XCast: A Personal and Groupwise Broadcasting System for Social Event Networking

by

Sung-Hyuck Lee

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
in partial fulfillment of the requirements for the degree of

Master of Science in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2008

Author .....
Program in Media Arts and Sciences,
September 2008

Certified by.....
Andrew B. Lippman
Senior Research Scientist,
Program in Media Arts and Sciences,
Thesis Supervisor

Accepted by.....
Deb Roy
Chairperson
Media Arts and Sciences Committee
Program in Media Arts and Sciences

XCast: A Personal and Groupwise Broadcasting System for Social Event Networking

by

Sung-Hyuck Lee

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
on September, 2008, in partial fulfillment of the
requirements for the degree of
Master of Science in Media Arts and Sciences

ABSTRACT

This thesis describes the design and development of a system that is aimed for personalized and group-wise broadcasts to collaboratively distribute information and to coordinate locally created events in infrastructure-free milieu. This system, called XCast, has two targets: One is to permit people to create personalized communicators, "broadcast stations" over mobile devices, for extemporaneous events or individually motivated presentations. The other is to provide people with a cognitive platform for social event awareness that informs what is happening around them and then timely coordinates the events. This system applies wireless/mobile peer-to-peer networking technologies, such as 802.11 ad-hoc and mesh networking.

To achieve the targets, in the thesis, we focus on newly designing architecture of the cognitive platform and then developing a robust and agile protocol which makes it possible for the platform to reliably work in wireless ad-hoc/mesh networks. The former work is to build a group of broadcast stations into a distributed crystal-gazing system to be aware seasonably of what is going on in our surroundings. With respect to the latter, we propose a distributed protocol, dubbed "Broadcast Resource Schedule Protocol (BRSP)." It has broadcast operations more reliable and scalable in wireless ad-hoc/mesh networks via synchronization and neighbor knowledge schemes. In the end, the BRSP evolves a wireless peer-to-peer network into a cognitive network to support the platform.

This system offers a riper breeding ground for creation of a platform for social event networking and of cooperative media for a local community. The value of this is in considering community networks that are matrices of social collaboration, rather than point connections, as well as sources of novel civic media initiated by grassroots.

Thesis Supervisor: Andrew B. Lippman
Title: Senior Research Scientist, MIT Media Laboratory

XCast: A Personal and Groupwise Broadcasting System for Social Event Networking

by

Sung-Hyuck Lee

The following people served as readers to this thesis:

Thesis Advisor: _____

Andrew B. Lippman
Senior Research Scientist,
MIT Media Laboratory
Thesis Supervisor

Thesis Reader: _____

David P. Reed
Adjunct Professor,
MIT Media Laboratory

Thesis Reader: _____

Christopher M. Schmandt
Principal Research Scientist
M.I.T. Media Laboratory

ACKNOWLEDGMENTS

First of all, I would like to thank my advisor Andrew Lippman who gave me opportunities to be a member at MIT Media Lab, and to be amused on a technology playground for inventing a better future, without a glitch. I especially thank him for believing in me and for providing me with an atomic idea to be a seed project for this thesis. I am also thankful for him helping me out in numerous ways with my research and in my development of XCast system. I am truly grateful to have an opportunity to further realize the vitalness of linguistic senses from his speech of passionate eloquence.

Also, I would like to thank my thesis readers, David Reed and Christopher Schmandt who gave me a lot of encouragement and helped me to think about the direction of my thesis topic. First, I thank David, my another advisor at Viral Communications group, for giving me ideas on the virtual Jukebox, which was the most essential scenario and application of XCast system in the thesis. As always, I respect him for his unceasing passion on communications and networking research, for constantly producing stunning ideas, and for encouraging me to be toward such passionate behaviors.

When I first asked Chris to be my thesis reader, I could feel his acute insight to penetrate ideas and directions of the thesis. His comments always let me take right steps. This especially helped me refine my thoughts and my project in many ways. I would really like to appreciate his efforts. I hope to have better opportunities to interact with him, again, in the near future.

I would truly like to thank William Mitchell, Mitchel Resnick, Ellen Hume, Ingeborg Endter, and Michail Bletsas for their encouragements, comments, and supports on XCast project. Especially, I thank Bill for his warm considerations and encouragements. I had opportunities to further interact with him through the City Car project and also to apply the XCast system into the City Car project. I would like to appreciate him to show me a

new world. I thank Mitchel, Ellen, and Ingeborg for their wholehearted comments on the XCast project for the Future Civic Media. They motivated me to join the Center for Future Civic Media, and I often had discussions and brainstorms about my research with them. This collaboration helped me to further develop my ideas as well to ferret out interesting scenarios for the XCast project. I would also like to appreciate Michail for providing me with OLPC XOs for the implementations and the experiments.

I am so grateful to have an opportunity to work with colleagues at different groups of the Media Lab, Junki Lee (Personal Robotic group), Sanghoon Lee (Physical Language Workshop Group), and Leonard Bonanni (Tangible Interfaces Group) for their efforts on developing the first pilot testbed. Especially, I would like to thank the Media Lab affiliates, Donghoon Lee (Samsung), and Simon Ji (LG) for donating experimental devices.

I would also like to thank my colleagues at Viral Communications group, Kwan Hong Lee, Fulu Li, Durga Pandey, Dimitris Vyzovitis, Grace Woo, Polychronis Ypodimatopoulos, Dawei Shen, Nadav Aharony, David Gauthier, Nick Knol, and Quingchun Ren. They often motivated me to produce fruitful ideas in future ommunication areas. Particularly, I wholeheartedly thank Kwan Hoong, David, Nick, and Qingchun, for their supports and implementations on the XCast system.

I cannot end without thanking my family, on whose constant encouragement and love I have relied. Their unflinching courage and conviction will always inspire me, and I hope to continue, in my own small way, the noble mission to which they gave their lives. I dedicate this thesis to my family and my wife Mi Young.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Primary Motive:

How are we seasonably aware of what is happening around us?

CHAPTER 1. INTRODUCTION

Wireless/mobile peer-to-peer (P2P) communication and networking technologies have provided a possibility to explore novel avenues of communications in local areas. In the distributed system, power and intelligence are evenly shared among all the entities that form a community network. It has led to unprecedented aspects in cultural and economic facets even if they are still immature.

Peer-to-peer connections in wireless networks are naturally localized due to the limitations of coverage of radio signals. It rather provides, however, its users with opportunities to be able to have a good awareness of locality. In such mobile distributed systems, our attentions to the locality are undoubtedly moved to socializing with locals regardless of what type of spaces they are located in. In other words, the systems that allow their users to have direct interactions with each other without any help of infrastructural nodes lay the groundwork for building a platform of localized social networking.

Broadcasting is one of the most effective methods in communications. Its natural traits of radio signal radiation enable fast deliveries and simultaneous accesses to multi-users in wireless networks. The traditional public broadcasting systems such as television and radio, however, are asymmetric systems where the privileges are concentrated in a central antenna or transmitting station. In other words, these systems have not served any method for their watchers or listeners to proactively respond to the broadcast contents.

On the other hand, Internet has changed this a bit with multicasting/broadcasting, where a set of channels is available to all and the asymmetry is more dynamic. That is, receivers are still receivers and transmitters are still points of origin, but they can change places periodically. In result, it enables us to build a more dynamic model where everyone is a broadcaster on a moment-to-moment basis and the channels used by them are shared in real-time manners.

Furthermore, the wireless/mobile P2P communication system provided its users with opportunities to personalize radio resources, while making them remind of the importance of local societies. That is, the system enables the users to transfer their locals what they want to share at any time. The use of broadcasting in wireless/mobile P2P networks makes the broadcast system itself personalized, while having benefits from the characteristics of group-wise connections. This thesis explores the ways of how a broadcast system can be personalized to play a pivotal role of a local communicator in wireless/mobile P2P networks. It will be performed in the context of XCast project [1], which is about a personal and group-wise broadcasting system to collaboratively distribute information in Wi-Fi ad-hoc/mesh networks. Ultimately, this thesis describes how such the broadcasting system offers a riper breeding ground for a platform supporting social networking driven by events arising in local areas.

1.1. Problems

This section defines the problems we would like to figure out in this thesis. The approaches we use to define the problems are as follows. At first, we have observed ways people unwittingly behave in the daily lives and then found some unreasonable commitments we need to correct. Next, we researched what is needed to proficiently deal with such findings in cultural and behavioral angles. Afterward, we scrutinized technological methodologies to support the applicability of the solution that helps release the distresses.

1.1.1 Inexistence of local communicators

We prefer a lifestyle that involves staying at a fixed domicile rather than a vagabond life. Such a life style has made us mainly have interactions with locals. As traditionally perceiving "proximity is communicability," however, we have overlooked the needs of communicators fit for such the local interactions. Such the recognition has had effects on

the direction of development of communication systems: It resulted in gargantuan proliferation (or even full maturation in some areas) of research and development on long-haul communication systems, while the systems supporting for local communicators were still being quite primitive. The imbalance caused the complexity and costliness of the local communication systems. For instance, the communication gadgets that we make daily use of even when to communicate with our friends or colleagues in the same building or campus should be always first connected to central node located in the infrastructure before reaching our targets, notwithstanding existence of direct paths. This means that the resources that we make use of for local communications can be highly lavish due to the architecture centered to remote communications.

The advent of Wireless LAN (WLAN), known as IEEE 802.11 series, in wireless communications gave rise to possibilities of wireless/mobile P2P communications and networking dubbed 802.11 ad-hoc and mesh networking. In other words, it provides some solutions on problems of the structural complexity and costliness caused by the infrastructure-based local communication systems. The system, for example, allows its users to connect directly to their locals without any help of intermediate or core nodes in the infrastructure networks. Also, the extrication from the infrastructure led to novel first mile networks as the cost-free spaces for communications in the economic facet. Nevertheless the considerable efforts of research and development of the systems, however, any killer application suited for wireless/mobile P2P networking or any platform supporting for various applications has not appropriately been created or burgeoned. It has resulted in failing to transform the ad-hoc/mesh networking system to an authentic local communicator so far.

1.1.2 Awareness of local events

An issue that we readily ferret out in the information report context is the unbalance between quantities of hard news and soft news. In other words, the news has always covered subjects that catch people's attention and differ from their "ordinary lives." The

news is often used for escapism and thus normal events are not newsworthy [2]. We may find the reason from the intellectual curiosity that we hanker to be aware of what we have never experienced rather than what we already tasted. Whether the subject is love, birth, weather, or crime, journalists' tastes inevitably run toward the unnatural and the extraordinary. Newspapers and news channels of radio or television broadcasting, for instance, normally give much detail on hard news stories, such as those pertaining to murders, fires, wars, and so on while sacrificing other, decidedly less global stories.

Under such the circumstance, however, frequently we do not acquire any timely information of what is going on in our surroundings such as at buildings, at campuses, and even in local communities where we daily stay, while having an instant access to news happening in the most far-flung corners of the globe, at any time, via radio, television, Internet, and even cell phones. This has made people be really up the creep, especially, in emergency cases rising in highly populated places, for example, a US shooting rampage at the Virginia Tech University [3]. Unambiguously, we could appropriately deal with a myriad of local events including emergency situations if being able to know what's happening around us in real-time. It results in posing the problem on the awareness of local events: People are not seasonably aware of what is happening around them.

As described in the previous section, the wireless/mobile P2P systems such as 802.11 ad-hoc and mesh networking have the potential intelligence to play a pivotal role of a local communicator. It depicts that the systems could be media to collect extemporaneous yet crucial events arising in the ordinary lives and to timely inform their users of the events. The current systems of 802.11 ad-hoc and mesh networking, however, have not been systematically designed to ferret out, gather, and notify the events cohesively with the peer nodes. It is just a congregation of individual end terminals that transfer or sometimes route data packets among the nodes rather than a set of socialized networks that serve as sources for local events.

1.1.3 Broadcasting jumble

This section addresses the characteristics broadcasting has inherently and the issues multiple streams of broadcasting give rise to in wireless infrastructure-free milieu.

1.1.3.1 Broadcasting in infrastructure-free networks

The broadcast is largely spontaneous in the wireless networks [4]. It is caused by the attribute that the broadcasting operation is wrought by a flooding originated from a mobile node in non-cognitive wireless networks, at any time it wants. That is, it is not easy to previously gather any kind of global topology knowledge, especially in wireless/mobile P2P networks, due to the reasons such as the host mobility and the lack of synchronization before the transmission. It makes the broadcast unmanageable, sometimes.

The broadcast is unreliable. No acknowledgement mechanism is used as well. The reason is that acknowledgements may cause serious medium contention surrounding the senders, and in many applications (e.g., the route discovery in multi-hop routing protocols [5] [6] [7] [8]), a perfectly reliable broadcast is unnecessary. Any attempt to adopt the acknowledgement method, however, needs to be made to infallibly distribute a broadcast message to as many hosts as possible, without paying too much effort. The motivation to make such a trial is that a mobile node may miss a broadcast message because of being off-line, it is temporarily isolated from the network, or it experiences repetitive collisions.

1.1.3.2 Blind Broadcasting

The philosophy that the wireless/mobile P2P networking systems independently work from the public infrastructure systems provides their users with a stepping stone to a personalized communication system. The personalization is achieved by that the users are allowed to manage wireless radio channels, without a glitch, for the purpose of communications. In the broadcast context, the systems enable the users to have their own

(personalized) broadcasting systems to share with their local targets whatever they want at any time.

However, a problem arises if there exist simultaneously multiple streams of broadcasting over just one wireless channel in such the personalized broadcasting system. That is, broadcasting packets would often be collided or dropped provided that multiple nodes at the same time broadcast over the same channel, *without "any cognition"* of each other: We call it *"Blind Broadcasting."* By the philosophy of infrastructure-free, the wireless P2P networking systems have not a centralized administrator to control the broadcast but rather distribute all powers and intelligence evenly to all members in the networks. In even the state-of-art systems, moreover, a mobile node can not be exactly aware of what is happening on other nodes although they are within the range of the same radio signal. Actually, the nodes are not really networked in a cognitive manner yet. The increase of the Blind Broadcasting causes serious redundancy, contention, and collision to make performance of the wireless/mobile P2P networks severely degraded or the system itself even freeze. In the end, the Blind Broadcasting turns into *"broadcasting jumble."*

1.1.3.3 Channel hopping

The Blind Broadcasting results in bringing about the similar effect to co-channel interference. It suggests that the wireless channel-hopping may be a solution to avoid a myriad of the blind flooding caused by the co-location issue only if target receiving nodes, with the broadcast transmitter, can be moved simultaneously to one of other available channels. The following paragraphs represent, however, that in 802.11 systems, finding and hopping to a clean channel for data transmission would not be an appropriate approach.

The 802.11b or 802.11g standards, broadly deployed in the real networks and products, transmit their signal in a narrow radio frequency range of 2.4 GHz. The 2.4 GHz Wi-Fi signal range is divided into a number of smaller bands or "channels," similar to television channels. Any of the Wi-Fi channels 1 to 11, in the United States and Canada for

example, can be chosen when setting up a wireless LAN. Setting this WiFi channel number appropriately provides one way to avoid sources of wireless interference. Unlike television channels, however, some Wi-Fi channel numbers overlap with each other, known as "adjacent channel interference." Channel 1 uses the lowest frequency band and each subsequent channel increases the frequency slightly. Therefore, the further apart two channel numbers are, the less the degree of overlap and likelihood of interference. As well known, practically the 2.4GHz space only supports three non-overlap channels, channels 1, 6, and 11 [9]. With a total of three channels, we can now see that, in 802.11 b/g ad-hoc and mesh networking, eliminating the co-channel interference would be a challenge, while removing the adjacent interference is simply not possible. It means that even with the channel-hopping method, the Blind Broadcasting in 802.11 b/g ad-hoc and mesh networking systems is unavoidable.

At speeds of 54 Mbps and greater, 802.11a is faster than any other unlicensed solution. The 5 GHz band in which it operates is not highly populated, so there is less congestion to cause interference or signal contention. In other words, the eight non-overlapping channels allow for a highly scalable and flexible installation. This higher frequency compared to 802.11b, however, shortens the range of 802.11a networks. The higher frequency also means that 802.11a signals have more difficulty penetrating walls and other obstructions [10]. Unfortunately, in addition to its expensive cost, such the problem has prevented it from being spaciously deployed in the consumer market even though its adjacent interference immunity. In the co-location issue that the target nodes exist still within the bound of an overlapping-likely channel while broadcasting, 802.11a does not have an agile method to resolve that problem. Therefore, in the literature, we narrow our focus on 2.4GHz band.

Going back to 2.4GHz spectrum, 802.11n, the newest IEEE standard in the Wi-Fi category, was designed to improve on 802.11g in the amount of bandwidth supported by utilizing multiple wireless signals and antennas instead of one. This is referred to as multiple-input-multiple-output (MIMO) system. As such, a MIMO system introduces a bigger impact on interference to other consumer electronics devices though, such as

Bluetooth based devices, cordless phones, microwave ovens, and even existing 802.11b/g Wi-Fi networks. The impact comes from two challenges: The first, the wider bandwidth also means a reduced number of "clean" channels for other devices to operate in the same 2.4 GHz frequency (ISM) band, which is known as a co-location issue. Second challenge is increased bandwidth causing increased sideband effects that reduce the Signal to Noise Ratio (SNR). Again, the "available channels" are also very noisy because of the reduced SNR [11]. The upcoming 802.11n technology will result in posing a "killer" interference threat to proprietary 2.4 GHz wireless device peripheral applications. In this case, the Blind Broadcasting would deadly degrade the performance of 802.11n ad-hoc and mesh networks and may cause the system to even break down: It leads truly to "broadcasting jumble."

1.1.3.4 Virtual Carrier Sensing

In this section, we review RTS/CTS (Request to Send/Clear to Send) mechanism, named "virtual carrier sensing mechanism" in the 802.11 wireless networks and investigate how it works in the broadcast operation and tackles the problem on the blind broadcasting.

For the wireless ad-hoc network, the IEEE 802.11 standard specifies a basic access protocol called Distributed Coordination Function (DCF). It provides a contention free service based on carrier-sensing random access protocol, dubbed Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). RTS/CTS is an additional method to implement virtual carrier sensing in the CSMA/CA. It was adopted by the standard to reduce the frame collisions and as well to fix the performance degradation caused by two well-known problems, the hidden terminal problem and the exposed terminal problem the wireless medium generally suffers from, respectively.

Figure 1 shows a typical "hidden terminal" scenario. Let us assume that station B is within the transmitting ranges of both A and C, yet A and C can not hear each other. Also, let us assume that A is transmitting data packets to B. Provided that C has a frame to be transmitted to B, it first senses the medium and then finds itself free to access B,

according to the DCF, because of being not able to hear A's transmissions. C begins to transmit the frame, but this transmission will results in a collision at the station B owing to the frame sent by the station A.

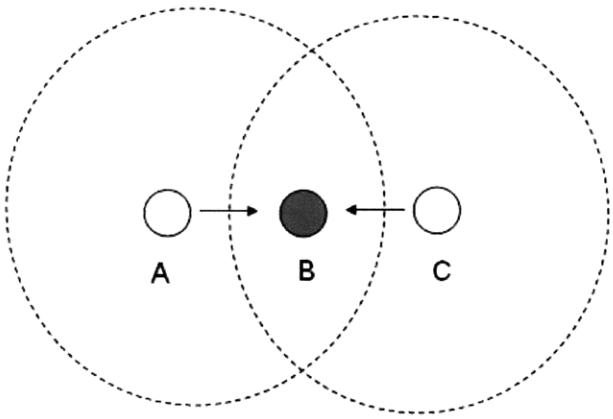


Figure 1. The hidden terminal problem

Figure 2 depicts a feasible scenario where the "exposed terminal" problem may occur. Let us suppose that both stations A and C can hear transmissions from station B, but the station A can not hear transmissions from station C, and vice versa. Also, let us assume that B is transmitting data frames to A, and that C has a frame to be transmitted to D. According to the DCF protocol, C senses the medium and finds it busy because of B's transmission. Therefore, C refrains itself from transmitting a frame to D even if this transmission would not cause a collision at A. The "exposed station" problem may thus result in a throughput reduction.

A node wishing to send data initiates the virtual carrier sensing process by first sending a RTS frame. If the intended receiver successfully receives the RTS frame and the channel is clear, it responds with a CTS frame. After receiving the CTS frame, the sender starts to transmit the packet. Any other node receiving the RTS or CTS frame should refrain from sending data for a given time to solve the hidden node problem. The amount of time the node should wait before trying to get access to the medium is included in both the RTS and the CTS frame. This protocol is designed under the assumption that all nodes have the same transmission range. If the packet size the node wants to transmit is larger than

the threshold, the RTS/CTS handshake gets triggered. If the packet size is equal to or less than threshold the data frame gets sent immediately without RTS/CTS dialogue. For the reference, RTS/CTS packet size threshold is 0 to 2347 octets. Typically, sending RTS frames is turned off by default (threshold ≥ 2347) [12].

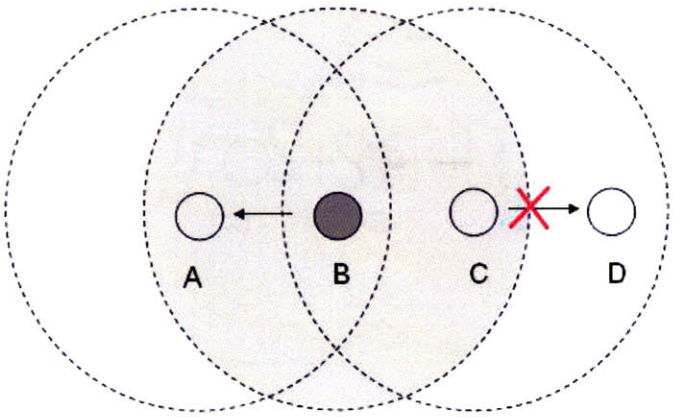


Figure 2. The exposed terminal problem

The RTS/CTS mechanism helps solve the exposed terminal problem only if the nodes are synchronized. When a node hears an RTS from a neighboring node, but not the corresponding CTS, that node can deduce that it is an exposed node and is permitted to transmit to other neighboring nodes. If the nodes are not synchronized the problem may occur that the sender will not hear the CTS or the ACK during the transmission of data of the second sender.

As described in [13] [14] [15], however, the RTS-CTS handshake fails to completely resolve all situations of the hidden terminal problem. Collision avoidance based on CTS fails if the hidden terminals do not receive the CTS correctly due to either collisions or packet reception errors. It could happen very often in a dense network that the transmitter does not receive CTS from all the intended receivers. This is because either the RTS was not successfully decoded by the receiver or the receiver is not allowed to send CTS as it might be a hidden terminal to another ongoing communication. Attempting to retransmit RTS to nodes who have not replied could create a very long delay.

Moreover, the increase in performance using RTS/CTS is the net result of introducing overhead (i.e., RTS/CTS frames) and reducing other overhead (i.e., fewer retransmissions). If there is no hidden node, then the use of RTS/CTS will only increase the amount of overhead, which reduces throughput. A slight hidden node problem may also result in performance degradation if RTS/CTS is implemented. In this case, the additional RTS/CTS frames cost more in terms of overhead than the gain by reducing retransmissions. Therefore, with this handshake process, collisions occur most likely among short RTS frames instead of among long data frames. Note that the RTS/CTS exchange may not be used when the data packet is short.

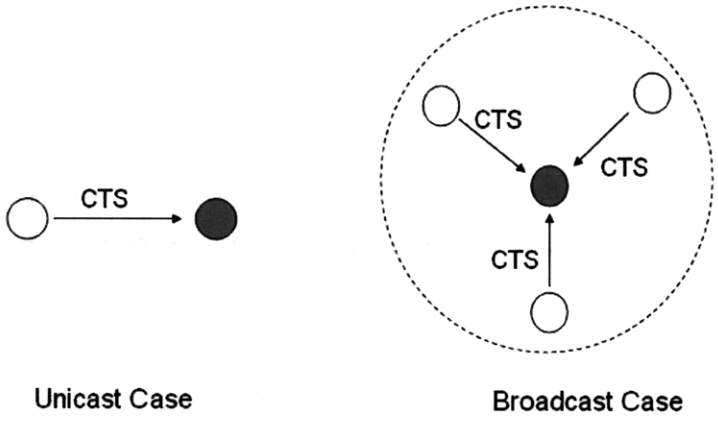


Figure 3. The CTS frame collision in the broadcast case. The gray nodes are initiating senders and the white nodes are target receivers.

Originally, RTS/CTS is designed fit for the unicast data transmission. Extending the handshake to a broadcast scenario is not straightforward. Having more than one potential receiver in the broadcast case creates a chance of collision among the CTS or ACK packets as shown Figure 3. If there are collisions among CTSs, the sender will not be able to successfully receive the CTS frames and thus the handshake process fails. In addition, the RTS/CTS weakness introduced in the previous paragraph becomes worse in a broadcast scenario. The IEEE 802.11 standard has completely ignored the RTS/CTS/ACK handshake for the broadcast packets. Therefore, the simple broadcast scheme provides no protection against the hidden terminals.

For reliability and protection against hidden terminals, though, some efforts have been performed to propose ways of extending the RTS/CTS handshake to a broadcast scenario [15] [17] [18] [19]. However, the methods still give rise to some side effects such as high overhead, long handshake delay, and additional out-of-band control channels.

We deduce that the use of RTS/CTS would not be enough to deal with collision cased by the Blind Broadcasting.

1.2. Proposed approaches

To solve the problems addressed in the previous sections, this thesis proposes two approaches: One is to newly design a platform to serve event-driven social networking in the infrastructure-free milieu, and the other is to create a robust and agile protocol to manage the quite tricky broadcast operations in the wireless/mobile ad-hoc and mesh networks. It is wrought by XCast project we have dived into.

XCast is a system for personalized and group-wise broadcasts based on the technologies of wireless/mobile ad-hoc and mesh networking. Its goal is to serve a platform, dubbed *"Local Life Interface"* which predicts, creates, and even copies with local events, timely, in context of the communications and networking system. Ultimately, the project pursues to build a platform for *social event networking* that reinvigorates our local lives.

The rationale behind XCast system architecture incarnates two contrary notions, 1) "personal" yet 2) "group-wise," in a broadcast system. First, the "personal" means that users can possess their own communication channels. Namely, XCast system enables the users to address whatever they want at any time without a glitch. The "group-wise" is naturally one of traits of broadcasts. It means that the system provides a collaborative methodology for systematical group communications having people socialized via broadcasting events. With the underlying principle, XCast has two focuses to resolve

both issues on the inexistence of local communicators and the awareness of local events. The first focus is on creating an instant multimedia broadcast "station", based on the personal and group-wise broadcast system, for individually motivated presentations or extemporaneous events rising in local areas. The second one is to develop the station further into a platform for social event awareness to be able to coordinate the various local events. Over the platform, we could have a good taste (or crystal-gazing) for what is practically happening around us in real-time manners. Note that in the thesis, the approaches that we have to tackle the problems above are toward system architectural and network topological facets, in the base of technologies of the ad-hoc/mesh networking and of the location data mining.

With respect to the broadcasting jumble, we propose a distributed protocol to cohesively manage broadcast resources, called "Broadcast Resource Schedule Protocol (BRSP)." It is fundamentally wrought by synchronizing broadcast operations of all nodes within an ad-hoc/mesh network. In other words, BRSP encourages all nodes within a space where they are directly reachable each other to previously share the status of resources scheduled for broadcasts before that transmissions. The synchronization is performed by a distributed queuing mechanism; afterward, they can be aware organically of what is going on all the broadcast transmission; this helps prevent nodes from blindly broadcasting in the ad-hoc/mesh networks. The BRSP results in resolving the broadcasting jumble caused by the blind broadcast.

In the BRSP protocol, the distributed queuing mechanism is the most crucial core component to make the protocol practically work. It starts under an assumption that each node has its own broadcast queue, which is used to queue data supposed for broadcast, if any. The process for the synchronization has four steps: sensing, grouping, sharing, and updating. First, each node senses which neighbor nodes stay in its own radio coverage. Next, it tries to build a broadcasting group with the nodes found. In the case, the group corresponds to *a logical or virtual cell* in an ad-hoc/mesh network. After the grouping, all members of the broadcast group start to share their own queue information with each other. If the initial update for the broadcast queues is completed, a *virtual broadcast*

queue is created. In other words, the virtual broadcast queue is a product by the synchronization among individual queues, and then the each queue plays a role of a queue slot for the virtual queue. In the end, the virtual queue schedules the transmission of broadcast in orderly and cooperative ways. Later, all nodes update their queues whenever there is any update information on a broadcast queue.

