri, A. K. Saha, and D. B. Johnson, “Design and evaluation of a metropolitan area multitier wireless ad hoc network architecture”, in *Proc. WMCSA*, Monterey, CA, Oct. 2003.

[3] T. Li, K . M. Chan, J. Liew, and W. Seah, “Mobile Internet access in BAS”, in *Proc. International Conference on Distributed Computing Systems Workshops (ICDCSW’04)*, Tokyo, Japan, Mar. 2004.

[4] IETF MANET WG charter: http://www.ietf.org/html.charters/manet-charter.html

[5] B. Yang, and H. Garcia-Molina, “Designing a super-peer network”, in *Proc. International Conference on Data Engineering (ICDE)*, Bangalore, India, Mar. 2003.

[6] KaZaA: http://www.kazaa.com

[7] T. Klingberg, and R. Manfredi, “Gnutella 0.6. RFC”, *The Gnutella RFC*, Jun. 2002.

[8] SBS Transit: http://www.sbstransit.com.sg

[9] S. Prakasam, and A. Wang, “Implementing vehicle location system for public buses in Singapore”, *Journal of the Institute of Engineers*, Vol. 44, No. 2, 2004.

[10] I. Stojmenovic, “Position-based routing in ad hoc networks”, *IEEE Communications Magazine*, Vol. 40, No. 7, Jul. 2002.

[11] T. Kalker, D. H. J. Epema, P. H. Hartel, R. L. Lagendijk, and M. Van Steen, “Music2Share – Copyright-compliance music sharing in P2P systems”, *IEEE Proceedings*, Vol. 92, No. 6, Jun. 2004.

[12] T. Phan, L. Huang, and C. Dulan, “Challenge: Integrating mobile wireless devices into the computational grid”, in *Proc. ACM MOBICOM*, Atlanta, GA, Sep. 2002.

[13] IEEE P802.11: http://grouper.ieee.org/groups/802/11/

[14] Bluetooth: http://www.bluetooth.com/

[15] IEEE 802.16 Task Group ‘*e*’: http://www.ieee802.org/16/tge/

[16] IEEE 802.20: http://grouper.ieee.org/groups/802/20/

[17] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, “Chord: A scalable peer-to-peer lookup service for Internet applications”, in *Proc. ACM SIGCOMM*, San Diego, CA, Aug. 2001.

[18] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, “A scalable content-addressable network”, in *Proc. ACM SIGCOMM*, San Diego, CA, Aug. 2001.

[19] A. I. T. Rowstron, and P. Druschel, “Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems”, in *Proc. International Conference on Distributed System Platforms*, Heidelberg, Germany, 2001.

[20] M. Mauve, H. Fußler, J. Widmer, and T. Lang, “Position-based multicast routing for mobile ad-hoc networks”, *ACM SIGMOBILE Mobile Computer and Communications Review*, Vol. 7, No. 3, 2003.

[21] B. N. Karp, and H. T. Kung, “GPSR: Greedy perimeter stateless routing for wireless networks”, in *Proc. ACM MobiCom*, Boston, MA, Aug. 2000.

[22] A. Oram (ed.), *Peer-to-Peer : Harnessing the power of disruptive technologies*. O'Reilly, 2001.

[23] J. P. Macker, M. S. Corson, "Mobile ad hoc networking and the IETF", *ACM Mobile Computing and Communications Review*, Vol. 2, Number 1, Jan. 1998.