Actually, BRSP is a protocol for building an ad-hoc/mesh network into *"a cognitive network"* where all nodes organically recognize each other and are aware of what is going on in other nodes. The cognitive network consists of more than one broadcasting group dubbed a *"virtual ad-hoc cell."* It allows the wireless network to have global topology information and results in alleviating "spontaneous" traits of broadcasts. There exists a broadcast arbiter within a group for inter-group interactions if any. The arbiter is a node which is simultaneously pertain to more than one group, and transforms broadcasts among groups into unicast transmission. Such the reduction of data streams completely removes the redundancy caused by unnecessary re-broadcasting in the multi-hop routing.

The broadcast operations are performed by the unit of a group. Strictly speaking, the operations are too much closed to multicasting. With the group-based broadcasting, we can consider adaptation of a reliable transmission protocol to infallibly distribute a broadcast message to as many hosts as possible, just with less effort. For the reason, the group-based broadcasting decreases the number of receivers responding with the acknowledgements provided that the transport protocol makes intelligently use of the acknowledgement method for the reliable packet transmission. This lessens the effects on the issue of overhead introduced by the use of acknowledgement in the transmission protocol for broadcasts. In the thesis, we use TCP-like multicast protocol, called Group-based Reliable Transmission Protocol (GRTP) for broadcasts.

All in all, the BRSP architecture effectively copies with most problems caused by broadcasting in the wireless/mobile ad-hoc and mesh networks, such as spontaneity in the lack of global topology information, unreliability of the transport protocol, redundancy

via re-broadcasts, and contention and collisions by blindly flooding. No broadcasting is prohibitive any more in the wireless ad-hoc and mesh networks.

1.3 System description

What is needed to achieve the XCast system is a mobile platform equipped with 802.11 ad-hoc/mesh networking. We use OLPC (One Laptop Per Child) XO machines [20] and experimental prototypes we made using VIA mini-ITX boards [22] and Netgear *madwifi* cards with Atheros chip set [23] for that platform.

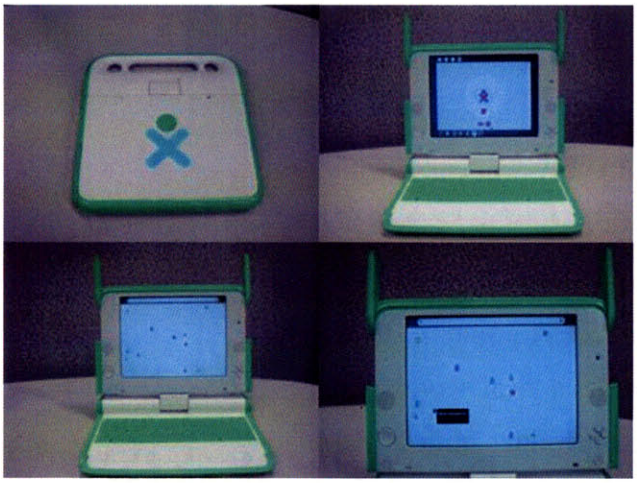


Figure 4. The features of OLPC XO

One Laptop per Child (OLPC), originally a new research initiative that the MIT Media Lab launched to develop a $100 laptop in January 2005, is a new, non-profit association which is independent of MIT. With a motto, *"a technology could revolutionize how we educate the world's children,"* it is developing a novel laptop, XO. The XO is a potent learning tool created expressly for children in developing countries, living in some of the most remote environments. In wireless networking point of view, it targets at that children in the most remote regions of the globe—as well as their teachers and families—will be connected both to one another and to the Internet via XOs [21]. Figure 4 shows features of the OLPC XO laptop.

The laptop comes with a built-in wireless card compatible with 802.11b/g standards. A unique ability it has is that the wireless chip has very low-level mesh routing capabilities built-in. To achieve the design goal, the self organizing multi-hop networking features are added to the laptop's network adapter. The constraints imposed by its Mesh Network Details mandate the use of System on Chip (SoC) Wireless Adapter, with the mesh networking protocol running directly on the adapter's CPU. Thanks to this, the laptop is able to act as a wireless router while the main processor is idle.

In the XCast system, the topology is formed by the group-based methodology for broadcasts. Each group constitutes of several, individual broadcast nodes. A wireless ad-hoc/mesh network can be embodied by more than a group. There also exists a local server to run the event navigator for the social event awareness within an ad-hoc/mesh network. In this case, XOs become mobile gears to deploy the broadcast stations or localized servers. A group of stations are wirelessly connected to a local server by the multi-hop networking capabilities built in XOs. In the literature, we use six XO laptops for the performance tests of the BRSP protocol.

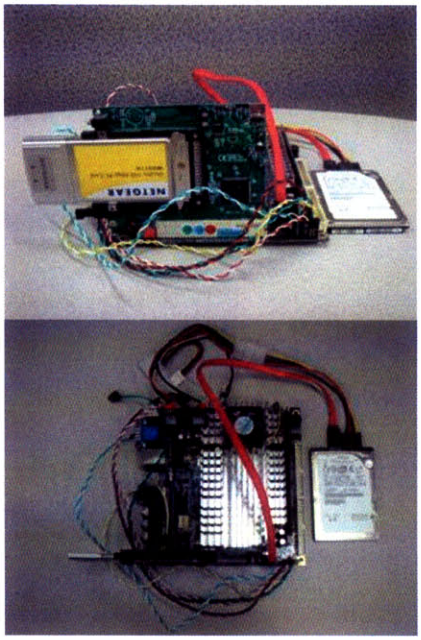


Figure 5. The Pilot Platforms for the wireless ad-hoc nodes

We have created experimental prototypes for wireless ad-hoc nodes. They were built by using VIA mini-ITX boards and Netgear *madwifi* (Atheros chipset) cards. Actually, the prototypes were used for a primary testbed to experiment the XCast system in a simplified ad-hoc network. Figure 5 shows the prototype we built.

1.4 Thesis outline

Chapter 2 collects some of the work relevant to the thesis. The chapter is divided into two main subsections in platform and protocol contexts. In other words, the former scrutinizes the existing platforms of personal broadcasts, event notification, and event organization. The latter describes the contemporary protocols developed for reliable broadcast transmission in the wireless ad-hoc/mesh networks. Both subsections give a brief overview of the state-of-art in, review some ideas used for, and see the main contributions of those platforms and protocols, respectively.

Chapter 3 furnishes the more details of the XCast system to see its forest. At first, the chapter explores all the possible scenarios that it could face. Second, it describes the reasoning behind of the system design as well as the architecture design of the system. Next, we take a look at the user interfaces of the system. Finally, we introduce algorithms created to solve defined problems.

In Chapter 4, we dive deeply into the broadcast resource schedule protocol (BRSP). The chapter describes the reasoning behind of the algorithm, and shows that how it works to successfully deal with the problems caused by the broadcast operations in the wireless ad-hoc/mesh networks.

Chapter 5 shows describes the implementation details of the system to build a working system and introduces the experiments on the BRSP protocol we have conducted. Based on simulation and experimental analysis, we assess the system built under the design principles described in the thesis, and also gather the lessons learned from designing and building the system.

Finally, in Chapter 6, this thesis concludes by summing up the contributions this system made, as well as the outstanding or future problems that could come about.

CHAPTER 2. RELATED WORK

This chapter reviews the existing work to have tackled the issues XCast system introduced in the previous chapter. The chapter is divided into two principal sections, "platforms" and "broadcast protocols": The first section gives descriptions both on the traditional and the-state-of-art platforms to serve as personalized broadcasting and event-caring systems, respectively. The second one compares the contemporary protocols to make broadcast transmissions efficiently work in wireless/mobile ad-hoc network.

2.1 Platforms

2.1.1 Personalized broadcasters

2.1.1.1 Amateur Radio

Amateur radio, often called ham radio, since its advent in 1920 has been both a hobby and a service that uses various types of radio communications equipment to communicate with other radio amateurs for public service, recreation and self-training [24]. This was a first trial which made broadcast media personalized as shown in Figure 6. Throughout its history, amateur radio enthusiasts have made significant contributions to science, engineering, industry, and social services. Research by amateur radio operators has founded new industries, built economies, empowered nations, and saved lives in times of emergency. It, though, has not been an easy-to-use medium we can readily carry and rather asked us to be trained for use of it.

Figure 6. Amateur Radio

2.1.2.2 YouTube

'YouTube' [25] is a popular, web-based content provider which provides users with a video sharing space where users can upload, view and share video clips. It allows users to indirectly build personalized broadcasts through Internet, but it is practically closer to a unicast way than a broadcast one, and does not also provide any method to immediately address temporary events like emergency cases.

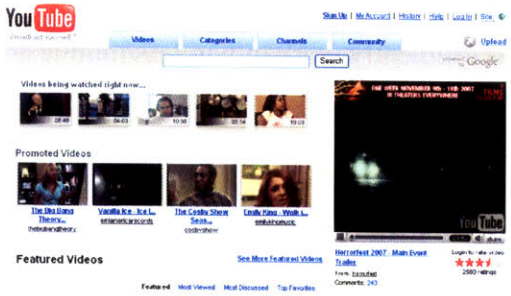


Figure 7. YouTube: Web-based Personalized Broadcasts

2.1.2.3 Podcast

Podcast made by Apple [26] is a feed using XML files to download digital media files, which is distributed over the Internet, into portable media players and personal computers. This is a kind of RSS dealing with multimedia files. Although enabling users to listen to the latest audio files downloaded by the automatic update mechanism, it just feeds recorded audio files to users, not streams ones in real-time. Also, it is web-based broadcast system.

Figure 8. Apple Podcast

2.1.2.4 FluidVoice

FluidVoice is a broadcast system yet rather close to a conferencing system, deployed in wireless/mobile ad-hoc networks, which enables people to listen simultaneously to all in-coming broadcast sounds [27]. Also, it has been developed to serve a group communication tool to share audio files over Wi-Fi-enabled mobile devices. Currently the FluidVoice system, however, is not only inherently exposed to the *blind broadcasting issue* discussed in the previous chapter but also not movable due to limits of 802.11 a/b/g protocols. That is, it focuses on not providing any methods to resolve packet collisions or network system failures caused by broadcasts but developing robust schemes to at a time, capture all audio data being broadcast in 802.11 series system (using CDMA/CA) environments.

2.1.2 Event-caring systems

2.1.2.1 Web-based event notification: WikiCity, Rome

Real Time WikiCity Rome is the MIT SENSEable City Lab's contribution to the 2006 Venice Biennale, directed by Professor Richard Burdett [28]. The project is designed to aggregate data from cell phones (obtained using Telecom Italia's Lochness platform), buses and taxis in Rome to better understand urban dynamics in real time. By revealing the pulse of the city, as shown in Figure 9, the project aims to show how technology can help individuals make more informed decisions about their environment. In the long run, it would be possible to reduce the inefficiencies of present day urban systems and open the way to a more sustainable urban future.

With the efficiency of real-time control systems, such as energy savings, regulation of the dynamics, increased robustness and disturbance tolerance, The WikiCity project targets at how a city performs as a real time control system. It has four key components:

- Entities to be controlled in an environment characterized by uncertainty.

- Sensors able to acquire information about the entity's state in real-time.

- Intelligence capable of evaluating system performance against desired outcomes.

- Physical actuators able to act upon the system to realize the control strategy.

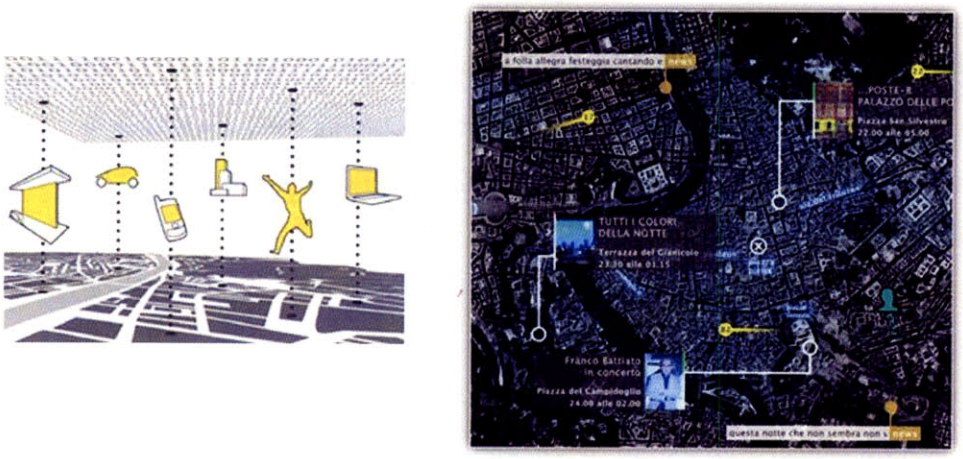


Figure 9. The concept and interface of WikiCity, Rome

As an example of the components, the Real Time Rome project used cellular phones and GPS devices to collect the movement patterns of people and transportation systems, and their spatial and social usage of streets and neighborhoods. Although the city already contains several classes of actuators such as traffic lights and remotely updated street signage, a much more flexible actuator would be the city's own inhabitants.

Consequently, the goal of WikiCity is to create a new platform for storing and exchanging data which are location and time-sensitive, making them accessible to users through mobile devices, web interfaces (as shown in Figure 9) and physical interface objects. This platform enables people to become distributed intelligent actuators, which pursue their individual interests in cooperation and competition with others, and thus become prime actors themselves in improving the efficiency of urban systems.

The basic idea of the WikiCity topology, which is illustrated in Figure 10, is to create a digital effigy of a real-world city. Thus, the system allows users, i.e. the inhabitants of a city, to integrate information themselves and update their own digital context. Furthermore, external provider services can add data to the system in real-time by automated web services. Generally, all data in the system are related to one or more categories in the ontology, which is the basis for a structured query system and well-defined data maintenance. User services act as interfaces between the end user and the WikiCity system by providing particular capabilities and operations, which are performed on a certain kind of data.

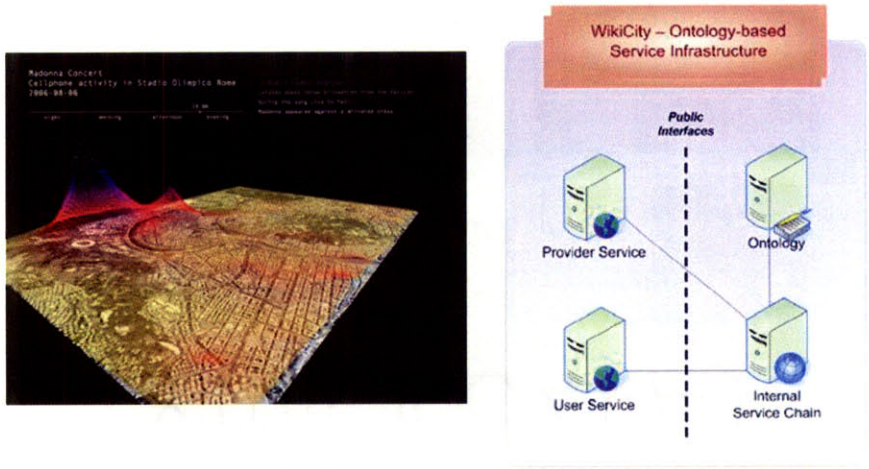


Figure 10. WikiCity basic topology

This project plays a role of a steppingstone to a real-time event notification system, It, however, is just developed as a concept project and remains still on-going to the development; the detailed functionalities of the system are not yet developed or deployed in a real city.

2.1.2.2 Hybrid event organizers: Nokia, Lifeblog

Nokia Lifeblog is a Multimedia diary and website administration tool that automatically collects all the photos, videos, and sound clips that the users create on their own mobile (cellular) phones, including text messages and MMS messages that were sent and

received [29]. It also allows the user to create text and audio notes. It organizes all the contents in a Timeline and renders the diary searchable via its contents and via automatically and manually created metadata, including time, location, tags, descriptions, filenames, sender and recipient information.

Lifeblog comprises of two applications, Photos for Microsoft Windows PCs and Share Online for Symbian O/S-based mobile phones which synchronize with each other. The PC application also allows the import of photos and other compatible items from devices other than mobile phones.

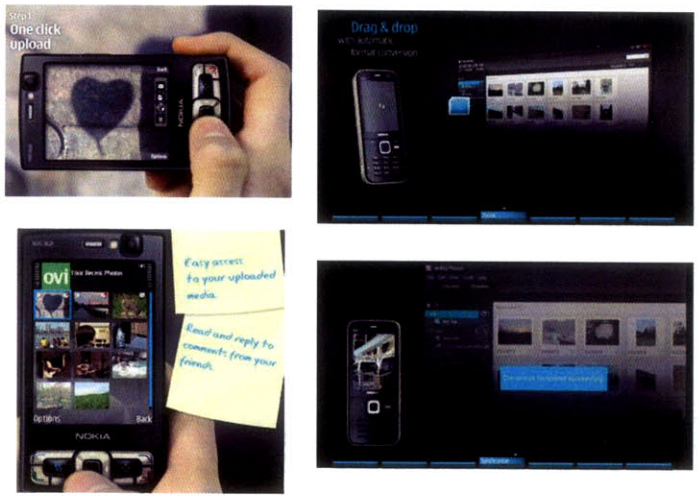


Figure 11. Nokia Lifeblog: Photos and Share Online

To post items to a blog from Lifeblog, you can use Atom-enabled blogs such as LifeLogger, TypePad, LiveJournal, Jutut, LifeType, MediaBlog, and SaunaBlog, be sent to a Flickr account. These can also be used in the two applications, Nokia Photos and Share Online.

The Lifeblog is a platform to directly create personalized events on the mobile phones, organize those events on the PCs, and share them with other users through online social networking sites such as Flickr, Ovi, and so on.

2.2 Protocols for reliable broadcasts

In this section, we describe the existing broadcast protocols to cope with the inherent issues caused by broadcast operations in the wireless/mobile ad-hoc networks.

2.2.1 Categorization of Protocols

We describe a comprehensive comparison of the reliable broadcast protocols in the section. This significant work helps glean an overall understanding of strengths and weaknesses of the protocols. Our approach for the excise is to categorize the protocols into four families according to their main contributions: Simple Flooding, Probability-based methods, Area-based methods, and Neighbor-Knowledge methods. In described in Section 5.3.1, we select one or two representative protocols from each category except for the Area-based methods, to compare the protocols with our proposed method, BRSP. This allows us to make conclusions regarding applicability of the BRSP protocol based on the traits of the selected protocols. We give a capsule overview of each category in the following paragraph, and then provide the more details in the succeeding subsections.

Simple Flooding requires each node to re-transmit all received broadcast packets one time. Probability-based methods use some basic understanding of the network topology to assign a probability to a node to re-broadcast. Area-based methods assume that nodes have common transmission distances; a node will re-broadcast only if the retransmission reaches sufficiently the additionally requested coverage area. Neighbor-Knowledge methods maintain states on their neighborhood, via beacon signals (e.g., "Hello" packets), which are used in the decision to re-transmit broadcast packets. Note that in the succeeding section, we explore the categories in order of the increase of intelligence, algorithm complexity and of state requirement per node for example. The goal of the added cost is to reduce the baleful effects caused by the re-transmissions of broadcast packets, such as the overhead and the packet collisions.

2.2.2 Simple Flooding

Assuming that there exists a source node broadcasting a packet to all neighbors, authors in [30] [31] propose the algorithm of Simple Flooding. That is, each of the neighbors in turn re-broadcasts the packet exactly one time and this continues until all reachable network nodes have the packet. In [30], Ho et al. illustrate a "Flooding" scheme to achieve reliable broadcast/multicast in highly dynamic networks. Also, Jetcheva et al. in [31] make use of the Simple Flooding for broadcast/multicast transmission in ad hoc networks which are characterized by low node densities and high mobility.

2.2.3 Probability-based methods

2.2.3.1 Probabilistic scheme

Ni et al in [4] propose the Probabilistic scheme similar to Flooding, except that nodes only re-broadcast with a predetermined probability. The probability is determined dependent on the node density in a wireless ad-hoc network. In dense networks where multiple nodes share the common transmission coverage, for example, the probability is chosen as "low." This prevents nodes from re-broadcasting and thus saves network resources without any baleful effect on the packet delivery. In sparse networks, there is much less shared coverage; thus, the probability is decided as "high." When the probability is 100 percent, this scheme is identical to Flooding.

2.2.3.2 Counter-based scheme

Ni et al [4] also suggest the Counter-based scheme, which decides to re-transmit broadcast packets in the basis of the number of times a packet is received at a node. The principle behind this scheme is that the expected additional coverage (EAC) a node is in charge of for the broadcast transmission would be lower as the number of times of which a packet reaches at a node increases. The algorithm of this scheme is as follows: Upon reception of a previously unseen packet, the node initiates a counter with a value of one and sets a "*channel assessment delay (CAD)*" as the time delay to acquire a clear channel where no collision happens. During the CAD, the counter is incremented by one for each

redundant packet received. If the counter is less than a threshold value when the CAD expires, the packet is re-broadcast. Otherwise, it is simply dropped. In [4], the authors show the threshold values above four make the EAC below 0.05 percent. In the results, in a dense area of the network, the scheme prevents nodes from frequently re-transmitting broadcast packets. In sparse areas of the network, however, it allows many nodes to re-broadcast.

2.2.4 Area-based methods

Area-based method enables a node to evaluate the expected additional coverage based on all received redundant transmissions. For instance, the additional area covered by the retransmission of broadcast packets is quite low, provided that a node receives a packet from a sender that is located only one meter away. On the other extreme, if a node is located at the boundary of the sender's transmission distance, then the expected additional coverage a re-transmission for broadcast reaches would increase significantly: The authors of [4] shows the probability of the expected additional coverage is 61 percent via the mathematical method. Note that the Area-based methods only focus on the coverage area of a transmission regardless of whether nodes exist within that area. We introduce two area-based schemes proposed in [4].

2.2.4.1 Distance-based scheme

The Distance-based scheme allows a node to compare the distance between itself and each neighbor node that has previously re-broadcast a given packet. Upon reception of a new packet, the scheme initiates a delay timer, CAD and stores redundant packets without any re-transmission. When the CAD expires, it investigates the locations of all source nodes to see if any node exits closer than a threshold distance value. If true, the node doesn't re-transmit for the received broadcast packets.

2.2.4.2 Location-based scheme

To decide the transmission of broadcast packets, the Location-based scheme utilizes a more precise estimation of the expected additional coverage. In this method, each node has a method to determine its own location, Global Positioning System (GPS) for example. The algorithm is as follows: Whenever a node originates or re-broadcasts a packet, it adds its own location to the header of the packet. Upon initial reception of a packet, a receiving node checks the location of the sender and calculates the additional coverage area obtainable were it to re-broadcast. If the additional coverage is less than a threshold value, the node will not re-transmit the received broadcast packets, and all future receptions of the same packet will be ignored. Otherwise, the node assigns a CAD before the re-transmission. If the node receives a redundant packet during the CAD, it re-calculates the additional coverage area and compares that value to the threshold. The coverage calculation and the threshold comparison occur with all redundant broadcasts received until the packet either reaches its scheduled sending time or is dropped.

2.2.5 Neighbor-Knowledge methods

The overriding compelling features of the previous schemes are its simplicity and its inherent adaptability to local topologies. They are less intelligence than the schemes to be described in the section, in decision of whether to re-transmit broadcast packets. The intelligence the schemes described in the section make mainly use of is "Neighbor Knowledge" methods. There are two types of the methods: One is to enable a node to use its own local topology knowledge for decision of re-broadcasting, and the other is to allow preceding nodes to decide the re-broadcast of succeeding nodes. The former is one of core methods our proposed BRSP protocol adopts. In this section, we further delve for the two categories of Neighbor-Knowledge methods for the broadcast transmission.

2.2.5.1 Flooding with Self Pruning

Lim et al. illustrate "Flooding with Self Pruning" scheme [9]. It is the simplest of the Neighbor-Knowledge methods. This protocol requires that each node has the knowledge of its one-hop neighbors, which is obtained via periodic beacon signals (e.g., "Hello" packets). A node adds its own list of the known neighbors to the header of each broadcast packet. When receiving a broadcast packet, a node compares its own neighbor list to the sender's neighbor list in the header of the received packet. If the receiving node would not reach any additional nodes, it refrains from re-broadcasting; otherwise the node re-broadcasts the packet.

2.2.5.2 Scalable Broadcast Algorithm (SBA)

Peng et al. suggest the Scalable Broadcast Algorithm (SBA) [32], which requires all nodes have knowledge of their neighbors within a two-hop radius. In the algorithm, this neighbor knowledge is coupled with the identity of the node which sends a broadcast packet. Such the neighbor knowledge allows a receiving node to determine if it would reach additional nodes by the re-transmission of broadcast packets. Two-hop neighbor knowledge is achievable via periodic beacon signals (e.g., "Hello" packets); each beaconing packet contains the node's identifier (IP address) and the list of known neighbors. After a node receives a beacon packet from all its neighbors, it has two-hop topology information centered at itself.

Suppose a node R receives a broadcast packet from a neighbor node S. Then, the node R knows all of its neighbors, common to the node S, that have also received the broadcast packets from the node S. If the node R has additional neighbors not reached by the node S's broadcast, it schedules the packet for delivery with a CAD. If the node R receives a redundant broadcast packet from another neighbor, the node R again determines if it can reach any new nodes by re-broadcasting. This process continues until either the CAD expires and the packet is sent, or the packet is dropped.

The authors of [32] propose a method to dynamically adjust the CAD to network conditions; they weight the time delay on the basis of a node's relative neighbor degree. Specifically, each node searches its neighbor tables for the maximum neighbor degree of any neighbor node. It then calculates a CAD based on the ratio of the maximum number of neighbors and current number of neighbors. This weighting scheme is greedy; nodes with the most neighbors usually broadcast before the others.

2.2.5.3 Dominant Pruning

Lim et al. also describe the Dominant Pruning, which applies two-hop neighbor knowledge, obtained via beacon signal (e.g., "Hello") packets, in order to decide broadcast routing [33]. Unlike SBA, however, the Dominant Pruning uses a proactive approach to look for re-lay nodes for broadcast packets. That is, it enables nodes to in advance choose some or all of its one-hop neighbors to re-transmit broadcast packets. Those chosen nodes are only permitted to re-transmit broadcast packets. When a node receives a broadcast packet, it first checks the header of the packet to see if its own address is part of the list. If so, it makes use of a *Greedy Set Cover algorithm*(2.1) [34] to determine which subset of neighbors should re-broadcast the packet, given knowledge of which neighbors have already been covered by the sender's broadcast. The greedy algorithm for set covering recursively chooses one-hop neighbors which cover the most two-hop neighbors and then re-calculates the cover set until all two-hop neighbors are covered.

2.2.5.4 Multipoint Relaying

Qayyum et al. suggest the Multipoint Relaying [35], which is similar to the Dominant Pruning in that the broadcast relay nodes are explicitly chosen by upstream senders. Suppose that node S is originating a broadcast packet, for example. The node S has chosen some nodes (or in certain cases all) of its one-hop neighbors to re-transmit all

(2.1) The greedy algorithm for set covering chooses sets according to one rule: at each stage, choose the set which contains the largest number of uncovered elements [44].

broadcast packets received from the sending node S. The selected nodes are dubbed *"Multipoint Relays (MPRs)"* that are the only nodes allowed to re-broadcast a packet received from the node S. Each MPR is required to select a subset of its one-hop neighbors to act as the MPRs as well.

Since a node knows the network topology within a two-hop radius, it can select one-hop neighbors as the MPRs that most efficiently reach all nodes within the two hop neighborhood. In a node, the algorithm to choose its MPRs has four steps:

- Step 1: Find all two-hop neighbors that can only be reached by a one-hop neighbor. Assign those one-hop neighbors as the MPRs.

- Step 2: Determine the resultant cover set, namely the set of two-hop neighbors that will receive the packet from the current MPR set.

- Step 3: Find one-hop neighbors that would cover the most two-hop neighbors not in the cover set, from the remaining one-hop neighbors not yet in the MPR set.

- Step 4: Repeat from step 2 until all two-hop neighbors are covered.

Optimized Link State Routing (OLSR) protocol adopts the Multipoint Relay as part of its protocol [36]. In this protocol, beaconing (e.g., "Hello") Packets include fields for a node to list the MPRs it has chosen. Whenever a node receives a beacon packet, it checks if it is an MPR for the source of the packet. It should relay all the broadcast packets received from the source if it is an MPR. Unambiguously, the update interval for the beacon packets must be carefully chosen and, if possible, optimized for network conditions.

2.2.5.5 Ad Hoc Broadcast Protocol

Peng et al. introduce the Ad-Hoc Broadcast Protocol (AHBP) [37], which utilizes an approach similar to the Multipoint Relaying. In the AHBP, a node that is designated as a

Broadcast Relay Gateway (BRG) within a broadcast packet header is allowed to re-lay broadcast packets like the MPR. The BRG is also proactively chosen from each upstream sender which is a BRG itself. The algorithm to choose its BRG set is identical to that used in Multipoint Relaying. However, the AHBP differs from the Multipoint Relaying in three ways:

- Piggy-backing: In the AHBP, a node informs one-hop neighbors of the BRG list via the BRG designation within the header of each broadcast packet. This permits a node to calculate the most effective BRG set at the time a broadcast packet is transmitted. In contrast, the Multipoint Relaying informs one-hop neighbors of the MPR designation via beacon signals, "Hello" packets.

- Source Routing: When a node receives a broadcast packet and notes it is listed as a BRG, it utilize the two-hop neighbor knowledge to determine which neighbors also received the broadcast packet in the same transmission. If these neighbors are considered already "covered," they are removed from the neighbor graph used to choose next hop BRGs. In contrast, the MPRs are not chosen considering the source route of the broadcast packet.

- Mobility support: The AHBP is extended to account for high mobility scenarios, called "AHBP-EX (extended AHBP)." For example, suppose that a node A receives a broadcast packet from a node B yet does not list the node B as a neighbor because they have not yet exchanged beaconing, "Hello" packets. The node A refers to BRG status and then re-broadcast the received packets. Multipoint relaying could be similarly extended.

2.2.5.6 CDS-Based Broadcast protocol

A Dominating Set of a network is a subset of all the nodes such that each node is either in the dominating set or adjacent to some node in the dominating set. A Connected Dominating Set of a network is a subset of the nodes such that it forms a dominating set

in the graph (or the network) and the subgraph induced is connected. Peng et al. in [38] propose a broadcast protocol using Connected Dominating Set (CDS) concept. That is, it algorithm uses a more intensive calculation to select BRGs. The protocol additionally utilizes *the set of BRGs with higher priority* selected by the previous sending node while adopting the source of the broadcast packet AHBP uses to determine a receiving node's initial cover set.

Suppose a node A has selected nodes B, C, and D (in this order) to be BRGs, for instance. When the node C receives a broadcast packet from the source node A, the CDS-Based Broadcast protocol requires the node C to add neighbors common to the node A into the initial cover set. In addition, it also requires that the node C adds neighbors common to node B, because the node B is a higher priority BRG. Likewise, the node D is required to consider common neighbors with the nodes A, B and C. Once the initial cover set is determined, a node then chooses which neighbors should function as BRGs. The algorithm for determining this is the same as that for the AHBP and the Multipoint Relaying, as described in the steps 1 to 4 for choosing Multipoint Relays, in Section 2.2.5.4.

2.2.5.7 Lightweight and Efficient Network-Wide Broadcast (LENWB)

Sucec et al. suggest Lightweight and Efficient Network-Wide Broadcast (LENWB) [39], which makes use of two-hop neighbor knowledge acquired from the beaconing, "Hello" packets. In lieu of a node explicitly choosing nodes to relay broadcast packets, however, the decision is made implicitly. In other words, unlike the Multipoint-Relaying, the AHBP, and the CDS-based broadcast protocol, each node decides whether to re- transmit broadcast packets on the basis of knowledge of which of its other one- and two-hop neighbors are expected to re-broadcast. The information needed for that decision is the knowledge of which neighbors have received a packet from the common source node and of which neighbors have a higher priority for the broadcast relaying. The priority is proportional to a node's number of neighbors; the higher the node's degree the higher the priority. Since a node depends on its neighbors with higher priorities to relay the received

broadcast packets, it can proactively compute whether all of its lower priority neighbors receive those re-broadcast packets.

CHAPTER 3. XCAST: A PERSONAL AND GROUP-WISE BROADCAST SYSTEM

XCast is a Personal and group-wise broadcast system to collaboratively distribute information and to coordinate locally created events in infrastructure-free milieu. The name of XCast has two meanings: Actually, "X," the first character of XCast has two meanings. First, it means all directions, "broadcast" in communication point of view. The second meaning is "any". That is, it represents, *any*one can broadcast *any*thing in *any* place, at *any* time. In the project, we re-define it "personalization" to own resources requested for the broadcasts in orderly or preferred manners and then to freely manage those. As depicted from its name, XCast is a project to create a personalized broadcast system in the basis of the natural traits of broadcast, "group-wise."

At first, this project got started from "Andy-casting," an idea for an atomic project to explore the applicability of broadcasting in wireless ad-hoc/mesh networks.

> *Andy-casting is based on the idea that we might want to deliberately push information in our machine onto the network to prompt others to access it. It is already the case that PCs have "shared folders" and ad-hoc wireless modes that allow proximate machines to browse them. This is different in that you are providing something you desire to distribute, and creating a simple program in the PCs would allow them to "listen in" or "tune in" to it. For example, a song I am listening to, or a paper I am reading and perhaps printing could be radiated or advertised in an interesting way, or perhaps information about a study group, ride-sharing or just the web pages I am browsing. - Andrew Lippman*

We have further developed the idea into an intriguing project, XCast with the following motive, "*how do we know seasonably what is happening around us and then deal with the rising event?*" In other words, the project concentrates on developing a system

supporting for event-driven social networking, based on the personal and group-wise broadcast system: It is an ultimate goal the project pursues.

With the background, XCast has two targets: One is to permit people to creating personalized communicators, "instant broadcast station" over mobile devices, for extemporaneous events or individually motivated presentations. The other is to provide people with a cognitive platform for social event awareness that informs what is going on in our surroundings and then coordinate the events in timely manners. To achieve the targets, in the thesis, we focus on newly designing architecture of the cognitive platform and then developing a robust and agile protocol which makes it possible for the platform to reliably work in wireless ad-hoc/mesh networks. In the succeeding sections, we give the more detailed descriptions.

3.1 Scenarios

There are a variety of scenarios where XCast system would be useful. We approach to categorize feasible scenarios into four groups: journalism, emergency, commercials, and entertainment. Above all, what we need to take a closer look at is a fast rising neo-fashion of journalism: Simply, "everyone becomes a reporter." This gets started from the hypothesis that all people inherently want to share what they witness with others and so are socialized naturally. One example that demonstrates it is "OhmyNews," Korean online newspaper with a motto, "every citizen is a reporter." It is the first of its kind in the world to accept, edit, and publish articles from its readers, in an open source style of news reporting. About 20 percent of the site's content is written by the 55 staff, while most of the articles are written by other freelance contributors who are mostly ordinary citizens [51].

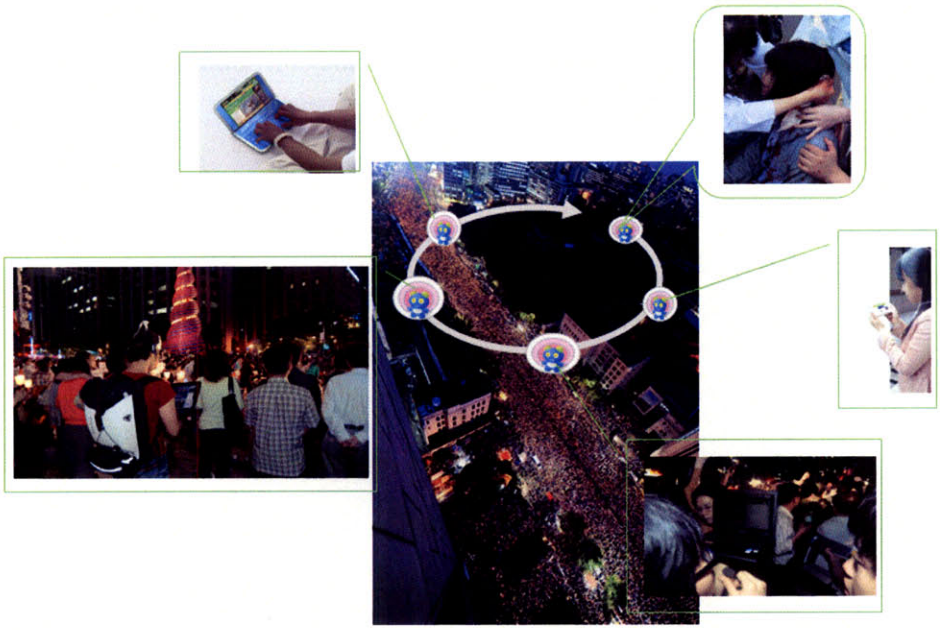


Figure 12. Street Journalist scenario: Reality Report

With the advent of the participatory online newspaper, recently, a marveling event happened in Seoul, Korea. That was the appearance of "Reality Reporters" XCast system could be applied to. For the last three months (e.g., May 2 to August 1, 2008), a lot of Korean citizens, nearly up to 700,000 people at a time, holding a candlelight in their hands, have demonstrated in peaceful manners against some policies on the wrong track or inaptitude of the trade deal between US and Korea, conducted by their government. Korean major conservative newspapers violently criticized the candlelight demonstrations and insisted there were masterminds behind the demonstrations, even if they were held by voluntarily participation of citizens. A lot of citizens vexed about the biased, distorted articles of the newspapers, and then some students and citizens equipped with just their Wi-Fi or WIMAX enabled laptops and webcams, started to broadcast the candlelight demonstrations in real-time, via Internet, to lively broadcast what is really happening in the demonstration places, as shown in Figure 12. It helped transfer the truth of candle demonstration to other citizens who have never attended at that demonstration. They really played a pivotal role of reality reporters, called "street journalists." From this example, we may imagine the creation of local reality reporters who inform local news or events via the personal and group-wise broadcast system of XCast in wireless P2P

networks. XCast system would help the local reality reporters directly interact with their locals and seasonably let them know what is going on in the surroundings. Therefore, this results in new "*civic media*[3.1]" created by grassroots.

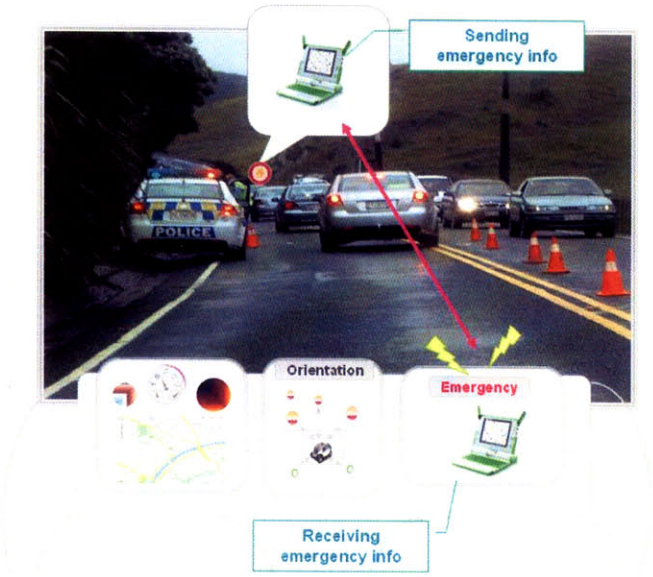


Figure 13. The scenario of emergency case: car accident notification

In general, it has been perceived that the infrastructure-free system would be useful in case of breakdown of infrastructure system. There exist, however, some situations where people can not benefit from the infrastructure system even if it is alive. For example, XCast system could help resolve situations that make people be really up the creep, caused by inexistence of wanted local helpers or by failure of communications with those helpers. In other words, the system enables its users to quickly ferret out local helpers around them and then fittingly interact with the helpers in the pinch. Also, the users are allowed to immediately post their urgency situations on Event Navigator of Event Collection Sever via the broadcast station. The possible scenarios would be various from car accidents to natural disasters. Figure 13 shows a scenario to notify other cars of the car accident to prevent secondary crashes or to ask for some helps to the surrounding

(3.1) The Media Lab initiated *"Center for Future Civic Media."* It targets at creating novel civic media leading to neo-journalism in civic boundaries and developing new technologies that support and foster civic media and political action [48].

drivers.

XCast system provides private broadcast spaces utilized in local areas. The private broadcasting spaces could be valuable and further proliferate for commercials. One possible scenario for that purpose would be "personal kiosk." That is, people can instantaneously broadcast what they would like to sell or address in local areas via their own broadcast station. In this case, receivers can tune one channel providing information flavored to themselves. When driving in downtown of cities, for instance, drivers may time-to-time want information such locations, landmarks, restaurants, and so on. Broadcasters at building or on the street locally distribute drivers what they want to share and then the receivers frequently adjust some channels to seek for the information useful to the driving as shown in Figure 14. In this case, the commercial advertisements could be Spam broadcasts. That situation should be avoided. To resolve the Spam issue, XCast system controls channel resources used for broadcasts through a decentralized broadcast queuing method and also schedules data queued on the broadcast queue in orderly or preferred ways. For the management of Spam, we give the more details in Chapter 4.

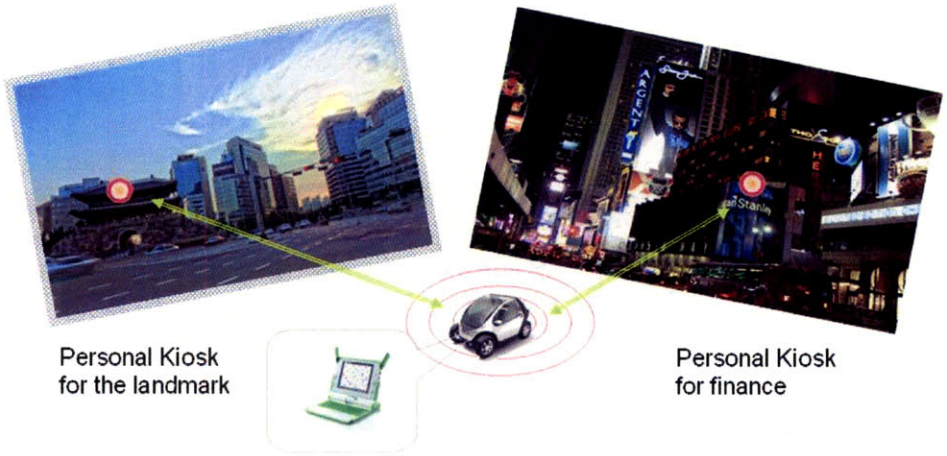


Figure 14. Personal Kiosk scenario

The scenario where XCast system would rev up is about socialization with local people for the entertainment purpose. The reason is that the system serves a platform where the users are socialized with other local users based on their preferences, and also that

broadcast station of the system the users instantly create permits them to share the multimedia streaming files such audio and video files with their locals. In this case, one simple example would be "song sharing" in the basis of the Andy-cast idea. It is similar to a jukebox which plays out various songs in order, selected from a music queue wrought by multiple people's contributions, at limited public spaces. In the XCast system, however, the song sharing is different in that a platform for the jukebox is virtually formed by collaboration among several stations, in that the sharing is performed in a temporary space the users stay, via wireless P2P networking, and in that the music queue is also created virtually by cooperation of the individual queues distributed on each station. We call it a *"virtual jukebox*[(3.2)]*."*

Figure 15. The scenario of Virtual Jukebox

Here is a specific scenario on the virtual jukebox. Jane is listening to her favorite music on a bench of campus. While listening to music, she would like to share her favorite songs with other students around her. She looks for other XCast stations that would be interested in the song sharing. After finding out some stations around her, she creates a virtual jukebox group through collaboration with others and then the virtual jukebox is established. Now, she broadcasts her songs to the members within the group over the virtual jukebox platform: 'JaneCast' is created! In this case, the broadcast songs are not copied; they remain on each station. If the other students that received 'JaneCast' also

(3.2) The idea of *Virtual Jukebox* is initiated by *Prof. David Reed* who is one of my advisors. He suggested a seed idea on the virtual jukebox and also the refined scenario written in the thesis.

want to share their songs with Jane, they can stack the songs on the queue of the virtual jukebox. Only one song would play at a time from the virtual jukebox, which is a movable audio source that appears on every user's interface. The songs shared can be mixed in among the talks on each user's mobile devices.

Anyone within the community can select and queue a song from the virtual jukebox collection - either one of their own or one contributed by another user. The collection is a file sharing system which can be created when a personal and group-wise broadcast community is built. It allows stations to freely access all files shared in the community. When selected, the song is added to the queue, at the end. The song is played out and broadcast from its home (exactly from the 'virtual broadcast' queue of XCast system, as described in Section 3.4), and when done, the next song is then played out from its home station.

All in all, XCast system helps create unprecedented aspects in cultural and application contexts. Namely, traits of the scenarios and applications the system supports are mostly impromptu but seasonable. Also, the personalized broadcasts people create over the system are built by immediate needs or as long-lasting interests within a local bound. In the long run, the system serves the scenarios over a platform with intelligence of local event awareness.

3.2 System design rationale

This section describes the underlying principles behind of the system design. The philosophy XCast system pursues is to support the primary motive, "how do we know timely what is happening around us?" That is, the whole system is designed to recognize what is going on in the surroundings and then to treat that rising issues in timely and collaborative manners, from low end to high end-channel collisions in the wireless networks to event coordination in the application layer, for example. In this case, the system is an embodiment formed by that individual nodes are cognitively networked with

each other. We call it a "cognitive platform." Particularly, the components in the cognitive platform are oriented toward awareness of sociably crated local events based on group communications, broadcasts. The succeeding subsections give more details of the essential elements needed to actualize the philosophy.

3.2.1 Personal vs. Group-wise

For the "personal" concept, in the system, it is much closer to the meaning of "personalize." To further understand "personalize" concept of the system, we need to look at its meaning at the dictionary(3.3): It is defined as "to make personal, as by applying a general statement to oneself." In the system, the personalization is transformed into "possession of a communication gear:" The system enables people to have their own broadcast stations while they apply wireless common channels to themselves for the private communication purposes.

In the *group-wise* concept, we consider two angles: The first frame of reference is for group or community communications. Using natural traits of broadcasting, the cognitive platform of the system is designed to support group communications. The other view point is to *cohesively* share limited channel resources of the wireless networks without any collision in a group, for the broadcast purposes. That is, the whole system is designed to build a group-based broadcasting transmission system.

The XCast system is architected to support for both characteristics of personal and group-wise concepts, and the two concepts are systematically integrated into the system. In other words, offering a cohesive channel allocation method, it allows the users to possess their own channels from the limited wireless resources in a broadcast group. Afterward, the broadcast-based group communications are achieved by applying orderly or prioritized scheduling approaches to the system itself.

(3.3) The definition is sourced from *Dictionary.com (http://dictionary.reference.com)*, based on the *Random House Unabridged Dictionary*, © Random House, Inc. 2006.

3.2.2 Local Life Interfaces

In the system, *Local Life Interfaces* is the denomination of the cognitive platform to timely deal with locally created events in the context of communications and networking system. It allows the users to predict, create, and even treat local events, fittingly. The cognitive platform plays principal roles of a caregiver for local life, of a gear for group communications, of a crystal-gazer for feasible events in the geographically isolated areas, and ultimately, of a collaborative platform for regional event-driven social networking.

To achieve the roles of Local Life Interfaces, the architecture of the system constitutes of components and protocols to be networked organically with surrounding mobile nodes, to cooperatively gather the events rising in that location, and finally to make it possible for users to promptly navigate what they want of those events, on the underlying structure of group-based transmission, "broadcasting." The components of the system such as Broadcast Resource Scheduler (BRS) of each mobile node and Event Collector of local servers, for example, are designed to run Broadcast Resource Schedule Protocol (BRSP) and Event Collection/Location Decision Protocol (ECLDP), respectively. In other words, first, BRSP enables each node to make cognitive of other nodes within a broadcast coverage and to build a broadcast group while cohesively interconnecting with other node. On the other hand, ECLDP practically gathers events from all mobile nodes within a wireless ad-hoc/mesh network and decides the locations the events occurred in order to provide the users with ability of the event navigation. For the more details of the components and their protocols are given in Section 3.3.2.

3.2.3 Cooperative media

One of the philosophies XCast system pursues is to create "cooperative media." It means that any operation, from the transmission layer to the application layer, a node intends in the system is achieved by collaboration among nodes in the wireless P2P network. As you could guess, in general, broadcasting is naturally spontaneous and unseeing. In the transmission of the system, however, the broadcast operation is wrought by that all nodes

within a broadcast group or community together create a distributed scheduler through the collaboration with other nodes, and then the scheduler allocates a wireless channel to a node wishing broadcasting in orderly or preferred queuing schemes. In the thesis, we call the cohesive, distributed scheduling method "Broadcast Resource Schedule Protocol (BRSP)," and provide the more details in Chapter 4.

The cognitive platform XCast system provides plays a role of a community sharing tool for multimedia data. One example that clearly shows it is the scenario, "virtual jukebox" as introduced in Section 3.1. In the scenario, the system serves a social platform enabling the users to ferret out their neighbors in the available coverage, then to collaboratively build a jukebox group with the neighbor nodes, next to form a virtual jukebox by using a distributed queuing method, and finally to share songs with the locals, via the jukebox, in orderly or preferential ways. That scenario benefits from collaborations with adjacent nodes to make maximum utilize of the limited resources. All in all, XCast system offers a collaborative platform supporting broadcast transmission and multimedia sharing.

3.2.4 Event-driven Social Networking

What the system ultimately quests after is in achieving a cognitive platform for event-driven social networking. In real-time manners, to be aware of what is happening in our surroundings means that we could have opportunities to timely deal with the events or issues locally created. Specifically, we could be socially networked while coping with the locally rising events or issues. It means that we are in a process that a social matrix grows and disappears, while connecting our own interaction points into the social networks where the events occur and afterward unreservedly leaving from the spaces, if the events arise in interactions with a group of people.

In the networking facet, the social event networking is a process that nodes are directly inter-connected around an event they are interested in or indirectly connected via event relay nodes to create a social event network. Figure 16 shows how the social event network is architected. Nodes together huddle around an event; It forms an event group. Each event group consists of nodes, relay nodes, an event initiator, and an event (circle).

In this case, an event naturally occurs or is created artificially by an initiator. As shown in the figure, the event circle is located more close to its initiator or the tightly related nodes. Note that the event happens naturally if it is located in the center of the event group without any bias. Also, each event has an importance factor according to urgency. In the figure, the size of event circle represents the degree of importance factor dependent on the urgency. The bigger the size of even circle is, the more urgent the event is. The relay node helps share the events with other nodes in other groups. The node which pertains to one more group becomes inherently a relay node for the groups it is.

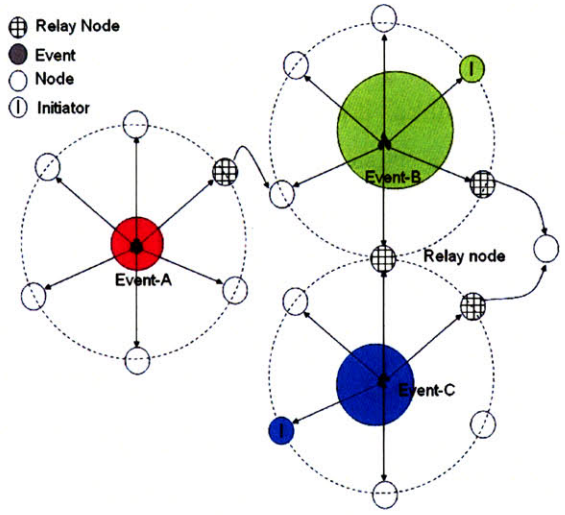


Figure 16. The conceptual topology of Social Event Network

The XCast whole system is designed to ultimately actualize the social event network to sense the occurrences regionally created or rising and to treat seasonably them. In the following sections, we describe how the social event network is formed in more detail.

3.3 System architecture

This section describes a cognitive framework of XCast system and components the framework constitutes of. Also, core algorithms letting the components practically work are given as a descriptive sketch.

3.3.1 Cognitive framework of Social Event Network

We describe how the framework of XCast system consists of to form the social event network, from the macro level to the micro level in the section. The framework, in the high level, is designed toward a cognitive networking system to coordinate social events regionally created or occurring in seasonable manners. For that purpose, the framework has the traits of neighbor finding, group building, group-based information sharing, and event notification/collection. These traits are embodied by the components the framework fundamentally constitutes of, such as Personal Broadcaster, Broadcast Group, Broadcast Resource Scheduler, Event Collection Server, and Social Event Navigator: Each personal broadcaster senses others and builds a broadcast group to share information they want. The broadcast resource scheduler makes it possible for the broadcaster to efficiently share data without any undesirable effect. These components are required for event creation. The event collection server plays a role of gathering the events created by the interaction of the components above and of running the social event navigator. Finally, the social event navigator provides the users with functions to gaze or search events. Figure 17 shows the topology of the framework.

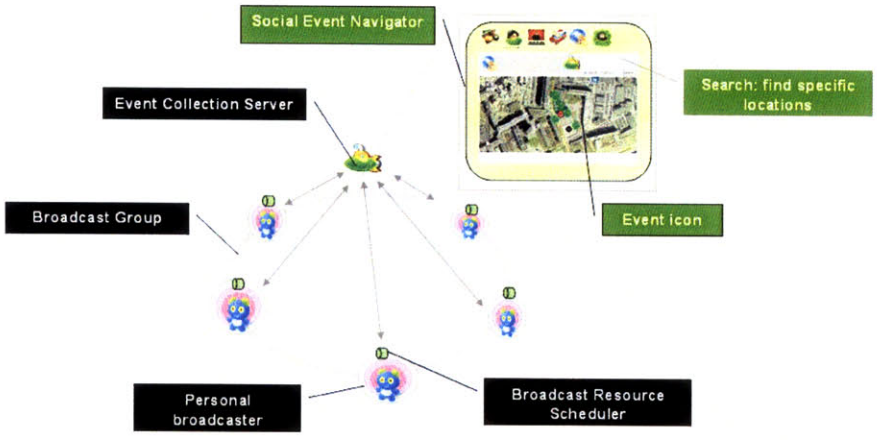


Figure 17. The topology of framework of XCast system

In the micro level, the framework is architected to transform an ad-hoc/mesh network into "*a cognitive network*" for the broadcast transmission: The cognitive network consists of a collection of *virtual ad-hoc cells*; a virtual ad-hoc cell is wrought by the creation of a

broadcast group. That is, all nodes within the coverage of the radio signal of the wireless system organically recognize each other and then form a broadcast group in the cohesive way. In results, the broadcast group is a collection of nodes that are directly reachable to each other via one-hop routing in the wireless ad-hoc/mesh network. It is similar to the concept of a cell in the wireless infrastructure network. This is why we call it a "virtual ad-hoc cell." The broadcast group enables all broadcast operations to be synchronized through a distributed resource schedule protocol. The synchronization scheme makes all nodes in the group can be fully cognizant of what is going on the broadcast transmission over other nodes. This provide a ground to make ad-hoc network more cognitive.

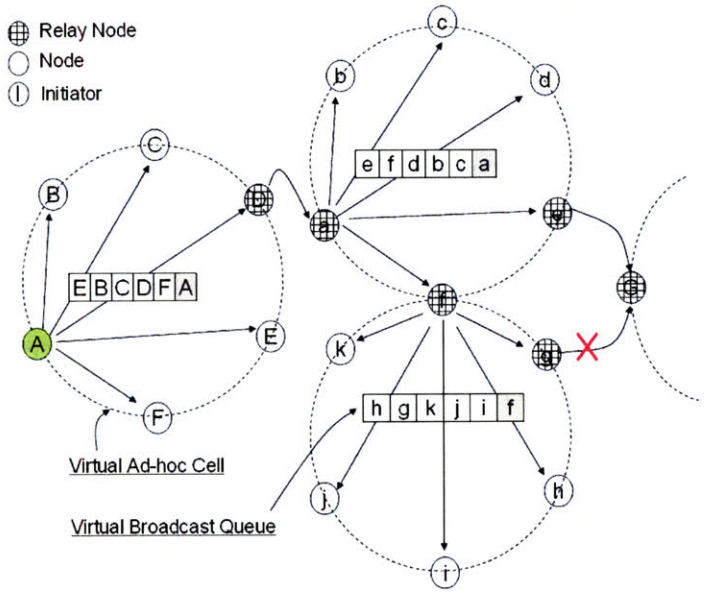


Figure 18. The topology of a cognitive network in the low level

To perform the broadcast operation among other virtual ad-hoc cells, the system makes use of multi-hop routing based on the unit of a group. The multi-hop routing is performed by unicast transmission amongst the broadcast relay nodes. In other words, there exits at least a broadcast relay node in a group that re-transmits or relays the broadcast packets to other broadcast nodes or relay nodes in other groups; a node which is belong to more than one group or directly reachable to other relay nodes in other groups could be the broadcast relay node. The multi-hop transmission via the broadcast relay nodes makes a group aware of what is going on the broadcast operation of other groups. In the end, the

multi-hop routing for the broadcast in the system is transformed into group-to-group unicast transmission. Additionally, it helps reduce the redundancy caused by re-broadcasting in the multi-hop routing. Figure 18 shows the topology of a cognitive network and also how it works.

3.3.2 System Components

This section goes through the proposed system components both for fulfilling the cognitive networking system and for supporting the social event network.

3.3.2.1 Personal broadcaster

We explore the role of the personal broadcaster as the most basic unit of the system, in the section. The roles the personal broadcaster plays are of finding neighbor nodes, building a broadcast group with the found neighbor nodes, and running the Broadcast Resource Scheduler (BRS) to share the limited radio resources with other nodes in the same broadcast group. After a broadcast group is established, the broadcast nodes are divided into two kinds of nodes: One is a master node to manage the broadcast group, and the other is several satellite nodes to support the master node for the broadcast operations. In the micro level, each node runs its own broadcast queuing function of the BRS and builds a distributed resource schedule method, *Virtual Broadcast Queue* via collaboration with other nodes. The master node leads the operation of the *virtual broadcast queue* to efficiently allocate the wireless channel to each node in orderly or preferred ways. For example, it collects the updated status of broadcast operations from each the satellite node and distributes the updated information to other satellite nodes. We give the more details on the virtual broadcast queue in Chapter 4. In the macro level, on the other hand, the master node represents its own broadcast group and notifies the event collection server of the information on the broadcast events it created with satellite nodes, whenever it is required.

3.3.2.2 Broadcast group

The broadcast group, named *"virtual ad-hoc cell (VAC)"* is the most fundamental unit of broadcast transmission: XCast system, as discussed in previous sections, utilizes a group-based broadcast-exactly multicast-protocol for the one-hop routing but an inter-group unicast protocol for the multi-hop routing. In the inter-group routing, the master node can be a main arbiter (or broadcast relay node) to transfer the broadcast group information and also relay broadcast packets to other master nodes in other groups, if there are direct connections among the nodes. Of course, some nodes of the satellite nodes can also be the arbiters if they meet the requirements of the broadcast relay node. The broadcast group is basically maintained by the Broadcast Resource Scheduler (BRS) to manage the wireless channels and have the system more scalable and reliable.

The reason the system approaches the group-based broadcast operation is to help release the distresses caused by the inherent characteristics of broadcast, *"spontaneous,"* *"redundant,"* and *"unreliable."* For example, the approach provides the applicability of a reliable transmission protocol to infallibly distribute broadcast packets to as many nodes as possible, just with less effort, because of the decrease of receivers replying with ACK messages. Also, the group-based neighbor knowledge used to maintain the broadcast group prevents the spontaneous traits of broadcast. Finally, the inter-group routing scheme removes the redundancy of re-broadcast in the multi-hop routing.

3.3.2.3 Broadcast resource scheduler

Each node has its own Broadcast resource scheduler (BRS) function. The role of BRS is to effectively manage the limited wireless radio resources through use of Broadcast Resource Schedule Protocol (BRSP). Its ultimate goal is to make the broadcasting practically work in wireless P2P networks: One example that shows it is the neighbor knowledge scheme in broadcast transmission which prevents the collision caused by the blind broadcasts. Fundamentally, the BRS function of each node is achieved by the cooperation with BRSs of other nodes, and internally has two functionalities such

Broadcast Queue for the packet scheduling and Neighbor Finder for the group management.

Broadcast resource schedule protocol

BRSP is a distributed protocol with centralized management approaches: As other distributed protocols, its operations are performed by interactions with BRSP of other nodes within a virtual ad-hoc cell. However, the interactions concentrate on approaching centralized management schemes such as the synchronization of broadcast queues for the broadcast resource management and the utilization of a master node within a virtual ad-hoc cell for controlling the broadcast queues.

The protocol has two core management functions: The first function is the resource management used to transmit or route broadcast packets and to allocate the wireless channel evenly or time to time with a priority. As discussed in the previous section, it has two different routing schemes. In the one-hop routing, group-based broadcast transmission is used to achieve fast delivery. In the multi-hop routing case, the inter-group unicast transmission is preferred to prevent the redundancy caused by unnecessary re-broadcasting. With respect to the channel allocation, the BRSP makes use intelligently of the neighbor knowledge method by synchronization of resource schedulers, broadcast queues each node has. The second one is the group management to find neighbor nodes for building a broadcast group and to decide fittingly a master node within a broadcast group, in case of the inexistence of the master node due to host mobility and failure.

The neighbor knowledge method is actualized by creation of the centralized resource scheduler, *Virtual Broadcast Queue* within a virtual ad-hoc cell. The protocol makes four processes to build the distributed broadcast queue into a centralized resource scheduler: sensing, grouping, synchronizing, and updating. In chapter 4, we take a much closer look at the management functions and the operations of virtual broadcast queue of BRSP protocol.

3.3.2.4 Event collection server

The principal role the event collection server plays is of mining events locally created in a social event network. Specifically, the event mining is divided into three steps: gathering, categorizing, and locating events. First, the server collects the event information from the master node of each virtual ad-hoc cell, whenever any event comes about. Next, the collected events can be categorized based on the previously decided classifications or newly created ones. Finally, the locations where events occur are determined through Triangular Address Sharing (TAS) method to estimate a location as utilizing three shared infrastructure IP addresses. The event collection and the location decision methods are further described in the succeeding subsection.

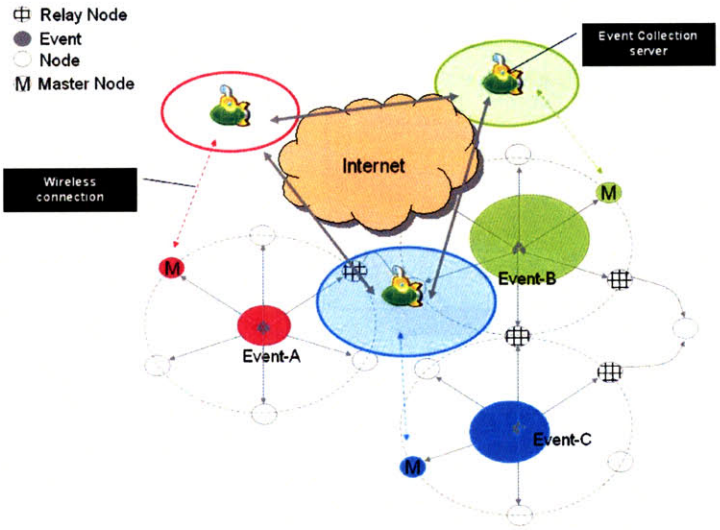


Figure 19. The topology of Event Collection Server for global and local events information

In the topology facet, at least, an event collection server is deployed in a social event network it is in charge of to perform the event mining. The server has two networking interfaces: One is for connecting to infrastructure systems, namely, it is to communicate with event collection servers in other networks, via Internet, for global event information as shown in Figure 19. The other is to wirelessly communicate with the master nodes in the virtual ad-hoc cells for local event information. Generally, fixed or nomadic nodes could be the event collection servers if they have the two network interfaces. Note that

the nodes dominated by traits of high mobility would not be appropriate as the event collection server.

The event server runs an application, *Social Event Navigator (SEN)* that enables users to search, to check, and even to join real-time events occurring in local areas where they stay or are interest in. Figure 20 shows the interface of Social Event Navigator. As shown in the figure, the interface shows what kind of social events are occurring in a specific area in the real time manner, helps the users interact with the events they are intrigued with, and even predict any feasible changes from the events.

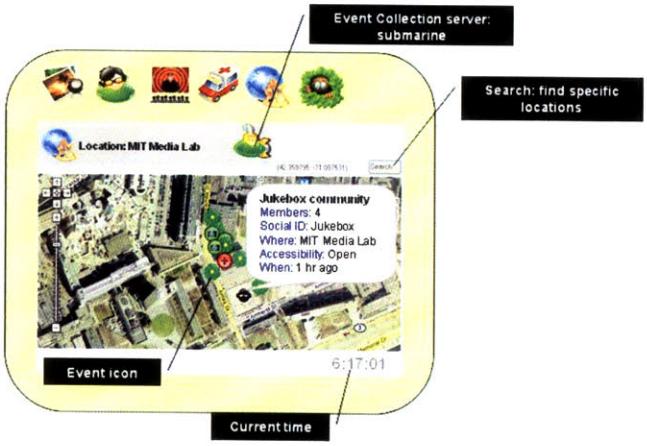


Figure 20. The user interface of Social Event Navigator.

Event collection and location decision

This section describes how the event collection server gathers events from personal broadcast nodes and then how it also decides the locations of the harvested events.

Whenever a broadcast group generates any event, the master node within the group sends an event server, which governs the social event network it pertains to, the information messages corresponding to the event via wireless ad-hoc/mesh networking. In the case, the event message includes the following information: "*how many nodes join the event,*" "*what kind of event happens,*" "*where it is occurs,*" "*whether it is open to the public or not,*" and "*when it comes about,*" as shown in Figure 20. The event server replies with an

ACK message when receiving the event message. Afterward, it collects those event messages in real-time manner, categorizes the received events into some families (e.g., jukebox, talk station, etc.), and displays icons matched to the events over the user interfaces of Social Event Navigator.

To decide the locations where events bring about, the event server uses Triangular Address Sharing (TAS) mechanism: When receiving an event info message, the event server checks how many other event servers can be connected to the master node which sent the message. We note that an event server has a list of master nodes it can communicate with and shares the list with other event servers. Through such the exchange of the list of master nodes, each event server can know how many other event servers the master node sending an event message is connected to. And then, the event server chooses at least three other event servers to acquire the infrastructure IP addresses of potential locations the master node could stay. Afterward, the event server transforms the IP addresses to the location data based on Domain Name Server enabling geographical location information [49] [50]. Finally, the event server selects one of feasible locations the event occurred. This method does not decide an exact location like GPS (Global Positioning Service), but could be applicable in real networks.

In general, it is not easy to find the geographical location of a host, given its IP address. For the reason, IP addresses are allocated arbitrarily, as there is no inherent connection between an IP address and its physical location, and there is no reliable method to do the trick. Some helpful experimental work has been performed, however. The most feasible trials are to define a DNS Resource Record to contain the geographical location. There are two methods: One is to use a DNS extension which allows hosts to enter their geographical location into their DNS record, based on an extension to DNS described in [49]. Another attempt to express a host's geographical location via DNS is done in [50]. However, these studies are out of score in the thesis. Since the IP address to location data transition still remains outstanding, the location decision is a future study.

3.4 Applications for Social Event Networking

In this section, we illustrate modular applications the XCast system provides to actualize the social event networking.

XCast system encourages the users to perform novel collaborative broadcast activities by simultaneously using some functionalities of participatory broadcasting applications it offers. The system supports two kinds of participatory applications for the cohesive broadcast activities: One is for community-based broadcasts, and the other is for individual broadcasts. The former applications lead the users to first build a community for the broadcasts. An example that shows it is multimedia files sharing. The applications are generally used for the entertainment or the information sharing. With respect to the individual broadcasts, the applications do not require any group or community for the broadcast activities. The examples we readily consider would be emergency notification and commercial advertisements. These may make the users tune frequently broadcast channels over their own personal broadcast station. Also, it may have them face external disruptions such as spam and overloading.

The system offers four conceptive modular applications for community-based broadcasts: "XC*Gallery*" for image file sharing, "XC*Library*" for in-style documentation sharing, "XC*Jukebox*" for song file sharing, and "XC*Talk*" for a small talk station. For personalized (or non-group) broadcasts, XCast serves an application to immediately deal with emergency situations: "XC*911*." For the social engagement purposes, the system provides a fundamental application to ferret out neighbors and to support social group activities. We call it *"IVY Networks."* As described in the previous section, the system also helps the users explore various events locally created or occurring through the "*Social Event Navigator*." The descriptions on each application are given in the succeeding subsections.

3.4.1 Design considerations

The applications are designed to follow two principal targets the system pursues. We remind of it, as follows.

One is to permit people to create personalized communicators, "broadcast stations" over mobile devices, for extemporaneous events or individually motivated presentations. The other is to provide people with a cognitive platform for social event awareness that informs what is happening around them and then timely coordinates the events.

To achieve this, first, the participatory applications play a fundamental role of "broadcast stations" for sharing in-style information and presenting imminent situations. Next, to support the awareness of social events, the applications utilize the functions such as the neighbor finding and the social group management via IVY Networks. Finally, for the actualization of the social event networking, they integrate the event mining application, Social Event Navigator, as described in Section 3.3.2.4.

With the design rationale, the whole structure of applications in the system is shown in Figure 21. The all applications are organically structured in the system. For instance, as shown in the figure, all participatory applications start to run after finding neighbors or building a social group through IVY Networks. The Social Event Navigator works to coordinate the local events while the participatory applications provide the users with a playground to perform the collaborative broadcast activities. Unlike the participatory applications, XC911 usually independently runs without building any social group. To support all functions of the applications, we design Application Management Function. It is comprised of Neighbor Finder, Social Group Builder, and Social Broadcast Queue modules. These functional modules cohesively work with those of the BRSP protocol in the lower layer to perform a broadcast operation. Section 5.1.3 describe the details.

The application structure enables the users to run one more applications at the same time. Specifically to say, they can simultaneously make use of some functions from each

application over the application framework. For instance, while broadcasting or sharing song files by XCJukebox, the users within a community can have a conversation with each other through the talk station of XCTalk. Through such integrations and interactions among the applications, users are encouraged to perform new cooperative broadcasting activities.

Furthermore, the community-based applications adopt Social Resource Scheduler method to efficiently allocate social broadcast resources to the users like BRSP protocol in the low level, as described in Section 3.3.2.3. One example that clearly shows it is the XCJukebox application. The application makes intelligently use of a social broadcast queue to build a music queue for the *virtual jukebox* scenario, as introduced in Section 3.1. The social broadcast queue plays a pivotal role of preventing Spam resulted by the blind broadcasting. Note that actually, the social broadcast queue tightly interacts with the Virtual Broadcast Queue created by Broadcast Queue Management function of the BRSP protocol to schedule the broadcast packets fittingly. We describe the detailed interaction in Section 5.1.3.

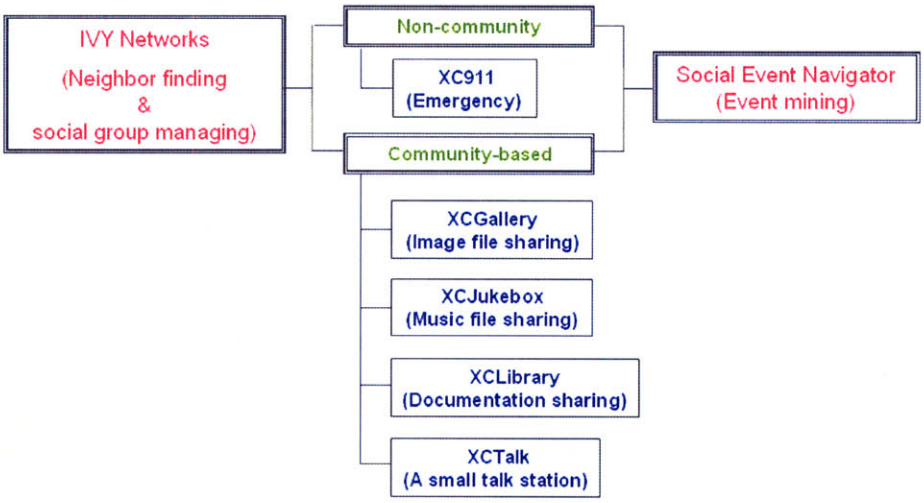


Figure 21. The structure of XCast applications

3.4.2 IVY Networks: neighbor finder and social group management

IVY Networks[3.4] has two roles, the neighbor finder and the social group management. As the neighbor finder, it has a neighbor viewer that shows how many "XCast users" exist in surroundings as shown in Figure 22. Actually, finding neighbors is wrought by the periodical beacon signals, "Heartbeat" sent in the low level of the system. For the social group management, the IVY Networks utilizes a function, Social Group Manager (SGM) that allows users to create a community they want. The function help the users build a community through the exchanges of invitation messages. That is, SGM broadcast invitation Messages to peers and distributes social group IDs if a user want to build a group.

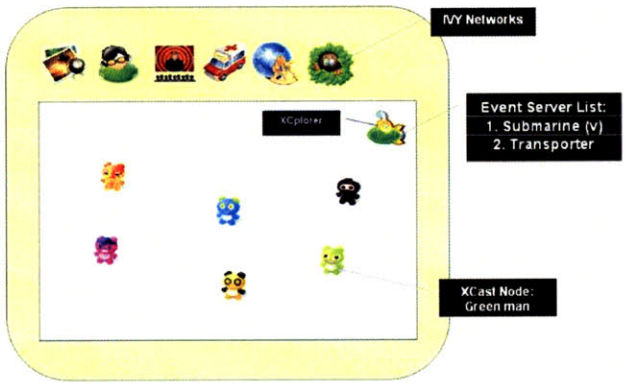


Figure 22. The neighbor view of IVY Networks

3.4.3 Participatory applications

This section gives a short description for each participatory application. Figure 23 shows the interface of XCast applications.

- XCGallery is a community-based application that helps users share image files (e.g., pictures) in real time and serves multiple slideshows depending on the

(3.3) IVY means the evergreen ivy leaf and thus IVY Networks conceptually represents "mesh networks."

number of members of a community. For the sharing, it uses the social broadcast queue.

- XCLibrary is also a community-based application that creates a network for in-style documentation sharing. It allows user to share information on the documentations seasonably read within a local community. The application works under the assumption is that people would like to know the trend of information-what is going on.

- XCJukebox is a community-based application designed to actualize the virtual jukebox scenario. It is a virtual music player that enables users to broadcast their favorite song files to others within a community in orderly ways, like the jukebox: Users possess a common music queue built by the social broadcast queue method and then can stack song files they want to share on the music queue. A song plays out at a time from the music queue.

- XCTalk is a talk station that allows people to freely communicate with the members of local community.

- XC911 is an emergency communicator that helps people be connected to local helpers, police stations, or 911 stations in emergency cases.

Note that in this thesis, I focused on further developing the two participatory applications for the multimedia file sharing, XCGallery and XCJukebox. Figure 24 shows the two applications. Also, the interfaces of XCast applications are designed for children who use OLPC (One Laptop Per Child). XCast interfaces on OLPC are colored with the bright such as light yellow, sky-blue, light green for the children. The whole concept of design on "XCast over OLPC" is to help children easily build a community, collaboratively create community events, and deal with various events occurring around them, without a glitch.

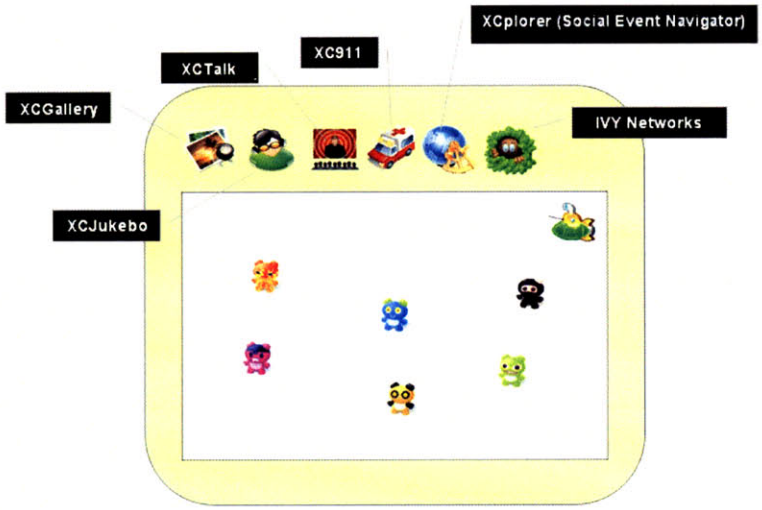


Figure 23. The interface of XCast applications

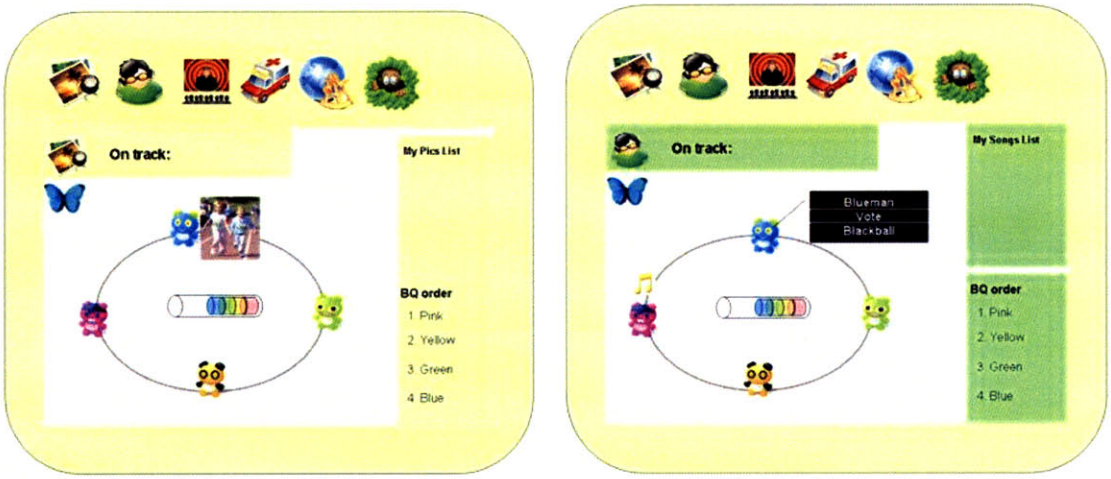


Figure 24. The interfaces of XCGallery and XCJukebox

CHAPTER 4. BROADCAST RESOURCE SCHEDULE PROTOCOL

In the chapter, we take a closer look at the core protocol of the system, Broadcast Resource Schedule Protocol (BRSP). Its goal is to build the collision-susceptible wireless network into the collision-less (even or –free) broadcasting space. To achieve this, BRSP pays attention to the cognition ability to be aware readily of "what is going on the surroundings," especially for the broadcast transmission. The succeeding sections describe the details of the underlying principles behind the system and of the algorithm the protocol takes for that purpose.

4.1 Design considerations

In the design of BRSP, it has two principal concepts, "cognition" and "cohesiveness." First, the cognition means to previously sense the occupancy rate of channels supposed to be used for the broadcast transmission. It is to previously know if there is any possibility of that broadcast transmission is performed over the same channel, at the same time and thus that packets are collided. On the other hand, the cohesiveness means that all operations are achieved by collaborative interactions among the neighbor nodes as in other distributed systems. We apply the concepts to tackle the problems caused by the inherent traits of broadcasting in wireless ad-hoc/mesh networks.

4.1.1 Cognitive network

The BRSP re-creates a wireless ad-hoc/mesh network into a cognitive network for the reliable broadcast transmission. That is, the cognitive network resolves the issues on the blind broadcasting while making all nodes cognizant of their surrounding operations.

In the cognition concept, BRSP is a protocol for building a virtual ad-hoc cell, a basic unit of the cognition network. Within the virtual ad-hoc cell, we need to look at how the BRSP protocol makes all nodes have the cognitive intelligence. In the virtual ad-hoc cell,

the resource scheduler of each node has its own queue as a sensor to recognize what is going on in other nodes for the packet transmission. All the queues are synchronized while they share the information of their own queues with each other for the monitoring or sensing method. Now, each node can acquire all changing status on the queues of other nodes in the real-time manner. That is, it knows the order of broadcasting within the cell: who first sends and receives the broadcast packets, and also when the node itself can send the packets. After the synchronization is completed, a virtual broadcast queue is established to practically manage all the broadcast transmissions within the cell.

In addition, the BRSP recognizes two different routing zones for the broadcast transmission: one-hop "broadcasting" and multi-hop "unicast" routing zones. In other words, for the multi-hop transmission, the BRSP uses two different transmission protocols according to the routing zones: First, it utilizes the broadcast protocol in one-hop routing zone, the virtual ad-hoc cell, and afterward, it is to use unicast transmission protocol between the virtual cells. The reason it uses unicast among the virtual cells for the purpose of broadcast transmission is to reduce the redundancy caused by unnecessary re-transmission of broadcast packets in the multi-hop routing.

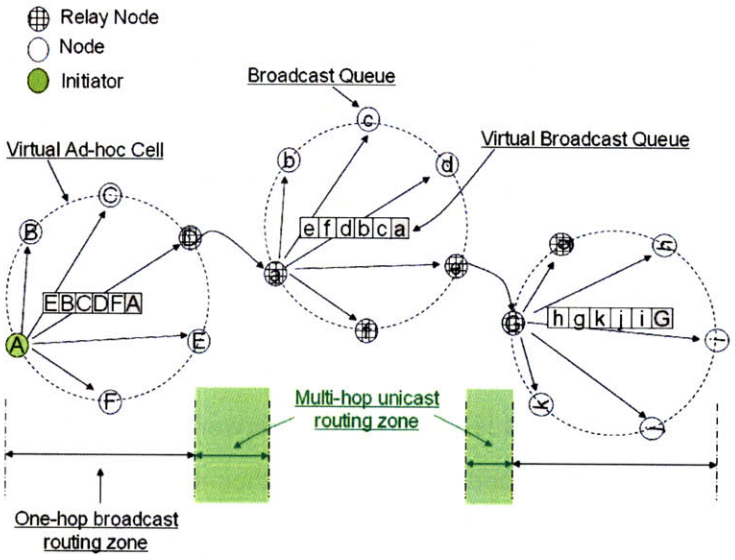


Figure 25. The topology of a cognitive network: Virtual Ad-hoc Cell, Routing Zones

Through such cognitive processes, the BRSP results in evolving an ad-hoc network into a cognitive network in the basis of the virtual ad-hoc cells. This helps resolve the blind broadcasting. Figure 25 shows the topology of a cognitive network and the virtual ad-hoc cells it consists of.

4.1.2 Cohesive behaviors

The BRSP works in a distributed network. It means that all nodes would collaboratively interact with each other for an operation. To build the ad-hoc network into a cognitive network, the BRSP protocol has some collaborative processes.

- The first collaboration is to build a virtual cell in the broadcast operation, as discussed in the previous section.
- The second one is to create a centralized resource scheduler, named *Virtual Broadcast Queue* based on the distributed queuing system.
- The third cohesive step is to schedule broadcast packets from the virtual broadcast queue.
- The final activity is collaboration among arbiters to replay broadcast packets to other groups.

Actually, a virtual broadcast queue plays a crucial role of a distributed yet centralized scheduler within a virtual cell. That is, all the nodes in the cell cooperate with each other to re-build the physically distributed queuing system into a virtually centralized scheduler. To achieve this, some internal processes are made in collaborative ways.

Synchronization: This make the distributed scheduler perform centralized operation. For the synchronization, BRSP performs the following two processes: sharing and updating. First, sharing is a process that all nodes within a virtual cell recognize each other and hold their own transmission status in common. That is, they cooperatively open their own information to share with other nodes. Next, updating is a process that all nodes within the virtual cell make a change on their own queue when a new operation is required. In

the end, synchronizing the queuing system requires all nodes to together take part in the sharing and the updating processes at the same time.

Scheduling: Practically, to transmit the broadcast packets, the virtual broadcast queue uses some scheduling methods such as First-In-First-Out (FIFO) and weight fair queuing (WFQ) [53, 54]. Individual broadcast queues react to the scheduling methods the virtual broadcast queue adopts. Logically, each queue then play a role of a queue slot of the virtually centralized queue. That is, it should not try to transmit its broadcast packets with other nodes at the same time: When one forwards, others should wait. Such operations are wrought by collaboration among nodes in the cell.

Routing: As described in the previous section, the cognitive network the system creates has two routing zones. The cohesive operations in one-hop broadcast routing zone correspond to those in the virtual ad-hoc cell. In the multi-hop unicast routing zone, an arbiter (broadcast relay node) interacts with other arbiters at different groups; the arbiter shares its own queue info with other arbiters to decide when to re-transmit the broadcast packets.

4.2 How does it work?

The operation of the BRSP protocol is divided into three main processes, 1) creating a virtual broadcast queue, 2) transferring broadcast packets, and 3) routing the packets to nodes in other groups. Additionally, BRSP has a mechanism to support the host mobility.

4.2.1 Functional structure

The BRSP protocol constitutes of four functionalities to perform the three principal operations: group management, broadcast queue management, routing management, and mobility management. Figure 26 show the functional structure.

Group Management: This function is mainly in charge of finding neighbor nodes, creating a broadcast group, dubbed a *"Virtual Ad-hoc Cell."* Actually, the BRSP protocol starts its operations through this function. This makes intelligently use of periodic beaconing signal messages to look for neighbors in surroundings and to maintain a broadcast group.

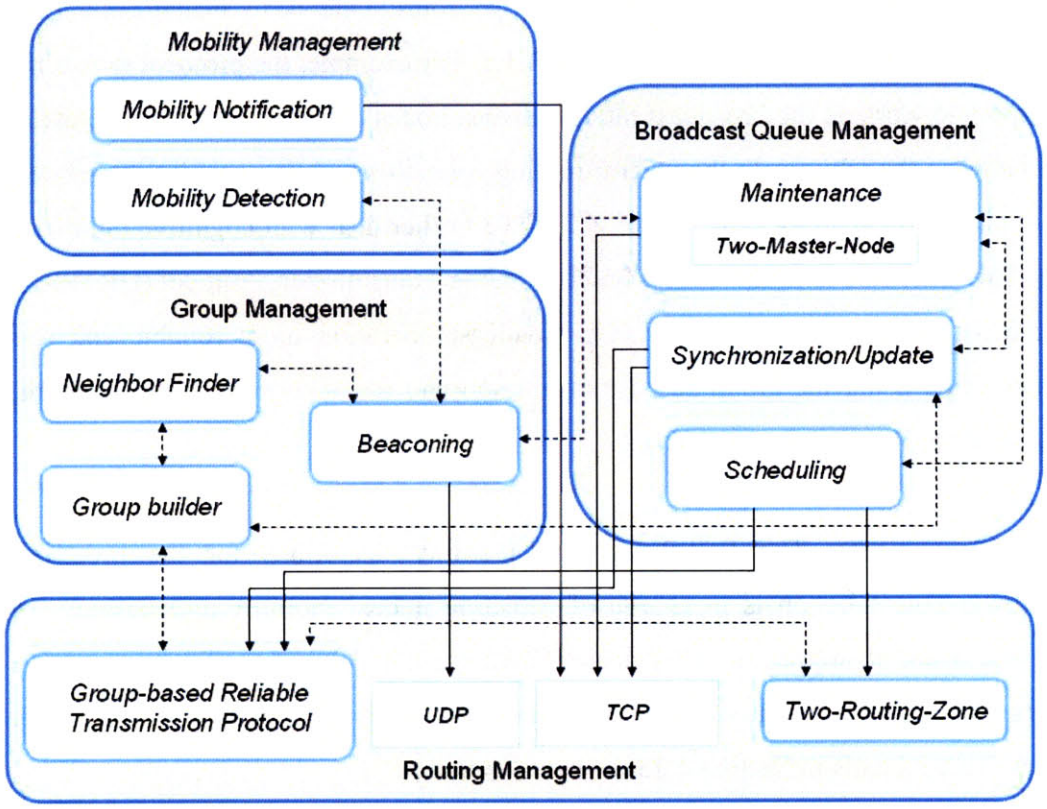


Figure 26. BRSP functional structure

Broadcast Queue Management: It internally consist of three core modules such as synchronization/update (Syn/Up), scheduling, and maintenance. Its main role is to create a logically centralized queuing system for the management of broadcast transmission, named "Virtual Broadcast Queue" and then to maintain its operations correctly without any failure in wireless ad-hoc/mesh networks. Specifically, the Syn/Up module synchronizes the status of all queues in a virtual ad-hoc cell and updates all the information if any change is made, in a wirelessly coordinated way. Next, the scheduling module sends out broadcast packets from the virtual broadcast queue in orderly or

preferred ways. Since transforming distributed queues into a centralized queuing system, this functional module is required to have a few super nodes to maintain the system: It applies a "two-master-node" approach, and the more detailed descriptions are given in Section 4.26. In the end, the maintenance module manages the super nodes and supports them to systematically collaborate with each other.

Routing Management: This functionality has two roles: One is to manage the two-routing-zone protocol, as described in Section 4.1.1. For example, the protocol makes the routing module work as the broadcast mode between nodes and an arbiter but operate as unicast between the arbiters. In the unicast routing zone, this function enables the arbiters to be cognizant of each other. In Section 4.2.5, we further deal with cognitive multi-hop routing. The other role is to run Group-based Reliable Transmission Protocol (GRTP) for the broadcast transmission. GRTP makes broadcast operation more reliable and less redundant while using the group-based transmission and the acknowledgment methods. We take a closer look at it in Section 4.2.4.

Mobility Management: The BRSP supports mobility operations for the broadcast transmission. This function is in charge of detecting nodes' mobility and failures. To achieve the roles, it applies a mobility notification method which utilizes two active notification messages and two broadcast group-dedicated beaconing messages. We discuss the more details in Section 4.2.6.

4.2.2 Control messages

In the section, we provide a capsule overview of what control signal messages are applied to the BRSP protocol, before going through the BRSP signaling operations to be described in the succeeding sections. Some messages organically work with others or are re-used to achieve an operation,

Three beacon signal messages: The BRSP uses three beaconing messages in order each node to sense other nodes and to address its existence to them. The use of three separate

beacon signal messages helps increase the efficiency and versatility of the protocol operations.

- *Heartbeat_Join (HJ)*: This beacon signal is generally used as heartbeat for a node's existence before a broadcast group is built.
- *Heartbeat_Group (HG)*: After a group is created, satellite nodes within a virtual cell make use of this signal. Also, it is used to detect nodes' breakdown or mobility by the master node.
- *Heartbeat_Group_Master (HGM)*: The heartbeat is just for detecting whether the primary master node is alive or dead. That is, it is use for the secondary master to monitor the status of the primary master node or for an outside node to contact the master node to join the broadcast group.

Two messages for group creation: The BRSP adopts two main control messages to build a broadcast group.

- *RequestForList (RFL/RFL_ACK)*: This message pair is utilized to ask other nodes the potential group list they collected for the establishment of a group and to reply for the request.
- *RequestForGroup (RFG/RFG_ACK)*: After gathering the group lists, nodes use this control message pair to initiate a broadcast group

Three signal messages for sharing and updating the queue information: After a broadcast group is established, the master and the satellite nodes make use of the following control signals to synchronize or update the status of broadcast queues in the group.

- *RequestForQueue (RFQ/RFQ_ACK)*: As soon as a group is created, the master broadcasts RFQ message to the satellite node to ask them to send their queue status and the satellites reply with RFQ_ACK.
- *SynForQueue (SFQ/ SFQ ACK)*: The master transfers SFQ to synchronize all the queue status and the satellites reply with the ACK.

- *RequestForUpdate (RFU/ RFU_ACK)*: A node sends RFU to the master node to notify the update of its own queue, and then the master replies with RFU_ACK.

One signal message for joining a broadcast group: An outside node utilizes to ask a master node to allow its joining of the broadcast group the master manages.

- *RequestToJoin (RTJ/ RTJ_ACK)*

Four signal messages for checking nodes' failures and mobility: The BRSP supports some methods to check out a node's system failure or mobility. The control signals are used for that purposes.

- *TransitionToMaster (TTM/TTM_NACK/TTM_ACK/NULL)*: The BRSP uses a "two-master-node approach-the primary/secondary master nodes-to manage the centralized queuing scheme, Virtual Broadcast Queue. The secondary node initiates TTM message to replace the primary node.
- *AskForLive (AFL)*: The master node sends AFTL message to a potential mobile node to check if the node is alive or still stay at the group.
- *NotificationToLeave (NTL/NTL_ACK)/ConfirmationToLeave (CTL/CTL_ACK)*: These are duded "NTL/CTL mobility pair." That is, the messages are used to support routing in the mobile networks. Also, NTL is used for a node to notify the master node of his leaving from the group.

4.2.3 Virtual broadcast queue

Include a figure show over all procedures of signaling to create a virtual broadcast queue.

To build a virtual broadcast queue, the protocol makes four procedures: sensing, grouping, sharing, and updating. As the first step, the sensing is to be cognizant of how many neighbor nodes exit in one-hop routing distance, namely over the radio signal

coverage the network device of system supports. The procedure of message flows is given as follows. Figure 27 shows the process of sensing.

- Each node periodically broadcasts *Heartbeat_Join (HJ)* beacon signals to detect neighbor nodes that it can directly communicates with via one-hop routing. *"HJ"* beacon signals are sent as a short packet in a bit or byte. In the case, for the transmission of beacon signal, the BRSP uses User Datagram Protocol (UDP). The beacon signal consists of a tuple of (IP address: Host ID). The beaconing period is set five seconds in the system. Note that the time makes UDP-based beacon signal acceptable in the wireless network due to reducing redundancy of broadcast packets.

- After some time (at the timer) passes, each node creates a group list and afterward shares the list with other nodes to sends a *RequestForList (RFL)* message by TCP-based multicast protocol. If a node receives RFL and accepts the group list sharing, it replays with *RFL_ACK*. Setting the timer is decided based on experimental approaches. In the system, we set the timer to ten seconds. That is, after receiving the beacon signal two times, each node makes a group list for building a broadcast group. In the case, the group list consists of a set of the tuples of (IP address and Host ID).

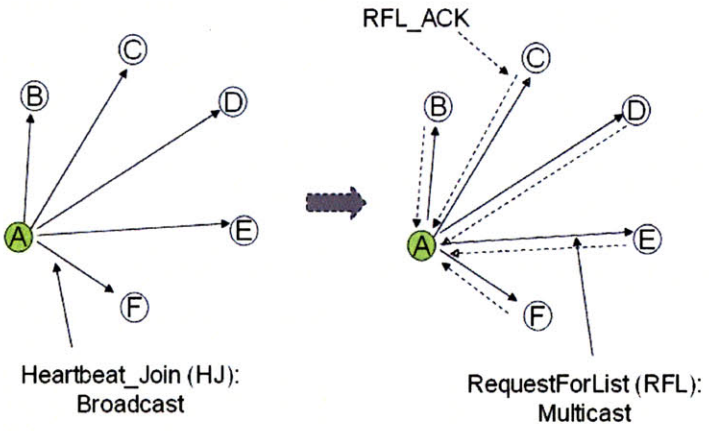


Figure 27. Sensing signaling for Virtual Broadcast Queue

Next, the grouping is a process for creating a virtual ad-hoc cell, and Figure 28 shows the procedure of exchanges of control signaling messages.

- After exchanging the group lists, each node compares its own group list with the other nodes' group lists received and then selects nodes with the same group list. That is, a group is built by the nodes with the same group list.

- A node with the longest prefix address become a primary master node to organize a broadcast group and initiates a group. The master node sends a *RequestForGroup (RFG)* message to other nodes and then the remaining nodes reply with an *RFG_ACK* message if the received *RFG* message has the same group list.

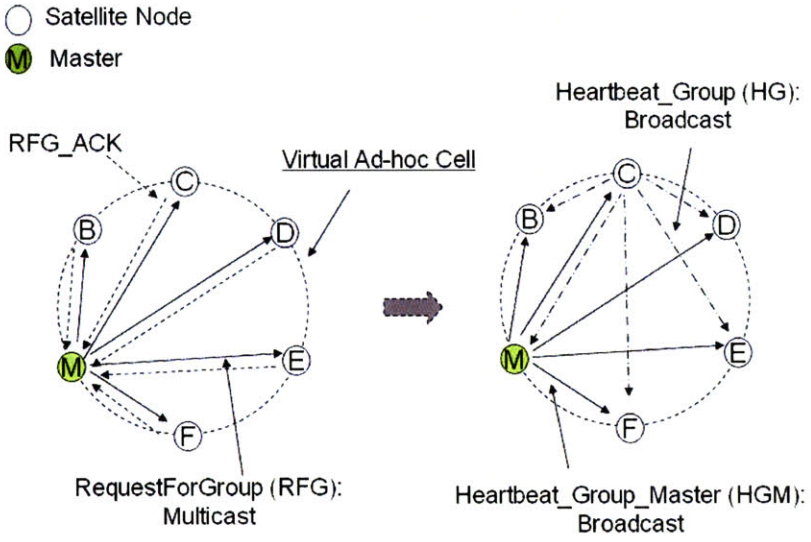


Figure 28. Grouping signaling for Virtual Broadcast Queue

- Finally, a broadcast group, a virtual ad-hoc cell is created. Afterward, all satellite nodes within the virtual ad-hoc cell broadcast *"Heartbeat_Group (HG)"* beacon signal instead of *Heartbeat_Join (HJ)*. On the other hand, the master node uses *"Heartbeat_Group_Master (HGM)"* beacon signal. HG and HGM beacon signals constitute of a triple of (Group ID: IP address: Host ID). We note that the transition of beacon signal helps increase efficiency of the protocol operations.

For example, a node wishing to join a group could distinguish a master node from satellite nodes through the *HGM* beacon signal. Also, the use of two separate beacon signals helps immediately recognize which node fails, the master or the satellite, due to breakdown or host mobility.

The sharing process is for building a virtual broadcast queue, after the sensing and grouping are completed. The procedure of control signaling is shown in Figure 29.

- First, the master node decided in the grouping process starts to send other nodes *RequestForQueue (RFQ)* messages to ask them to share the broadcast queue information, if any. The satellite nodes reply with *RFQ_ACK* including their own queue info, after receiving the *RFQ* messages.

- As soon as receiving the *RFQ_ACK* from the satellite nodes, the master node updates its own queue with the collected information. It then decides the order of broadcasts based on *"who first joins the group"* or *"who has data to be broadcast"* or *"who has priority to first be sent."*

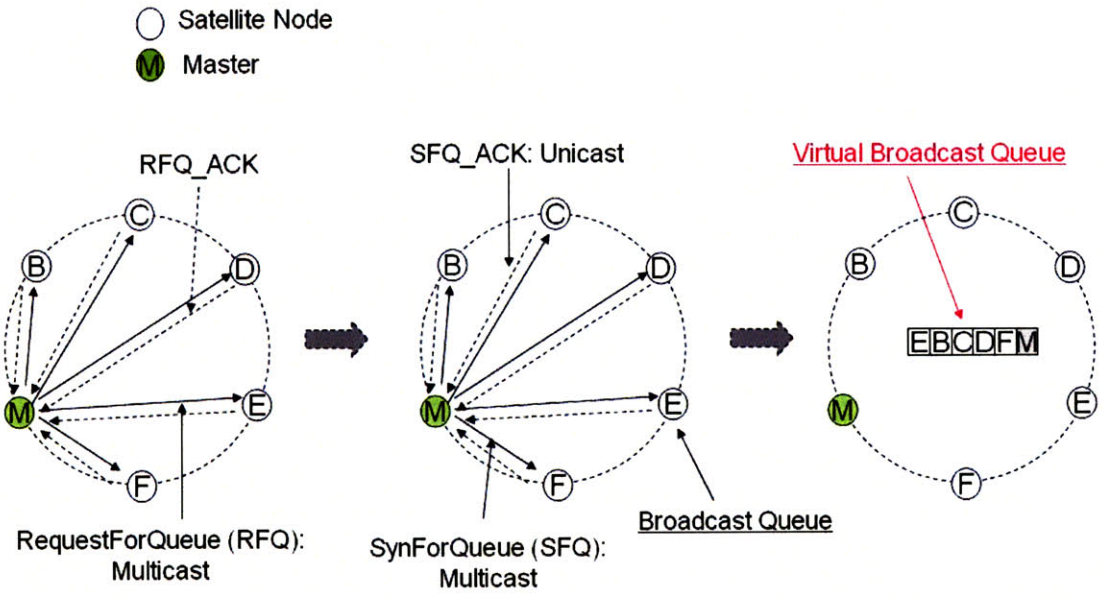


Figure 29. Sharing signaling for Virtual Broadcast Queue

- After the update, the master node sends its own updated queue information to the other nodes by reliable multicasting. For that, it sends *SynForQueue (SFQ)* message, and then the satellite nodes reply with *SFQ ACK* messages after updating their own queue with the information received from the master node. Now, a virtual broadcast queue is created!

Updating is performed whenever there occurs any update on the queues of nodes within the virtual ad-hoc cell. This is a process to keep the synchronization of broadcast queues. This process is shown in Figure 30.

- A node with any update information sends *RequestForUpdate (RFU)* message including the update to the master node. Then, the master node updates its own queue status with the newly received queue information and afterward responds with *RFU_ACK* message. In the case, if the node wishing to update has a higher priority, the order of scheduling packets on the virtual broadcast queue changes as shown in Figure 30.

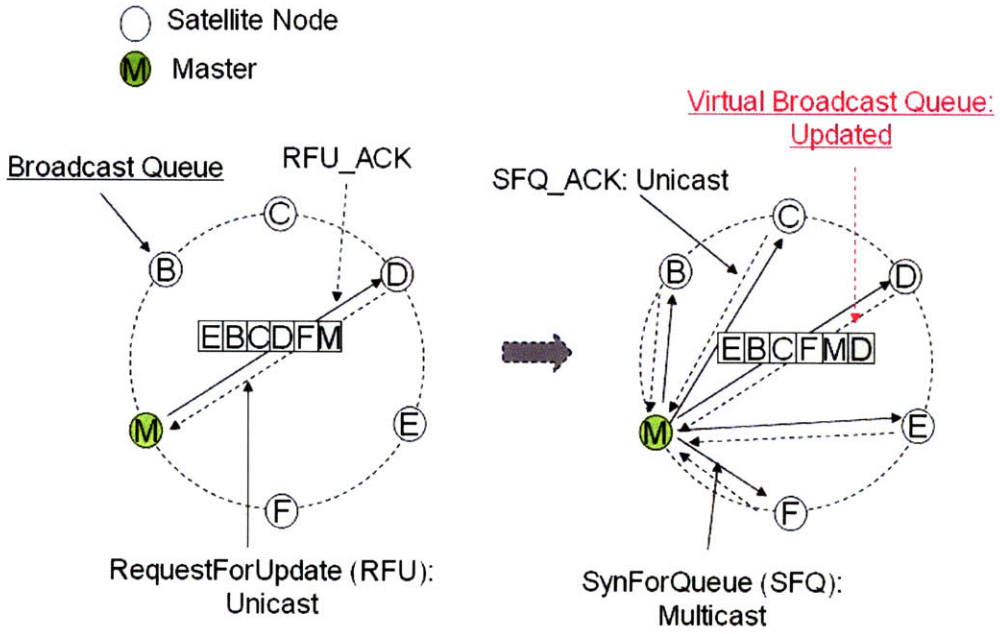


Figure 30. Updating signaling for Virtual Broadcast Queue

- To synchronize the updated queue information with the other nodes in the group list, the master node sends the satellite nodes *SynForQueue (SFQ)* message via reliable multicast transmission. After receiving the updated information from the master, the satellite nodes also update their own queue with the received information and then reply with *SFQ_ACK* messages. Synchronization of the virtual broadcast queue is kept.

4.2.3.1 Joining in a broadcast group.

This section describes the procedure of when a node joins the already established broadcast group. Figure 31 shows the signaling procedures.

- First, if an outside node wants to join a broadcast group, it checks if there is any *Heartbeat_Group_Master (HGM)* beacon signal in the area it stay to contact the master node at the group. If the node receives *HGM* beacon signal, it sends a *RequestToJoin (RTJ)* message to the master node. The master replies with an *RTJ_ACK* if receiving the RTJ message and accepting the request.

- After receiving the ACK, the node wishing to join sends its own queue information to the master node. For that, it uses *RequestForUpdate (RFU)* message. That is, the remaining procedure is the same as the updating process described in the previous section. The master node renews its own queue info as adding the newly received queue info into its queue after receiving *RFU* message and sends RFU_ACK message to the new member. Note that with respect to the order of the scheduling, the newly joining node has the lowest priority if it is no urgency.

- Finally, the master notifies the group members of the joining of a new node while multicasting the updated queue information to the satellite nodes. In the case, it sends *SynForQueue (SFQ)* message, and then the satellite nodes reply with *SFQ_ACK* messages.

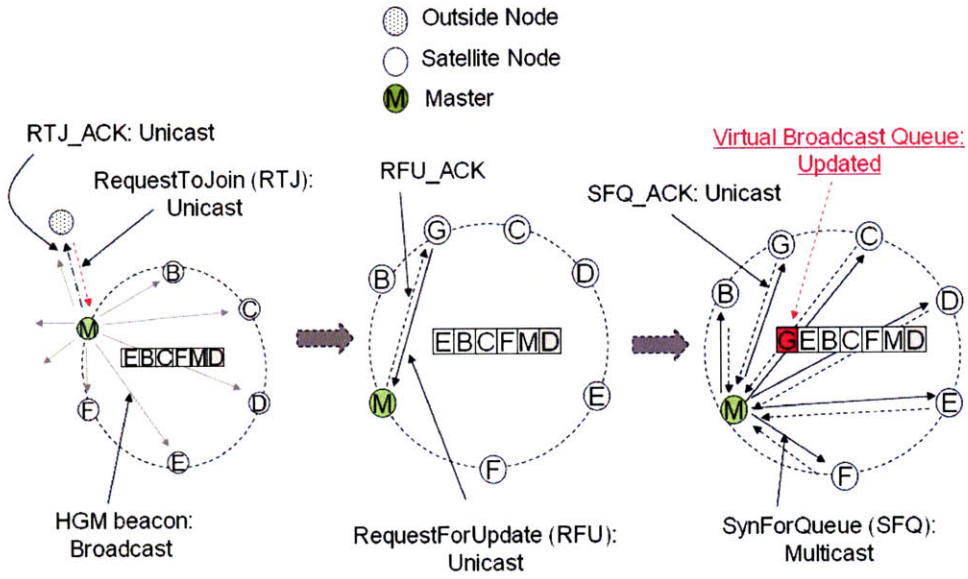


Figure 31. Signaling for joining the broadcast group

4.2.3.2 Leaving from a broadcast group

This section describes the procedure when a member leaves at the group. This process is also applied to support mobility scenarios. Section 4.2.7 gives the more details.

- Before a node leaves at the group it has joined, it sends a *NotificationToLeave (NTL)* message to the master. The master node simply replies with *NTL_ACK* renewing its queue information. The NTL message could be triggered by a module of the group management function. Triggering the NTL message, however, is a challenge. That is, deciding when to initiate the NTL message remains an outstanding issue.

- After receiving the *NTL_ACK* from the master node, the node secedes from the group. At the same time, the master notifies the remaining nodes of a member's leaving sending them *SynForQueue (SFQ)* message including the updated queue information. After receiving the SFQ message, the remaining satellite nodes update their own queues with the information received and then reply with *SFQ_ACK* messages.

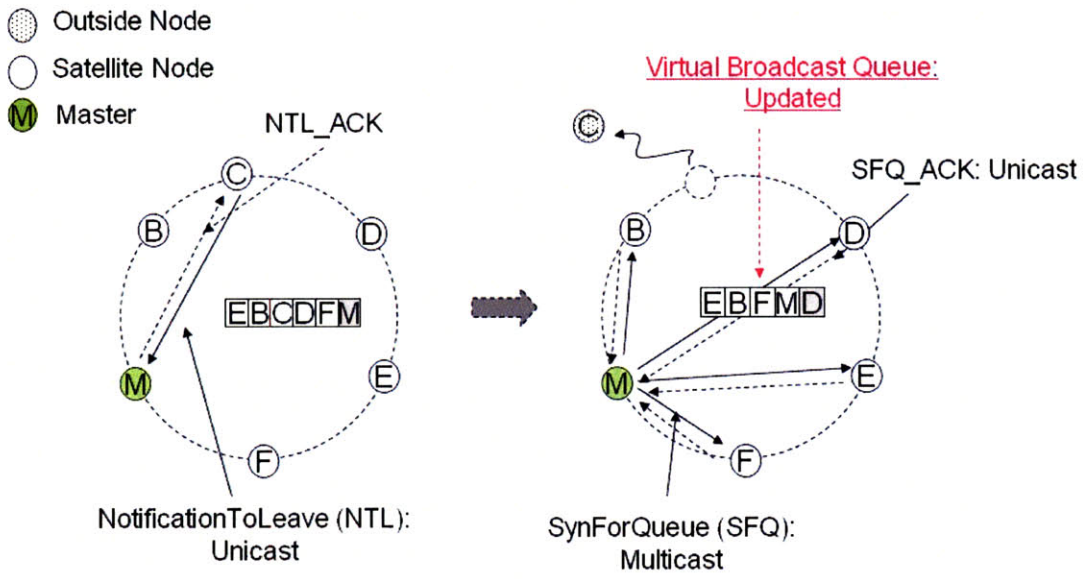


Figure 32. Signaling for leaving from the broadcast group

4.2.4 Group-based reliable transmission protocol for broadcasts

In a virtual ad-hoc cell, the broadcast transmission is performed based on the group. That is, it is the same as a multicast transmission. With the group-based broadcasting, we can consider adaptation of a reliable transmission protocol to infallibly distribute a broadcast message to as many hosts as possible, just with less effort. For the reason, the group-based broadcasting decreases the number of receivers responding with the acknowledgements, provided that the transport protocol makes intelligently use of the acknowledgement method for the reliable packet transmission. This approach lessens the effects on the issue of overhead caused by the use of acknowledgement in the transmission protocol for broadcasts.

The BRSP uses a broadcast transmission protocol with TCP-like behaviors, based on the unit of a broadcast group it built. We call it "Group-based Reliable Transmission Protocol (GRTP)." GRTP has multiple point-to-point connections with acknowledgement message like TCP mechanism. It makes use of the group list made in the process of Sensing, the first step to build the virtual broadcast queue. As soon as a broadcast group is established,

GRTP is used by the master node to send multiple satellite nodes signaling messages. As well, it is utilized by all nodes to transmit broadcast packets within a virtual ad-hoc cell.

4.2.5 Cognitive multi-hop routing

Each group has an arbiter (broadcast relay) node which can be connected to other arbiters at different groups or even be a member there. There are two scenarios that an arbiter is connected to other groups: The first scenario is an arbiter-to-arbiter connection. This is the case that the arbiter directly communicates with arbiters at other groups in basis of point-to-point (Unicast) connection for the broadcast transmission. The other scenario is an arbiter-to-multiple node connection. The case it works is that the arbiter belongs to other group at the same time. The arbiter re-broadcasts the packets to its other members at the different group through GRTP protocol. Those scenarios are shown in Figure 18 (of Section 3.3.1).

The BRSP makes intelligently use of the inter-arbiter routing for the multi-hop routing, as shown in Figure 25. In the broadcast multi-hop transmission, if an arbiter receives broadcast packets from a node at the same group but the target nodes of the packets exit at different group, the routing protocol at the arbiter transforms broadcast into "unicast" to just relay the broadcast packets to arbiters at different groups. As soon as receiving the camouflaged broadcast packets like unicast, the receiving arbiters at different groups check if the broadcast packets are already received or not: The BRSP adopts Counter-based scheme to exam acceptability of the duplication. This is further discussed while analyzing the algorithm efficiency of BRSP in Section 5.1.2. If the received broadcast packets are not duplicated with the previously received ones, the arbiters multicast that packets to their members. It results reducing the overhead caused by unnecessary re-transmission of broadcast packets in the multi-hop routing.

4.2.6 Virtual Broadcast Queue management

There are two approaches to manage the virtual broadcast queue: One is to use a centralized method, with a master node which is in charge of controlling operations of the virtual broadcast queue. The advantage it has is to make the protocol less complicated as reducing the number of message flows. However, it has a potentiality of the single point failure, especially caused by host mobility in the mobile networks. The other is to apply a distributed management scheme without a super node. In other words, all the power and the intelligence to control the broadcast queue are evenly distributed to all nodes within the broadcast group. Since all nodes try to be responsible for the management, there are quite many exchanges of control signal messages. In general, this makes it difficult for us to design or implement the protocol due to the increased complexity. However, it avoids the issue of single point failure.

In the protocol, we use the centralized management scheme with a master node because of preference of the philosophy, *"The simplest is the best."* However, we also consider the use of multiple master nodes to release the distresses by the single point failure. Here, we propose a centralized management scheme with multiple master nodes. The details are as follows.

- At first, we could select the number of the potential master nodes proportional to the size of the broadcast group. In the stage, to decide the number of mater nodes, we temporarily apply *Pareto Principle* [52], also known as the law of the vital few stating, "For many events, 80% of the effects come from 20% of the causes." For example, in the group consisting of ten nodes, two master nodes are chosen. However, the applicability of the principle remains a further study.

- In this thesis, practically we use a "two-master-node" approach regardless of size of the broadcast group assuming that a broadcast group has no high population. The two master nodes are chosen by the longest prefix address matching, as described in Section 4.2.2. For example, a node with the longest prefix address

becomes the primary master node and the other node with the second-longest prefix address is the secondary master. Alternatively, two master nodes could be selected in the basis of who is first served in scheduling broadcast packets, the order of broadcast.

- The primary node is responsible for managing the broadcast queues. As an alternative node, the secondary node, on the other hand, monitors status of the primary node. That is, it checks if the primary node fails or leaves at the broadcast group. If the secondary node assesses that the primary node fails to play a role of the master node, it becomes the primary master node, instead of the existing one.

- As the base of the assessment, the secondary node monitors if it successfully receives the "Heartbeat_Group_Master (HGM)" beacon signal from the primary node. If it does not receive the successive three beacon signals, it judges that the primary master fails and then sends TransitionToMaster (TTM) message to the primary node. If the primary node replies with TTN_NACK message, no transition happens. Otherwise, the secondary node becomes now the primary node. For example, if not receiving any response from the primary node until the period of beacon signal (e.g., five seconds) or getting TTN_ACK, the secondary node plays a role of the primary master node instead. The management function decides that the primary node would not work temporarily if it sends TTN_ACK. Figure 33 shows the signaling flows for the transition to the primary node.

- In case of failure of the existing primary master node, caused by the host mobility or the breakdown of system itself, the management function selects a new secondary node, dependent on the longest prefix address matching (or the highest order of broadcast scheduling). Again, one with second longest prefix address (or the second priority in broadcast scheduling) of the satellite nodes becomes the new secondary master node. Whenever the primary master fails, the procedures above repeat.

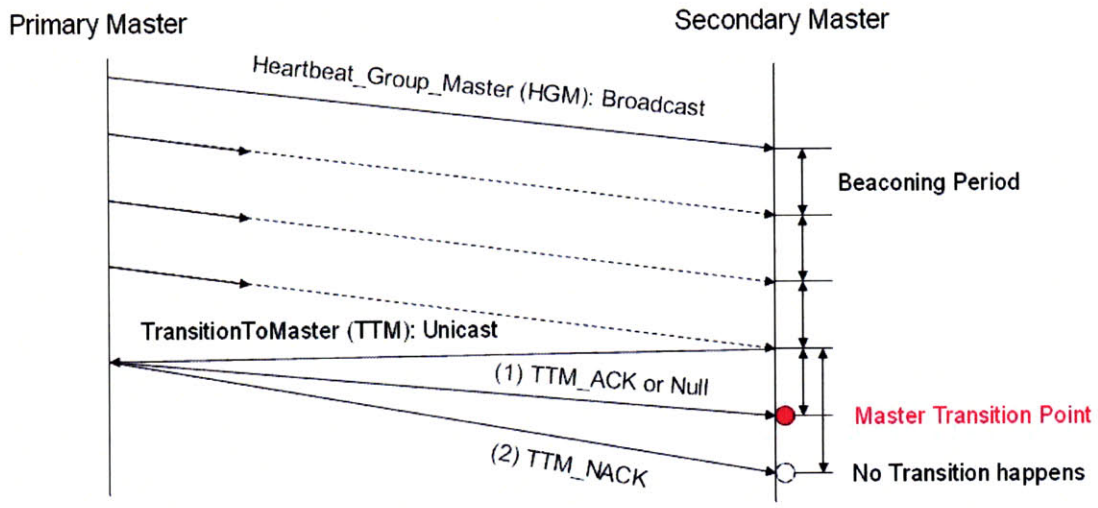


Figure 33. Signaling for the transition to the primary master

4.2.7 Mobility support

This section describes how the BRSP supports mobility scenarios. To face the disruptions of the broadcast group caused by the host mobility, the BRSP protocol uses a Mobility Notification method which utilizes three messages such as *NotificationToLeave (NTL), Heartbeat_Group (HG),* and *ConfirmationToLeave (CTL)*. There are three ways to check the host mobility out by using each message.

- The first one is to use an active notification message before leaving: If a satellite node wants to leave at a group, it first sends *NotificationToLeave (NTL)* message to the primary master. The primary master node replies with *NTL_ACK* message to confirm the leaving. Then, the wishing node changes the beacon signal, *"Heartbeat_Group (HG)"* into *"Heartbeat_Join (HJ)."*

- The second method is to use a passive notification: This utilizes the beacon signal to check the host mobility out. If the master node does not receive the successive three *"Hearbeat_Group (HG)"* signals from a node, without previously receiving the NTL message, it assumes that the node left abruptly at the group and then,

sends *AskForLive (AFL)* message to the node. Again, if the master does not receive any reply from the node until the period of the beacon signal, it judges that the node is estranged from the broadcast group. Note that the master's mobility could be checked out through the "Heartbeat_Group_Master (HGM)" signal, as discussed in the previous section.

- The third scheme is to transfer an active notification after leaving: If a node leaves at a group without sending any NTL message to the master node and then joins a new group, it sends *ConfirmationToLeave (CTL)* message to the old master node via the multi-hop routing. Provided that the old master node receives *CTL* message from the node supposed to leave at its group, it replies with CTL_ACK and confirms the vanished node's mobility.

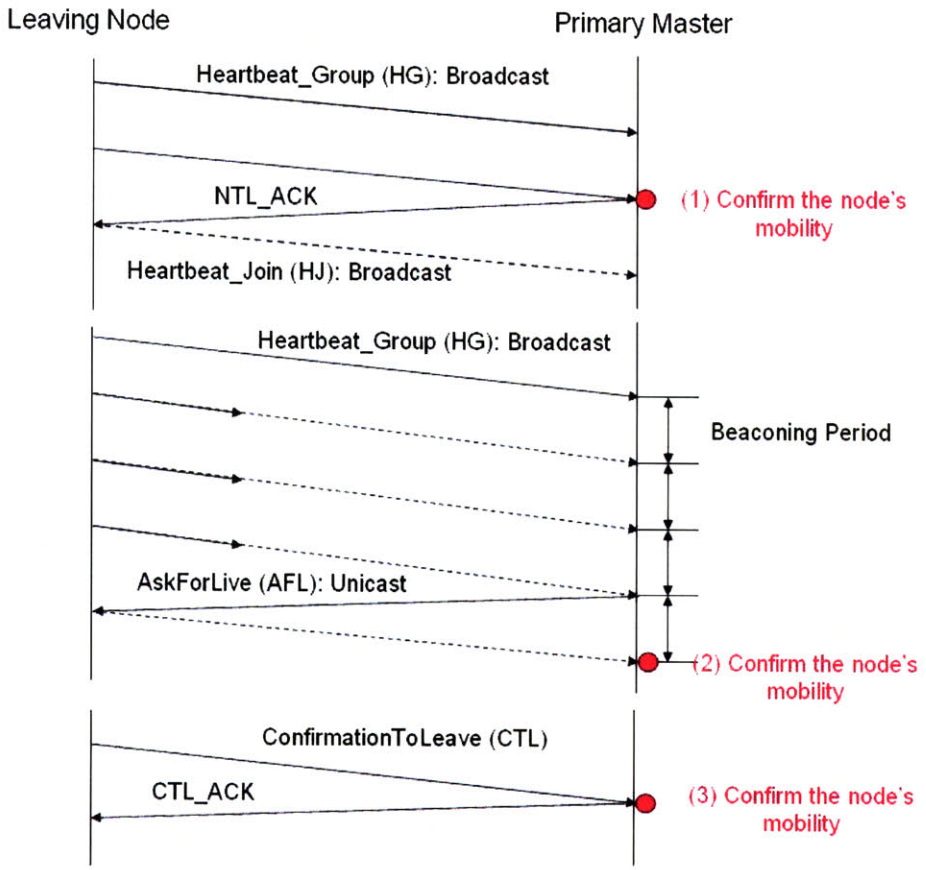


Figure 34. Signaling for checking the host mobility

After the checkout of host mobility, the master removes the node's slot from its own queue and sends *SynForQueue (SFQ)* message to other satellite nodes to synchronize all queue status within the virtual ad-hoc cell. The remaining satellite nodes reply with *SFQ_ACK* messages after updating their own queues.

The two active notification schemes above can be integrated to support the multi-hop routing according to the host mobility: We call it *"NTL/CTL mobility pair."* For instance, the leaving node-A notifies the master of its secession from the broadcast group with *NTL* message and then is away from the group. However, the master node expects to receive a note on the mobility from the moving node-A and then holds broadcast packets needed to route the node-A. Afterward, the node-A joins in other group and sends the old master *CTL* message. Now, the master node transmits the broadcast packets to the node-A to successfully finalize the routing. Due to the dynamic behaviors of mobility, however, the schemes above may not perfectly support various scenarios of mobile networks. Also, the decision of when to send NTL message is an outstanding issue, the mobility support still remains a further study.

CHAPTER 5. EVALUATION

This section focuses on how we implemented the XCast system, what experiments we have performed for the assessment of the system, and what lessons we learned for the implementations and experiments.

5.1 Implementations

We had two implementation strategies for the XCast system: One is to test the implementation on a pilot hardware platform, VIA mini-ITX board, and the other is to implement the system on a real ad-hoc/mesh networking platform, OLPC XO. On the pilot platform, we focused on incubating the initial prototypes and developing a testbed for a wireless ad-hoc network to evaluate the system. Afterward, we transformed the nurtured prototype into a standalone platform for broadcasts on the OLPC XO laptop to scrutinize its applicability in a wireless ah-hoc/mesh network.

5.1.1 Two Pilot Testbeds

We constructed two different ad-hoc networks constituting of pilot ad-hoc nodes in chronological order: The first testbed was created to share the image files through the broadcast transmission in a simplified ad-hoc network(5.1), as shown in Figure 35. The testbed consists of five static ad-hoc nodes and one robotic mobile ad-hoc node. To build a pilot ad-hoc node, we equipped the mini-ITX board with Netgear *madwifi* wireless network card and installed a Linux operating system, *Gentoo* on the board. The reason we selected Gentoo as the operating system for the mini-ITX board is that it can be automatically optimized and customized for just about any application or need and also shows high performances on the computing device not providing high horsepower, like the mini-ITX board, thanks to Gentoo's package management system called *Portage*.

(5.1) Actually, the testbed was built for the XCast demo at the Media Lab Sponsor Week, in May, 2007.

The robotic mobile node was created by collaboration among four graduate students at different groups of the Media Lab[5.2]. To construct the framework of the mobile node, we selected a *Roomba* robotic platform [55] for the mobility and integrated the mini-ITX board into the platform as shown in Figure 36. The robotic mobile node has three functions: The first function is to track the line on the floor and then take pictures at the previously decided several spots. The second one is to make it possible for the mobile node to broadcast the generated pictures to five wireless ad-hoc nodes. Finally, the function the robotic node has is to avoid any collision from objects in the front of its body. These functions help the robotic node play a crucial role of an authentic mobile node to broadcast image files. Figure 37 shows from the design of the mobile node to the actualized feature. The static ad-hoc nodes, on the other hand, while receiving broadcast packets from the mobile node, also take photos via webcams installed on the top of their monitors, exchange the produced photos to each other via the broadcast transmission, and finally displays all the photos they generated and received.

Figure 35. An initial testbed of XCast system for the image file sharing: It consists of a robotic mobile ad-hoc node and five fixed wireless ad-hoc nodes.

(5.2) Junki Lee (Personal Robotic Group), Sanghoon Lee (Physical Language Workshop Group), and Leonard Bonanni (Tangible Interfaces Group) were in charge of creating a line-tracking robot, developing a collision-avoidance system for the robot, and designing the superficial body, respectively.

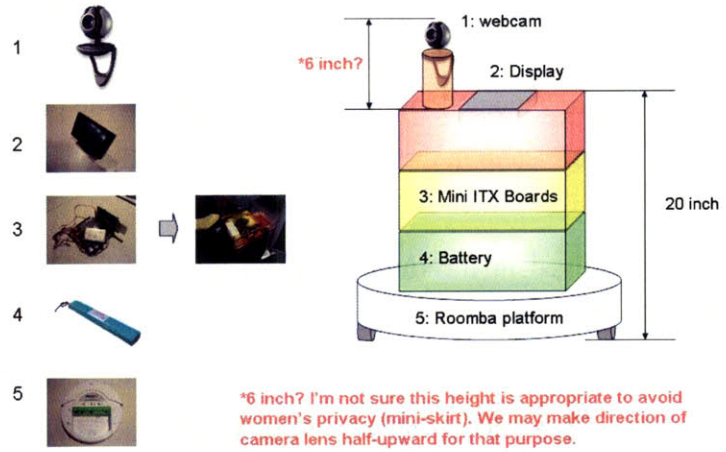


Figure 36. Framework of the robotic mobile ad-hoc node.

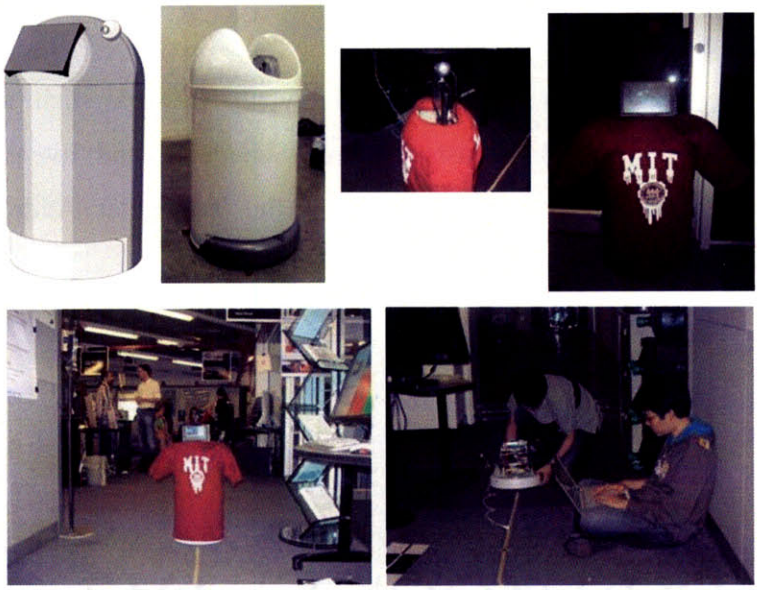


Figure 37. The robotic mobile ad-hoc node, from the design to the prototype.

Over all the nodes, we implemented the initial version of the BRSP protocol. That is, Syn/Up and Scheduling modules of the Broadcast Queue Management were partially implemented, only Neighbor Finder module of the Group Management was fully implemented, and GRTP was emulated by a protocol utilizing multiple point-to-point transmissions. With the functions, the virtual broadcast queue coordinates all broadcast operations to prevent collisions among the broadcast packets. Each node makes

intelligently use of the transmission protocol with the acknowledgement for the broadcast transmission, without building any broadcast group. In the testbed, the functionalities of XCast system were not fully implemented, but the developed prototype provided a gauge to assess the applicability of the system in the simplified wireless ad-hoc network.

The second testbed was deployed to incarnate the framework and components of the Social Event Network (as described in Section 3.3) and to further develop the BRSP protocol. It focused on sharing both the image and audio files through the broadcast transmission in a more simplified ad-hoc network. The testbed is shown in Figure 38. Based on the first testbed, the second ad-hoc network was comprised of three wireless ad-hoc nodes and one wireless event collection server: The ad-hoc nodes developed in the first testbed were re-used, and the event collection sever was constructed on a desktop PC equipped with a wireless network interface card (e.g., Wi-Fi card). In the testbed, we implemented all the components of the Social Event Network such as Personal Broadcaster, Broadcast Group, Broadcast Resource Scheduler, and Event Collection Server by Python programming language, even if the Triangular Address Sharing protocol of Event Collection Server was fully not installed.

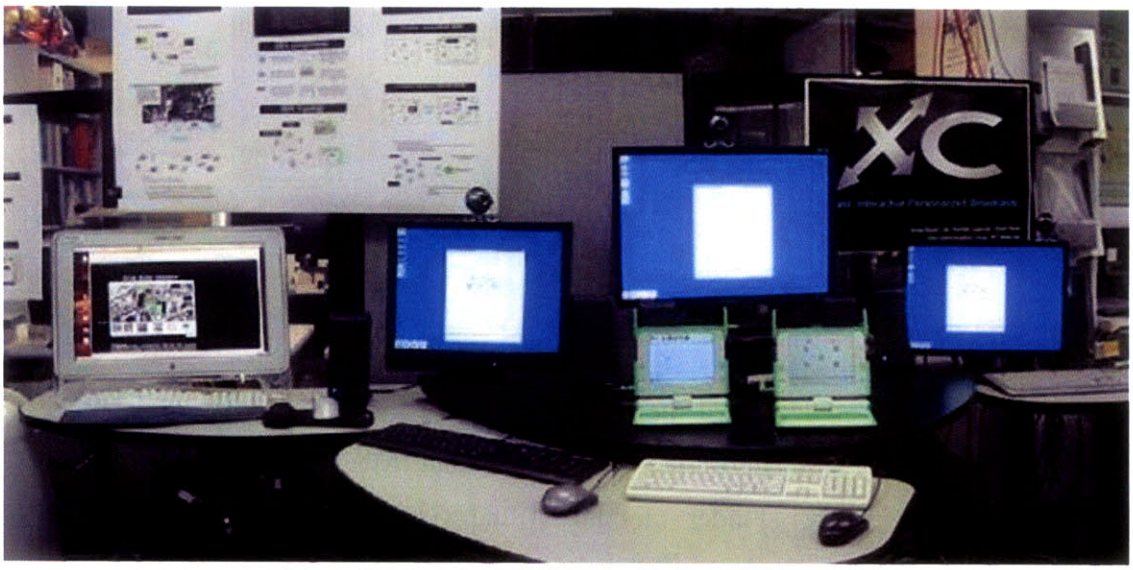


Figure 38. The second testbed of XCast system for sharing image and audio files: It consists of an event collection server and three wireless ad-hoc nodes.

We also developed two modular-based participatory applications on each ad-hoc node to share image and audio files in the basis of the functions of the BRSP, respectively. As well, to find neighbor nodes and manage a social group, a modular application dubbed *"IVY Networks"* was developed in the testbed. At first, as described in Section 3.4, the IVY Network is initialized to automatically ferret out neighbors in surroundings and build a social networking group for the file sharing purpose. Afterward, each participatory application runs to support each file sharing based on the socialize group. We first implemented the participatory application for the image file sharing, named XCGallery. Figure 39 shows the initial user interfaces of the application. The XCGallery can display multiple images broadcast from other nodes and also generated from itself, at the same time, That is, it simultaneously shows multiple slideshows for the images: This is a real example for the broadcast-based image sharing. Next, we developed the event-mining application, Social Event Navigator to incubate the Social Event Network. Figure 40 shows the initial version of Social Event Navigator running on the Event Collection Server. It was developed based on HTTP, AJAX, and Java script languages including Google Map API [56] and operated on World Wide Web independent on the Participatory application. Whenever the events on the photo sharing happen at the ad-hoc nodes, the server receives the event information messages from them, categorize the events, and display the corresponding event icons on the geographic locations. However, the full implementation of Event Collection Server still remains a future work due to the outstanding issue of TAS protocol as discussed in Section 3.3.2.4.1.

Figure 39. The interfaces of the participatory application for the image file sharing

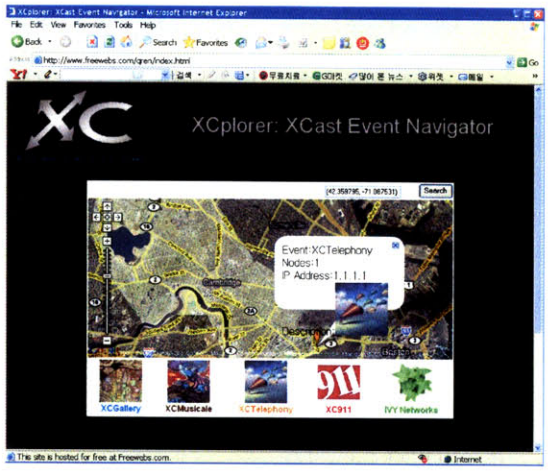


Figure 40. The initial version of Social Event Navigator

With respect to the BRSP functions, in the testbed for the image file sharing, we implemented the Syn/Up and the Scheduling modules of the Broadcast Queue Management, the Neighbor Finder and the Group builder modules of the Group Management, and the multicast transmission protocol with acknowledgements as the GTRP protocol of the Routing Management. Through the testbed, we could make sure that the BRSP protocol enables the image file sharing to reliably work with the broadcast transmission.

5.1.2 OLPC XO platform

For the audio file sharing, we totally re-designed the interfaces of the modular applications to follow the concepts of OLPC XO as shown in Figure 41. XCast interfaces on OLPC are colored with the bright such as light yellow, sky-blue, light green for protecting the children's eyes. The whole concept of design on "XCast over OLPC" is to help children easily build a social group with their local friends, create collaborative events based on the group, and coordinate various events occurring around them, without a glitch, using ad-hoc/mesh networking technologies. We also re-designed the web-based standalone the Social Event Navigator into a modular application which can be integrated to other modular applications such as IVY Networks, XCGallery and XCJukebox, for the

OLPC XO version. The interface is shown in Figure 42. Each function of the Social Event Navigator was implemented as following the descriptions in Section 3.3.2.4.

Before installing XCast system over the Fedora Linux O/S-based OLPC XO, we first make the best use of the second testbed to previously evaluate the re-designed modular applications because it is quite tricky to directly develop any application on the XO. Afterward, additionally we also implemented remaining functions of BRSP protocol such as Mobility Detection module of Mobility Management, Two-Master-based Maintenance module of Broadcast Queue Management, and Two-Routing-Zone protocol of Routing Management. However, Mobility Notification module, called "NTL/CTL mobility pair" of Mobility Management was not implemented due to the issue on the unintelligible decision time to send NTL as discussed in Section 4.2.7. The mobility support of the BRSP protocol requires a further study. Note that we adopt Gstreamer library [57] to process the audio packets in the XCJukebox.

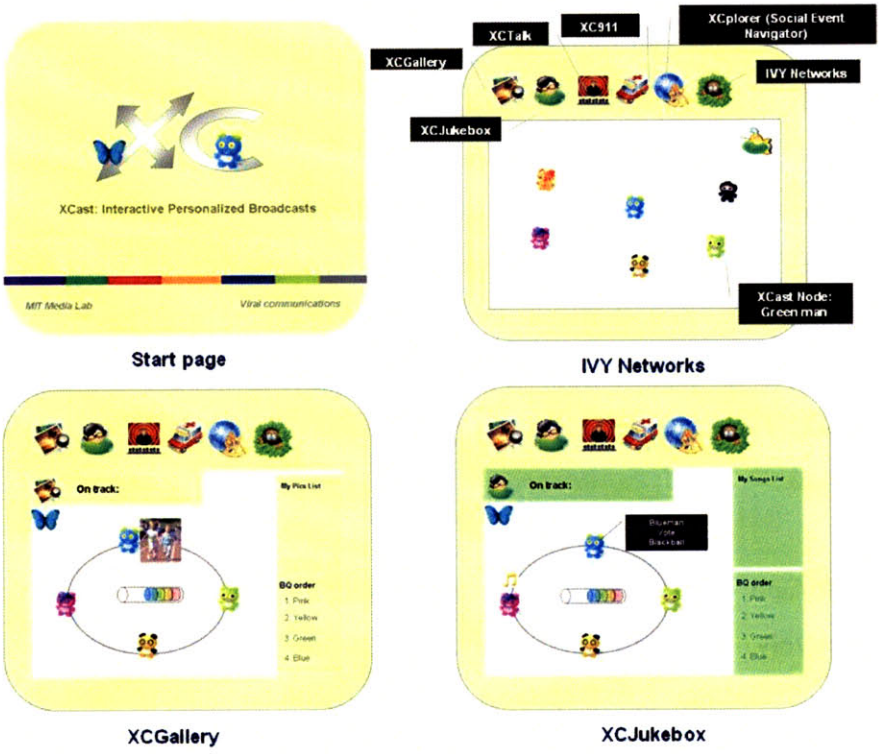


Figure 41. The user interfaces of modular applications for OLPC XO.

Figure 42. The user interfaces of Social Event Navigator for OLPC XO

Figures 43, 45, 46, 47, 48, and 49 show the interfaces of XCast system implemented on OLPC XOs. First, Figure 43 shows starting XCast application on Sugar user interface of OLPC XO and configuring the "mesh network" interface (e.g., the red circle changes into the green one). To port XCast on Sugar, we created an icon representing the broadcasting feature of XCast system as shown in Figure 44.

Figure 44. XCast icon

Next, the initial pages of XCast application and the interface of IVY Networks are shown in Figure 45. The IVY Networks interface has a neighbor view; as soon as the application starts, the neighbor view first shows a node icon representing itself as shown in the figure; different node icons are added whenever the corresponding nodes join the ad-hoc/mesh network. The node icons display the IP address and the host name of the corresponding node.

Third, Figures 46 and 47 show the interfaces of Neighbor Finding and Social Group Building, respectively. To ferret out neighbor nodes, the Neighbor Finder module of Group Management in the BRSP protocol uses beacon signals to periodically be broadcast via UDP. In the case, we set the period of beacon signals with five seconds. The change of neighbor view is shown in Figure 46 whenever the XCast application runs on each XO.

Figure 43. Starting XCast on Sugar user interface of OLPC XO

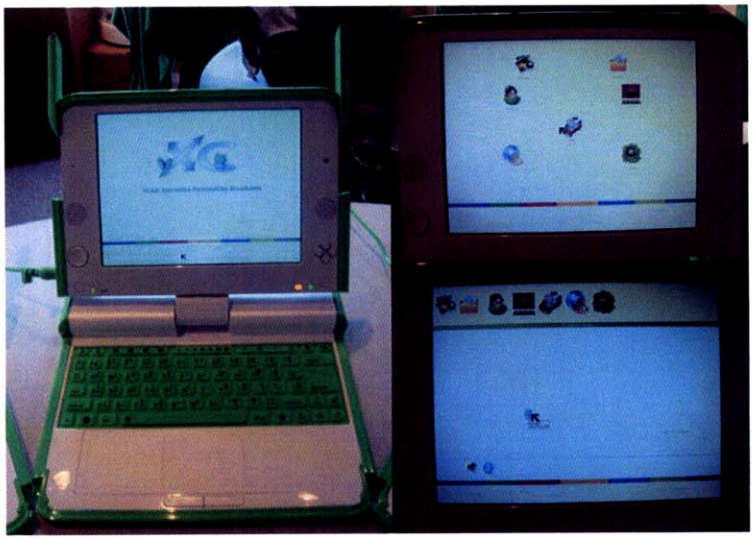


Figure 45. XCast interfaces on OLPC XO

After finding the neighbor nodes, Application Management of XCast system helps users build a group for a sociable ground to share multimedia data with their locals. Note that we need to distinguish the social networking group from the broadcast group. First, the

social networking group is built by *"Social Group Builder"* module of the Application Management according to the users' preferences. That is, they can select a group they want to join. The broadcast group, on the other hand, is automatically created by the *"Group Builder"* module of Group Management in the networking layer to play a fundamental role of a unit of the broadcast transportation, as described in Section 3.3.2. Figure 47 represents the functionalities of Social Group Management. To build a social group, a node broadcasts an invitation message to other neighbor, and then the social networking group is organized if the other nodes accept the invitation as shown in Figure 47 (a) and (b). The circle means a social group, and the butterfly icon on the top left of the IVY Networks interface depicts that the Social Broadcast Queue is established. In the case, all operations are synchronized via the social broadcast queue. Also, after XCJukebox application runs, the social group still remains the same and thus is re-used for the file sharing, as shown in Figure 47 (c) and (d).

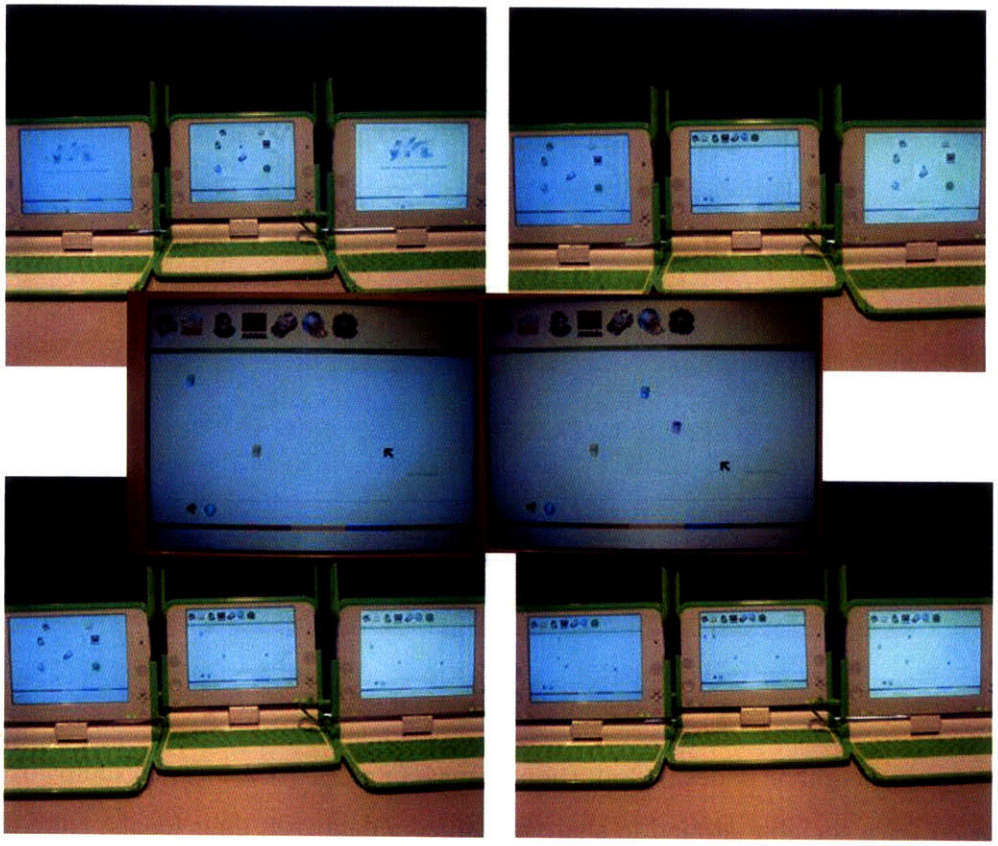


Figure 46. Neighbor Finding

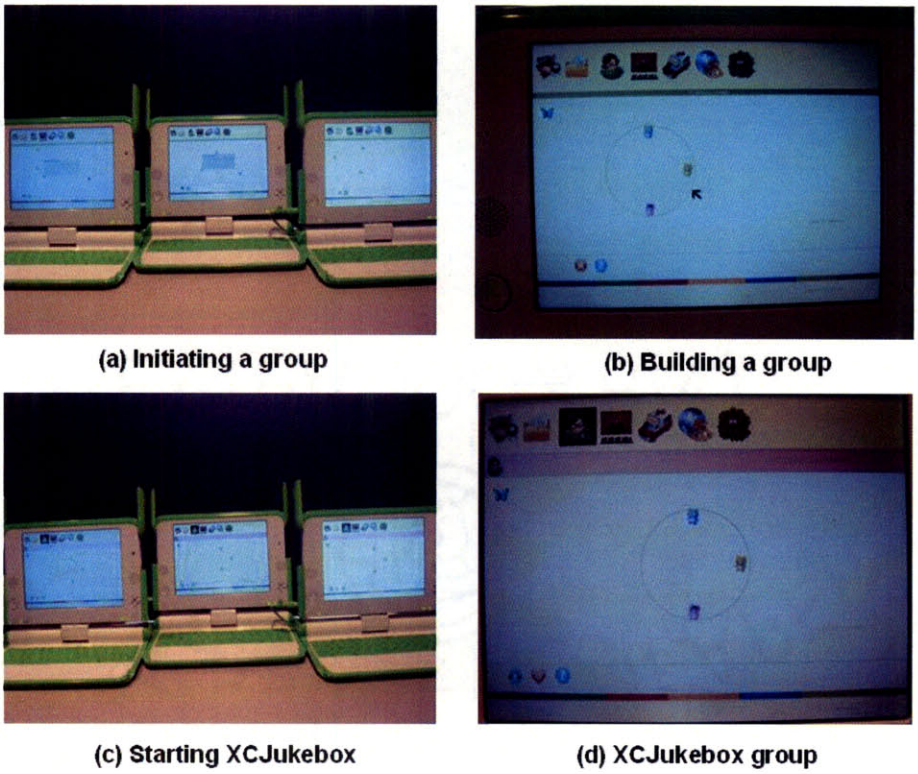


Figure 47. Social Group Management

Now, the users can create a collaborative, participatory application as using the social broadcast queue. Figure 48 shows that each user queues an audio file on the social broadcast queue in the orderly way like a jukebox, and then that the status of queue of each node are synchronized at the same time. In other word, each can be aware readily of *what is going on the broadcast transmission of other nodes* through the synchronization method. The social broadcast queue operates the same way as the Virtual Broadcast Queue as described in Section 4.2.3. Next, the social broadcast queue initiates the broadcast transmission of the stacked music files in the orderly way, as shown in Figure 49. In the interface of XCJukebox, the *"music note"* icon located on the top of the node icon in the interfaces depicts *"who is broadcasting."* That is, the music note icon is re-located according to the order of broadcasting as shown in the figure. The social broadcast queue even makes it possible for all the nodes to simultaneously listen to the song broadcasted from one node. In the end, these implementations actualize the scenario of *Virtual Jukebox* as introduced in Section 3.1.

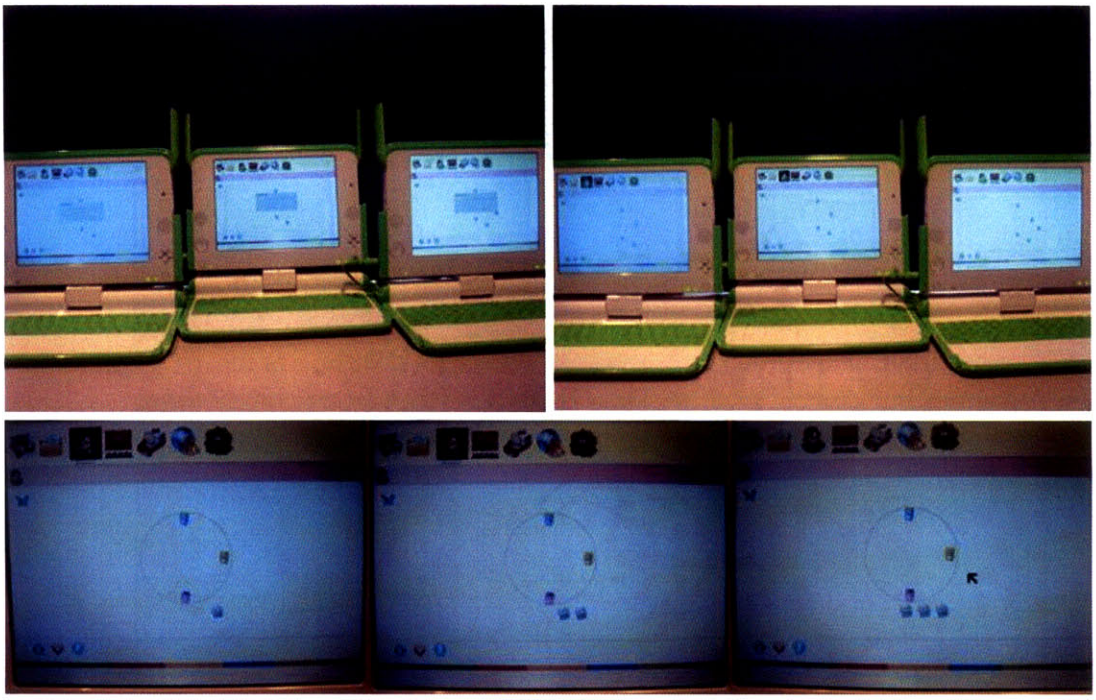


Figure 48. XCast interfaces on OLPC XO

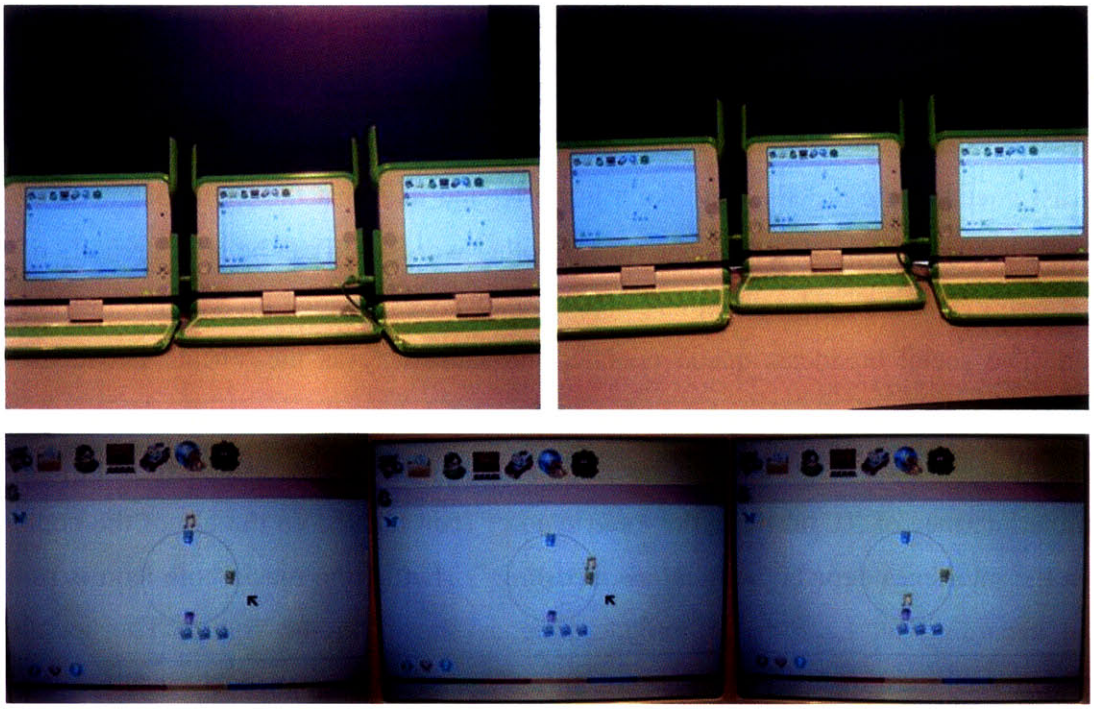


Figure 49. XCast interfaces on OLPC XO

We could find that the value of the implementation is in basically testing applicability of the social networking application in the wireless ad-hoc/mesh network and ultimately transforming an ad-ho/mesh networking into a collaborative, participatory social networking.

5.1.3 Application Management

Last but not least, we need to take a closer look at how the application functions are related to the functional modules of BRSP protocol. In the application layer, the XCast system has an Application Management function. It is comprised of the Neighbor Finder, the Social Group Builder, and the Social Broadcast Queue modules as shown in Figure 50. First, for the Neighbor Finder module, actually it is not embodied in the Application Management functionality. Instead, it is wrought by the Neighbor Finder module of Group Management. Namely, the Application Management schematizes the neighbor finding function (of Group Management) into graphical features. Next, as mentioned above, the Social Group Builder is distinguished from the Group Builder of Group Management, called Broadcast Group Builder: the Social Group Builder creates a social network group according to users' preferences but the Broadcast Group Builder establishes a broadcast group as a prerequisite for the broadcast transmission. Finally, we need to also identify the Social Broadcast Queue and Virtual Broadcast Queue. Both operate in the same methods, "synchronization" and "neighbor knowledge," and organically interact with each other. However, they are different in the target operations: The Social Broadcast Queue manages social resources issues such as Spam and overloading in the application layer but the Virtual Broadcast Queue deals with collisions caused by the blind broadcasting in the networking layer. Also, the broadcasting order in the Social Broadcast Queue may have an effect on deciding the priorities of packet scheduling in the Virtual Broadcast Queue.

For the implementation, all the three functional modules are actualized in the Application Management to support the participatory applications and IVY Networks and to interact with the management functionalities of the BRSP protocol for the transmission. For

instance, the Social Group Builder uses GRTP transmission protocol and TCP to exchange the control messages for building a group. Also, the Social Broadcast Queue provides the Scheduling module of Broadcast Queue Management with the priority information to decide the order of packets scheduling and transfers the update information of the broadcast queue to the Syn/Up module.

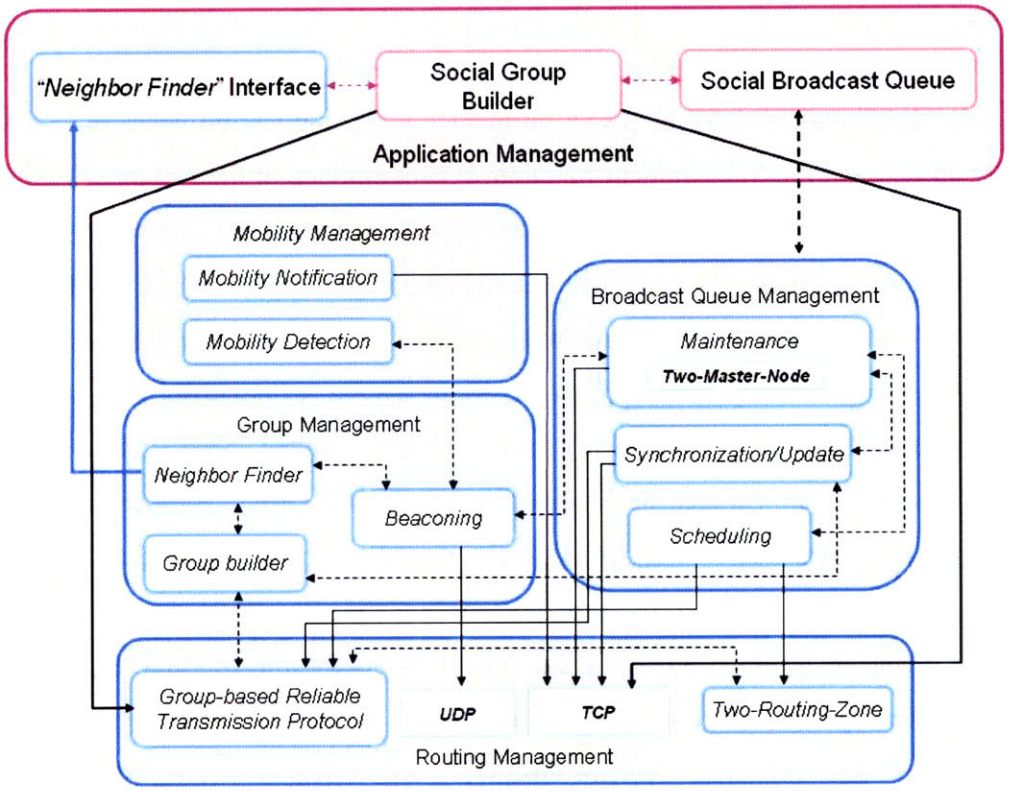


Figure 50. The functional Interaction flows between Application Management and other management.

5.2 Experiments with OLPC XOs

With the OLPC XO implements of the XCast system, we had experiments on the performance of the BRSP protocol before testing user experiences of XCast system and applications. The succeeding sections show the details of experiments and results of the performance tests. The user experience tests by COUHES (Committee on the Use of Humans as Experimental Subjects) leave a future work.

5.2.1 Scenarios

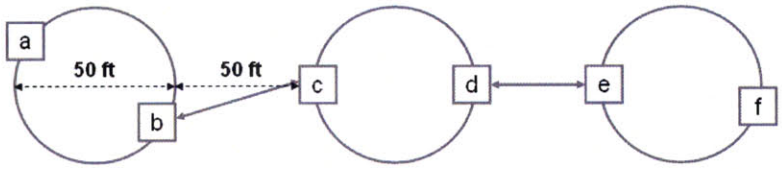
In the experiments with OLPC XOs, we have two scenarios according to routing types: One is a "one-hop routing" scenario where all XOs directly communicate with each other, and the other is a "multi-hop" routing scenario where target destinations exist beyond one hop. For the experiments, we performed outside field tests with six XO laptops. The location we used was a flat field in vicinity of MIT football stadium in Cambridge, MA. The reason we selected the outdoor experiments is to have a clean environment to minimize the amount of external Wi-Fi signal interferences. This is in contrast to the indoor experiments which are likely to often have troubles with massive Wi-Fi signal interferences.

In the one-hop routing scenario, our goal is to scrutinize the effect on the broadcast packet collision by the *"blind broadcasting,"* comparing the existing ad-hoc broadcast transmission with the BRSP-based broadcast transmission. To actualize the scenario, we deployed a simple ad-hoc network in the MIT Briggs Field, as shown in Figure 51. We also located a laptop running a packet sniffing tool (e.g., Wi-Spy) in the center of the testbed. To gather data on the collision rate of broadcast packets, we vary two parameters, 1) the number of nodes and 2) packet sending rate. First, for the variation of number of nodes, we start the experiment with two XOs and afterward gradually increase the number of XO nodes. Next, we raise the broadcast packet sending rate, from 1 packet/second (512 bits/s) to maximum 80 packets/second (40.960 Kbits/s), in the wireless ad-hoc network.

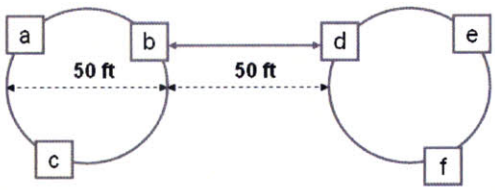
With respect to the multi-hop routing scenario, we exam the packet collision and the redundancy caused by the re-transmission of broadcast packets during the multi-hop routing. For that purpose, we compare the Two-Zone routing protocol of XCast system with Hybrid Wireless Mesh Protocol implemented on OLPC XO. First, we create two topologies to investigate the broadcast packet collision rate and the packet overhead by the re-routing of broadcast packets. The first one is a "three-hop" topology, which is composed by three broadcast groups with two XO members, and the second topology is

formed by two broadcast groups constituting of three ad-hoc nodes, as shown in Figure 52 (a) and (b), respectively. These topologies serve a spectrum to gauge efficiency of the cognitive routing ways of the BRSP protocol under constraint of the number of experimental nodes. The three-hop topology, for instance, allows the simplest routing. That is, it enables the Two-Zone routing protocol to transform broadcast routing into unicast routing among all hops. In the two-hop topology, on the other hand, the broadcast routing still exists in a broadcast group. Next, we analyze the broadcast transmission rate and the packet overhead in each topology as increasing the packet sending rate like the first scenario.

Figure 51. The topology of a simple ad-hoc network consisting of six XOs



(a) Three-hop topology



(b) Two-hop topology

Figure 52. The two topologies of the multi-hop routing scenario

Table 1 shows the experiment parameters we chose for the performance tests of the BRSP protocol in the OLPC XO-based ad-hoc/mesh networks.

Table 1. Experiment Parameters

Experiment Parameter		Value
System platform		OLPC XO Beta-4 (Build: 659, firmware: Q2D14)
Memory	NAND	1 Gigabyte
	RAM	256 Megabytes
CPU		AMD Geode LX 700@0.8W processor at 433MHz
Wireless Adapter		Marvell 8388 SoC Radio
Data packet size		64 bytes
Broadcast Packet Rate		2 Megabits/Second
Running time		240 seconds
Number of Trials		10

5.2.2 Analysis

Most of all, we define the "packet drop ratio" in *the probability of broadcast packets who do not reach to any target destination node in a static wireless network.* We explore how the packet drop ratio is for each protocol according to the conditions given in each scenario to assess the efficiency of each protocol. That is, the parameter results in becoming a criterion to gauge the reliability of the systems. Note that OLPC XOs use simple controlled flooding in the current implementation (e.g., Build 711) to propagate broadcast and multicast frames. That is, each node retransmits every broadcast frame that it receives one time.

As the first experiment, in the one-hop routing topology, to investigate the packet drop ratio as increasing the number of broadcast nodes, we set each node orderly generates broadcast packets with the constant sending rate, 40 packets/second, during the run time.

Figure 53 illustrates how the broadcast packet collision rate is for each protocol. Obviously, the transmission ratio of simple flooding of the legacy 802.11 will suffer in this scenario, as a congested network causes numerous collisions. Utilizing the neighbor-knowledge scheme, however, the BRSP efficiently controls the re-transmission for the broadcast packets. The benefit is especially from the methods to synchronize broadcast queues for local topology information and to restrain the unnecessary re-transmission, as described in Section 4.2.5.

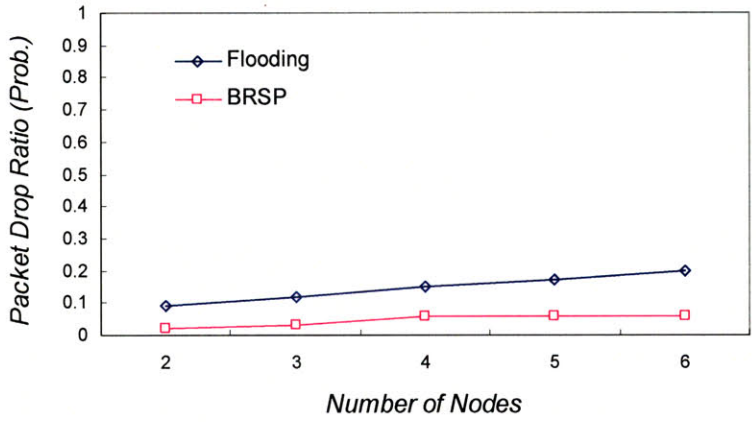


Figure 53. One-hop routing: Packet Drop Ratio vs. the Number of Nodes

Increasing the frequency of packet origination, we again experiment the packet collision rate in the one-hop routing topology consisting of six XO nodes. This experiment helps understand how each protocol reacts under congestion situations. Congestion can be obtained by increasing packet size, increasing packet sending rate, or both. We fix the packet size and vary the packet origination rate (source rate) as shown in Table 1 because we anticipate broadcast packets, as control type packets, to generally be small in size. Figure 54 shows how the packet drop ratio changes according to the augment of the broadcast packet source rate. For the simple flooding, the number of broadcast packets that successfully arrive any target destination node further reduces as the network becomes more congested, which directly illustrates the effect of collisions and queue overflows in congested networks. As shown in the figure, however, the BRSP protocol efficiently schedules broadcast packets to be sent reliably to all the target nodes through

the Virtual Broadcast Queue. In other words, the BRSP protocol produces about more than 93 percent transmission ratio even in the congested environments.

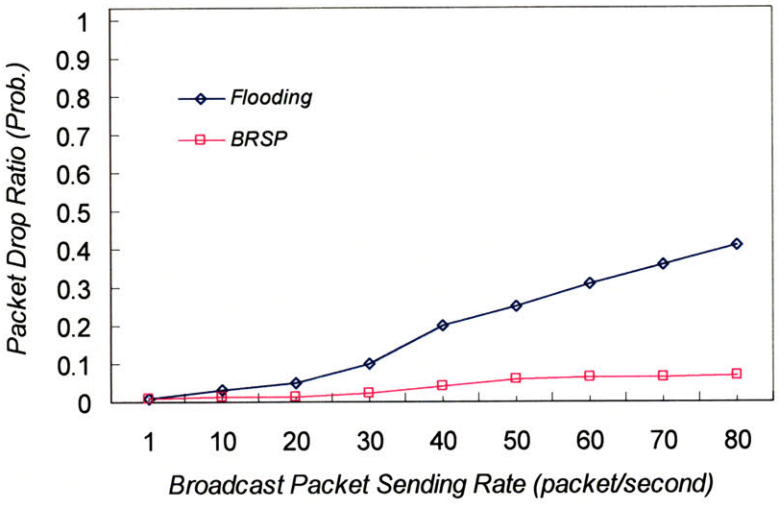


Figure 54. One-hop routing: Packet Drop Ratio vs. Broadcast Packet Sending Rate

For the multi-hop routing scenarios, we perform experiments to see how each protocol responds to the re-transmission of broadcast packets in the two different topologies, *"three-hop"* and *"two-hop"* topologies. In the experiments, we raise the number of packets sent per second, from 1 to 80 (packets/second). First, Figure 55 represents the packet drop ratio caused by the re-transmission of broadcast packets in the multi-hop routing, in four factors based on the topology aspects. As we can guess, the simple flooding frequently reproduces packet collisions or drops in both topologies as the sending rate increases. It results in failing to transfer broadcast packets to the target nodes. The BRSP protocol, on the contrary, keeps the packet drop ratio to be very low without any abrupt jumping point. It is from the cognitive algorithm that its routing protocol transforms the broadcast transmissions into unicast or multicast operations according to network topologies or target receiving nodes. For instance, in the three-hop topology, the BRSP protocol changes the broadcast routing totally into unicast routing every hop as shown in Figure 56. Actually, there exist no packet collisions or drops by the re-transmission of broadcast packets. We expect that the packet drop ratio shown in Figure 55 is mainly caused by the unreliability of wireless medium. Also, in the two-hop topology, as shown in Figure 57, the broadcast operations are transformed into multicast

within a broadcast group and unicast transmission between the arbiters (broadcast relay nodes), respectively. Unambiguously, this leads to the least number of packet collisions or drops.

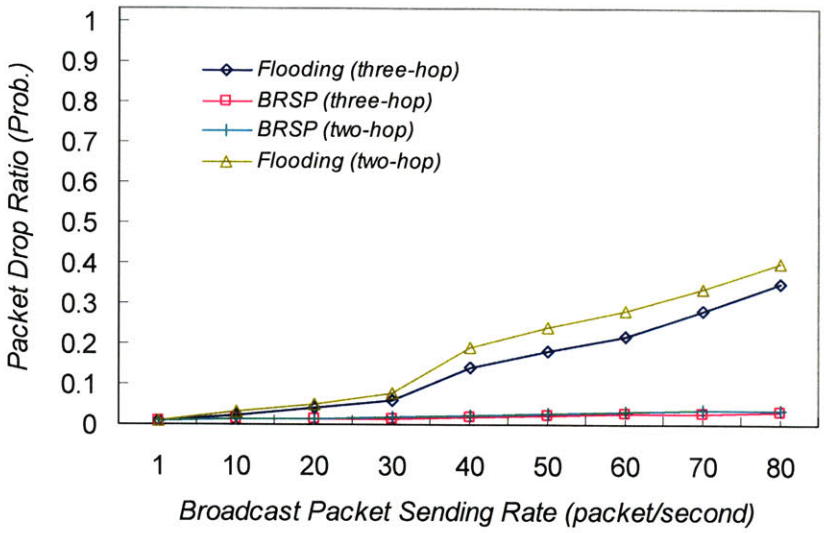


Figure 55. Multi-hop routing: Packet Drop Ratio vs. Broadcast Packet Sending Rate

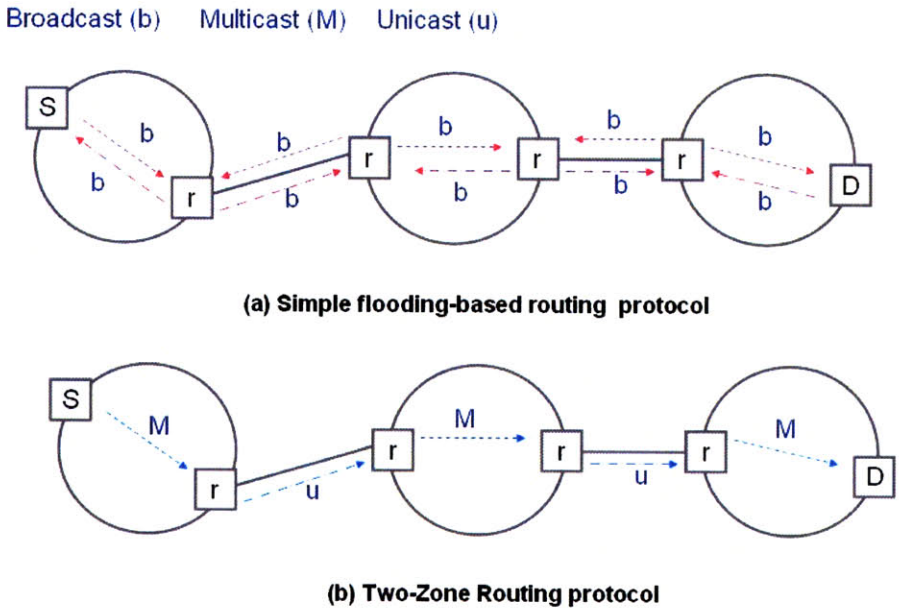


Figure 56. Three-hop topology

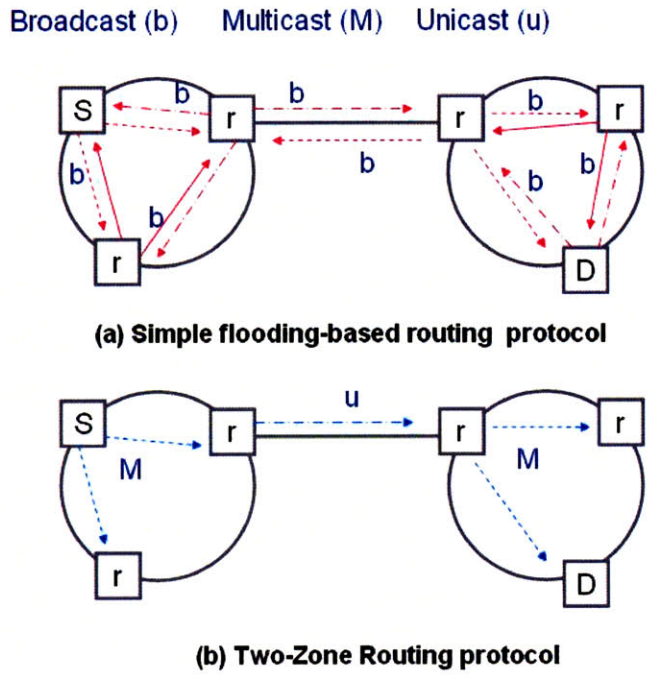


Figure 57. Two-hop topology

Next, we delve the redundancy resulted by the re-transmission of broadcast packets in the multi-hop topologies. In the experiment, we define the "total broadcast overhead" as *how many re-transmissions per broadcast packet are performed.* That is, we measure the number of all (control or payload) packets unnecessarily re-broadcast from each node during the multi-hop routing. Figure 58 shows the overhead of broadcast packets by the re-transmission as the packet sending rate increases. Needless to say, the simple flooding suffers totally from the broadcast overhead: The augment of such the blind broadcasting by the re-broadcasting leads to serious redundancy, contention, and collision to have the performance of wireless ad-hoc/mesh networks harshly degraded. This results in the broadcast jumble as discussed in Section 1.1.3. Contrary to the simple flooding, the Two-Zone routing protocol of BRSP performs very judiciously re-routing of broadcast packets via the neighbor-knowledge method. Extremely, the Two-Zone routing protocol leads to just simple unicast routing for the broadcast packets in the three-hop topology, while the simple flooding continually bring about broadcasting routing, unwittingly, as shown in Figure 56: The multicast transmission actually is transmuted to a unicast operation in a broadcast group because there exist only two broadcast nodes. Also, in the

two-hop topology, the BRSP routing protocol has the relay nodes quite prohibitive to re-transmit the broadcast packets. As shown in Figure 57 (b), after receiving multicast packets from the source node S, the relay nodes Rs decide whether to re-transmit the packets toward a target node. If the relay node is an arbiter node which is in charge of re-routing between the broadcast groups, it ferrets out other arbiter node in a different group and then re-transmits the received packets to the counterpart. This efficiently prevents any unintended transmission from being performed and in the long run, reduces the baleful effects by the overhead of broadcast packets and even the packet collisions.

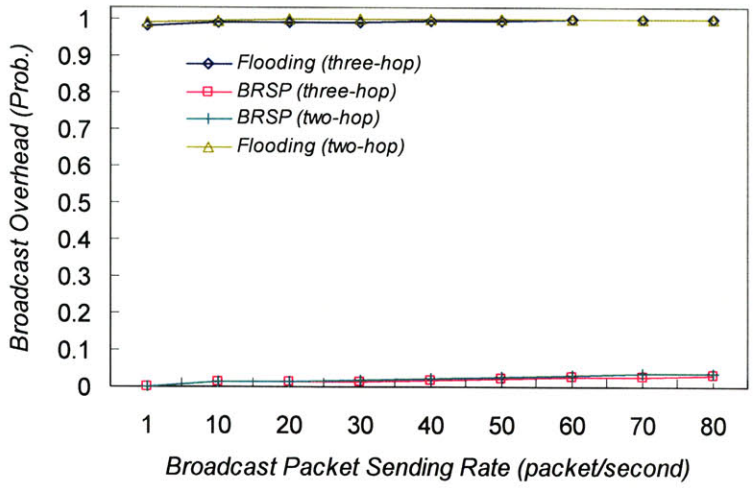


Figure 58. Multi-hop routing: Broadcast Overhead vs. Broadcast Packet Sending Rate

5.2.3 Lessons learned

First, what we learned from the outdoor field experiments is as follows.

- The most difficulty in the outside field tests is in providing the experimental XOs and the network packer analyzer with enough power, timely. Actually, we did not find any proper power supplier in the Briggs Field. This made the experiments frequently discontinued.
- We realized that there are too many sources generating Wi-Fi radio signals in our surroundings from offices to playgrounds, as well including machines with 2.4 GHz band signals such as microwave ovens, cordless phones, and so on.

Therefore, it is very tricky to ferret out a clean island where no Wi-Fi radio signal exists.

- To efficiently manage the XCast system running on XO, we created a system initiator which triggers the XCast applications and protocols' operations via sending some simple control messages. This made us spend less time on performing the experiments and also control the remotely located XOs.

We performed experiments on the packet collision and overhead resulted by the blind broadcasting and broadcast jumble scenarios, comparing the legacy 802.11 broadcast protocol installed on the OLPC XO with the BRSP protocol. The results of the experiments showed that our proposed protocol outperformed the existing 802.11 broadcast protocol in both the packet drop ratio and the overhead facets. However, we just tested them in a simplified network consisting of just six nodes. One might argue that those tests would not acceptable to verify the applicability of the proposed protocol in a large scale network. Thus, we build a simulated testbed and explore the suitability of our proposed protocol in the large-scale wireless/mobile ad-hoc network. The followings show what we focus on in the simulations. Section 5.3 provides the more details.

- The experiments were not enough to prove Scalability and reliability of the XCast system due to the small number of wireless ad-hoc nodes. We leave the large scale experiments with XOs a future work. Instead, in the timeline, we explore the effect on scalability and reliability of our propose system through some simulations.
- In the testbed, we did not perform any experiments for mobility support because of the outstanding issue on deciding when to send NTL message in the "NTL/CTL mobility pair." We substitute it with some simulations on a comparative protocol with mobility support (e.g., AHBP-EX), namely we could estimate usability and applicability of the BRSP mobility support in the mobile ad-hoc networks via the reference.
- Also, the comparison to the existing broadcast support protocols as described in Section 2.2 was not performed in the testbed. Actually, it requires too much time

and effort to implement all the prototypes of comparative protocols, and thus would not be affordable. Alternatively, we make the best use of simulation methods to examine in order to note the similarities and differences between the comparative protocols and the BRSP protocol.

5.3 Simulations

In the section, we describe some studies and simulations performed to assess the system. The reason to select the simulation as an approach for the evaluations is to compare the BRSP protocol with the related protocols in the given certain parameters that have effects on the performance of the system. This helps understand the efficacy of the system we propose and its applicability to the real networks.

5.3.1 Comparative protocols

For this simulation, we select four comparative protocols such as the Simple Flooding, the Counter-based scheme, Scalable Broadcast Algorithm (SBA) and Ad Hoc Broadcast Protocol (AHBP). The reason to choose those protocols is that our proposed scheme, BRSP is an embodiment constituent of representative functionalities of the selected protocols. That is, the analysis of performance for each protocol results in helping glean an overall understanding of that for the BRSP protocol. This section gives the more details.

First, for the Simple Flooding, only one protocol has existed in the category. It has widely been used as the most basic comparative protocol because of generally working as the lowest bound for performance. This makes our choice clear.

Next, in the category of Probability-Based methods, there exist two schemes proposed in [4]: the Probabilistic scheme and Counter-based scheme. The paper shows that the Counter-based scheme outperformed the Probabilistic scheme in most of the simulated

conditions. The BRSP uses partially the Counter-based scheme to decide whether to re-broadcast in the multi-hop routing as described in Chapter 4. Therefore, the analysis of performance of the Counter-based protocol helps understand how effect it has on that of the proposed system.

Many protocols utilize intelligence of "Neighbor Knowledge" as described in Section 2.2.5. The neighbor knowledge protocols can be classified by whether a node makes a local decision to retransmit a broadcast packet. For example, the protocols such as Flooding with Self Pruning, SBA, or LENWB makes use of this local topology knowledge for the re-broadcasts. However, Dominant Pruning, Multipoint Relaying, AHBP, and CDS-Based Broadcasting protocol allow preceding nodes to involve the decision at the succeeding nodes to re-transmit broadcast packets told, either via the packet or a previously sent control packet. We choose two protocols to represent each type of the neighbor knowledge methods: SBA and AHBP.

The SBA could be a good comparison to the BRSP to be aware of how effect the local decision method has on the performance. The following descriptions justify our choice. Authors in [33] shows that Flooding with Self Pruning is efficient in some network conditions even if it is inapplicable in others. Similarly, LENWB performs poorly over a range of network conditions common to mobile ad-hoc networks [39]. SBA, however, applies the two-hop neighbor knowledge efficiently and is shown to obtain good performance from simulation results in [32]. In the BRSP protocol, the local decision is made by the synchronization of broadcast queues within a one-hop radius, called virtual ad-hoc cell.

We make a choice of AHBP to represent neighbor knowledge protocols that do not use a local decision on whether to re-broadcast, even if there exist other protocols utilizing Connected Dominating Set-based routing, to efficiently approximate a minimum CDS in a wireless ad-hoc network [40] [41]. Basically, the AHBP outperforms Dominant Pruning and Multipoint Relaying in the ways to select broadcast relay nodes, as described in Section 2.2.5.5. The main reason for the selection is the mobility support of AHBP

protocol, called the AHBP mobility extension (AHBP-EX) designed in [37]. Similar to AHBP, our proposed protocol, BRSP also has at least one "broadcast relay node" per a one-hop radius (e.g., virtual ad-hoc cell), in order to decide whether to re-broadcast the received broadcast packets to other relay nodes or satellite nodes. As introduced in Section 4.2.7, however, the BRSP does not yet have full functionalities dedicated to hosts' high mobility. Therefore, our preference to AHBP and its mobility extension would be reasonable for the estimation of applicability to the mobility extension in BRSP as well as the performance comparisons.

5.3.2 Scenarios and implementations

With the broadcast protocols we selected in the previous section, we perform side by side comparison tests of the BRSP protocol. Specifically we have four strategies for the tests: One is to compare BRSP with the selected protocols over a range of network conditions including node densities, node mobility and packet sending rates. The second strategy is to pinpoint areas where each protocol performs well and then areas where the BRSP could be improved on the basis of the analysis of the comparative protocols. Third, we analyze the performance of BRSP in the basis of the weak and strong points of the comparison protocols. Finally, we check if the BRSP stands out in its current form as being appropriate for dynamic wireless ad-hoc network conditions. That protocol may serve as a model fit for standardization or as a benchmark for further work.

To achieve these strategies, we focus on four scenarios such as (1) protocol efficacy in reliability and scalability facets, (2) congestion, (3) mobility, and (4) combination. Each is outlined in the succeeding sections. We implemented a simulation code based on the code developed in [42] to test performances of the protocols selected for comparisons in the scenarios. While experiments for the four scenarios vary some network parameters, the ones outlined in Table 2 remain constant for all simulations. The network area and the node transmission distances were chosen to allow reasonably timed simulation of networks characterized by high node density, congestion level, and node mobility. Each point on each graph presented in this section is the average of 10 simulation runs. We

include error bars which indicate 95 percent confidence that the actual mean is within the range of said interval. In certain cases, the confidence intervals are small enough that they are obscured by the symbol itself.

Table 2. Simulation Parameters

Simulation Parameter	Value
Simulator	NS-2 (1B7a)
Network area	300 x 300 m
Radio signal coverage	100 m
Data packet size	64 bytes
Node Max. IFQ length	50
Simulation time	200 seconds
Number of Trials	10
Confidence Interval	95%

In addition to the simulation parameters, protocol parameters must be specified. Protocol parameters for all simulations, unless otherwise noted, are shown in Table 3. First, the Counter-based value is chosen to minimize the number of re-transmitting nodes with the constraint of maintaining at least 95 percent transmission ratio in a 30 node network (the effects of threshold values are illustrated in the first scenarios, "protocol efficacy"). Next, the time delay for the clear channel assessment, called "Channel Assessment Delay (CAD)," affects all protocols except Flooding, AHBP, and BRSP. The CAD maximum value is important for studies that make use of an 802.11 MAC, such as the *"congestion"* and the *"combination"* scenarios. The reason is that the adjustment of the value could help alleviate the redundancies or collisions caused by re-broadcasting. Finally, a beaconing period, for SBA and AHBP, is needed for scenarios where hosts are allowed to be mobile, such as the *"mobility"* and the *"combination"* scenarios. In scenarios with a static wireless network, beacon signal costs are ignored. Table 4 also sums up network parameters pertaining to each simulated scenario.

Table 3. Protocol parameters

Protocol Parameter	Value
Count-based Threshold	4
CAD Time max	0.1 (second)
Beacon interval	1 to 5 (signal/second)

Table 4. Scenarios and variation parameters

Simulation scenarios	Network parameters
Protocol Efficacy	Null MAC, a static wireless network
Congestion	802.11 MAC, a static wireless network
Mobility	Null MAC, adaptation of beacon periods
Combination	802.11 MAC, adaptation of beacon periods

5.3.2 Protocol Efficiency: Reliability, Scalability

The purpose of the first scenario is to evaluate the degree of reliability and scalability of the BRSP by comparing its performance with the selected protocols in a static wireless network. Each algorithm has a different level of intelligence; the intelligence has effect on how reliable and scalable the protocols are in a given network.

In general, the broadcast protocols are highly dependent on the density of the network. In sparse networks, the protocols are expected to perform similar to Flooding, because each node may have to rebroadcast to reach isolated neighbors. One would expect that as density increases, proportionally fewer nodes should rebroadcast. In this case, we need to have a criterion to gauge the relative performance of the protocols. We consider two bounds: One is the *"worst-case"* bound provided by Simple Flooding, as we could readily guess, and the other is a theoretical *"best-case"* bound provided by the Minimum Connected Dominating Set (MCDS) [43]. Note that a*n MCDS is the smallest set of re-broadcasting nodes such that the set of nodes are connected and all non-set nodes are within one-hop of at least one member of the MCDS.* It is impossible for any protocol to

perform better than the MCDS and unlikely to perform worse than Simple Flooding. Thus, these two bounds provide a useful spectrum to evaluate *"reliability"* and *"scalability"* of the protocols.

As known, the computation of the MCDS over a given network is in general an *"NP-complete"* problem, so approximations must be employed in practice. A number of papers have proposed approximation algorithms to determine MCDS (e.g., [40] [41] [43]). However, for the exact MCDS, we adapt an approach to use a Brute Force and Ignorance (BFI) [44]. Given a static topology, BFI iterates through every possible node combination for a proposed MCDS size to determine if indeed there is an MCDS of that size. The simulation model of [42] we refer to in the simulation uses the MCDS approach with BFI.

Table 5. Average number of neighbors according to numbers of networks nodes

No. of Network nodes	Avg. No. of neighbors
20	4
30	6
40	8
50	10
60	12
70	14
80	16
90	18
100	22

For this study, we varied the number of nodes randomly placed in the network area from 20 to 100. Table 5 shows the average number of neighbors (i.e., the actual neighbor count for each node averaged over 10 runs). Our average number of neighbors covers the gamut of sparse to dense networks.

By using a static wireless network and Null MAC, we avoid any effects that mobility and congestion may have on the protocols. In other words these conditions give a good quasi-theoretical view of the core algorithms of the protocols. In addition these conditions also approximate networks which may be characterized by high stability and very low traffic rates. A broadcast packet origination rate of 10 packets per second was used, although the use of a Null MAC renders the origination rate irrelevant.

5.3.2.1 Analysis

To assess the reliability of each protocol, we had a simulated test to see how the transmission ratio is for each protocol as the number of nodes is increased in a static wireless network. In this case, we define the transmission ration as *the probability of network nodes who receive any given broadcast packet.* Figure 59 shows that all protocols are highly reliable in dense networks; in sparse networks, most protocols except for Counter-based scheme, such as Simple Flooding, SBA, AHBP, and BRSP are the most reliable.

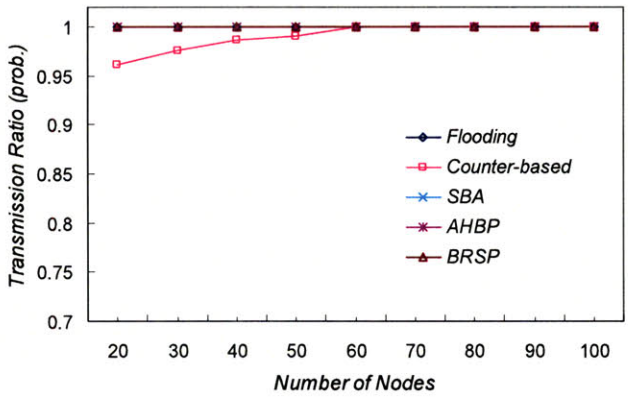


Figure 59. Transmission ratio versus Number of nodes

We measure the degree of scalability as testing *how many nodes are required to re-broadcast the packets they received as the node density increases.* Figure 60 shows the number of re-broadcasting nodes required by each protocol. The figure illustrates that

each broadcast scheme except Flooding is scalable in terms of higher node density in a fixed wireless network area. As expected, the protocols utilizing more intelligent algorithms require the least number of rebroadcasts. In the graph, the Neighbor-knowledge schemes require fewer rebroadcasts than the Counter-based schemes. In other words, neighbor knowledge schemes benefit from the "beacon" signaling and high algorithmic costs by having fewer nodes retransmit each broadcast packet. Note that utilizing the neighbor-knowledge in a one-hop radius, the BRSP requires the least number of rebroadcast nodes. The benefit is especially from the methods to synchronize broadcast queues for local topology information and to re-transmit broadcast packets in the basis of *"unicast"* in multi-hop routing, as described in Section 4.2.5.

In sparse networks, on the other hand, all schemes converge to Flooding because each node tries to retransmit each packet. In dense networks, for example of the simulations with 100 nodes, BRSP requires 17 percent of the nodes to rebroadcast versus the theoretical best-case, MCDS which requires 8 percent of the network nodes to rebroadcast. While there is room for improvement, BRSP approximates the minimum fairly well. We note that the MCDS size is relatively constant over the range of node densities in Figure 60, which intuitively makes sense.

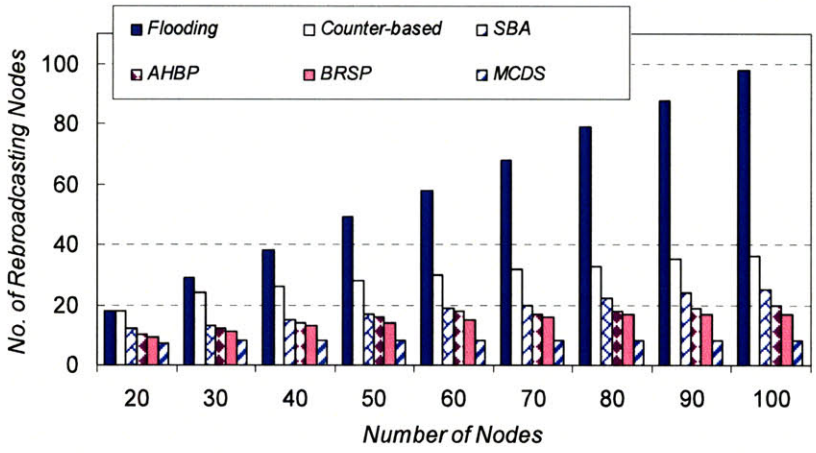


Figure 60. Number of Retransmitting Nodes versus Number of Nodes

As mentioned, the BRSP could adapt the algorithm of the Counter-based scheme to decide whether to re-broadcast in the multi-hop routing. It is worthwhile to take a closer look at how to enhance the Counter-based scheme. In Figures 59 and 60, the threshold value for the Counter is held constant. The values, however, is not optimal over the range of node densities simulated. Varying the threshold values, we re-test the performance of the scheme in two components, transmission ratios and the number of re-broadcast nodes.

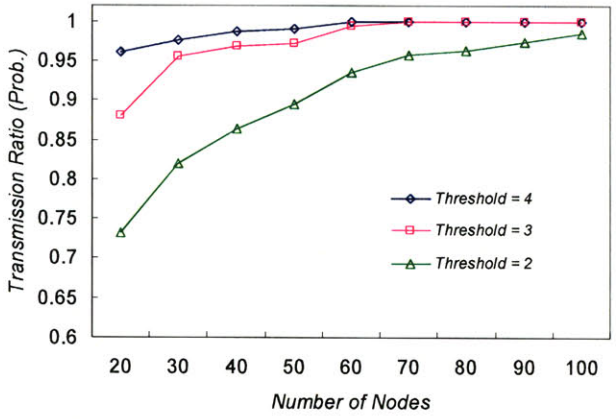


Figure 61. Counter-based scheme: transmission ratio to threshold value

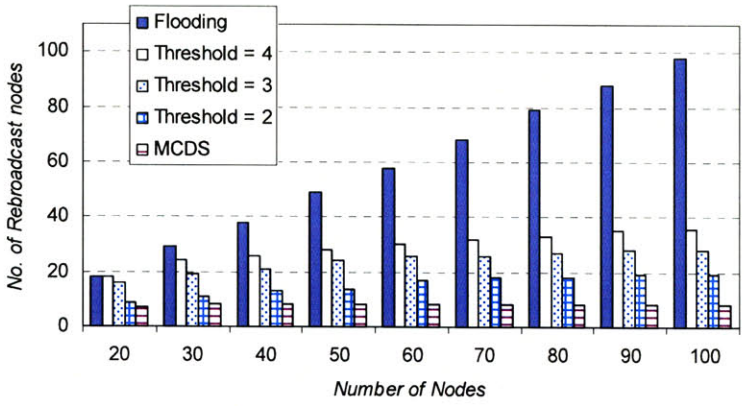


Figure 62. Counter-based scheme: the number of re-broadcast nodes to threshold value

First, Figure 61 shows the transmission ratios for the Counter-based scheme given three threshold values. To maintain a high transmission ratio in sparse networks, a higher threshold is needed; to maintain a high transmission ratio in dense networks, a lower

threshold can be used. Next, Figure 62 illustrates that a lower threshold value means fewer nodes retransmit. Thus, we conclude that extending the Counter-based scheme to adapt its threshold value based on the network density would improve the overall performance of the protocol. We also deduce that in the BRSP protocol, the adaptation to the counting threshold value suit for the network density would result in improvements of the performance. This remains for a future study.

5.3.3 Congestion

The purpose of this scenario is to quantify the effect of congestion on the BRSP protocol. In the congested scenario, the blind broadcast often happens and immediately turns into the broadcast jumble. We could estimate a spectrum to gauge performance of the BRSP reacting to congestion. First, unambiguously the transmission ratio of Simple Flooding would suffer in this scenario, as a congested network causes numerous collisions and/or queue overflows. The other protocols adopting more intelligent algorithms would be less sensitive to congestion.

Table 6. Simulation requirements and parameters for the congested networks

Parameter	Value
MAC	802.11 MAC
Range of packet sending rate	1 to 80 (packets/second)
Packet payload	64 bytes
Traffic pattern	Random generation
Number of nodes	60 (nodes)
Network model	A static wireless network

Table 6 shows simulation and network parameters used in the scenario. This implementation is performed using the contention-based 802.11 MAC scheme. Congestion can be obtained by increasing packet size, increasing frequency of packet origination or both. We fix the packet size and vary the packet origination rate (source rate) as shown in Table 6 because we anticipate broadcast packets, as control type

packets, to generally be small in size. In this scenario, to create a random traffic pattern, each new packet was assigned a source node randomly chosen from the entire pool of network nodes. Our simulation has the constant number of network node, even if we could expect significantly different results using different numbers of network nodes. However, the goal of this scenario is to obtain a general trend for the effect of a congested network. A static wireless network was used to ensure that the effects of mobility would not interfere with the effects of congestion. To avoid any anomalies in one static network topology, we average the results over 10 unique topologies for each packet sending rate.

5.3.3.1 Analysis

In this scenario, we test the BRSP and the comparative protocols under the constraint of limited bandwidth of the IEEE 802.11 networks (e.g., 2 Mbits/s for broadcasts). Thus, all figures presented in this section shows congested situations as the number of broadcast packets generated per second along the x-axis increase.

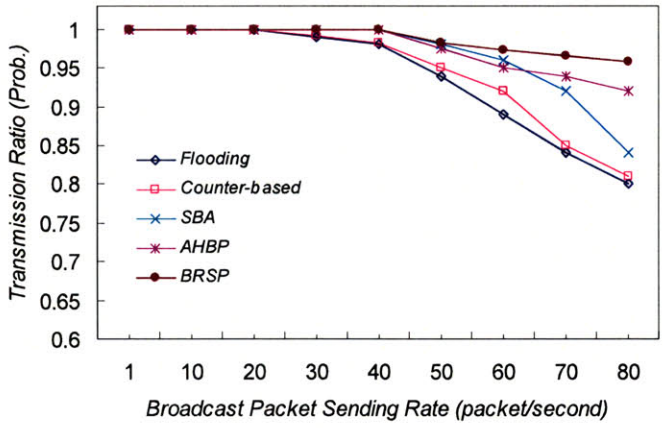


Figure 63. Transmission probability versus Packet sending rate

Figure 63 illustrates that each protocol suffers from the delivery of broadcast packets as the network becomes more congested. Comparing this graph to Figure 60 shows the relationship between performance in congested networks and the number of redundant re-

broadcasts: Figure 63 shows the neighbor-knowledge methods outperform Counter-based methods. In other words, we could estimate that the protocols that utilize intelligence to minimize the number of redundant re-broadcasts deliver the most packets in congested networks. The transmission ratio in most schemes has a breaking point at packet sending rate. The reason is that most protocols except for BRSP use UDP for the broadcast transmission, and it makes the protocols effectively not react to the congestion. In the test, BRSP is the best performer with respect to transmission ratio; in fact, it is the only protocol with over 95 percent transmission ratio for all congestion levels. Since making use of the TCP-like transmission method for the group-based multicasting as well as the intelligence to lessen the redundancy via the broadcast queues, the BRSP becomes more robust in the congested environment.

As the network becomes more congested, variation of the number of re-broadcast nodes is shown in Figure 64. Since the number of network nodes and the simulation area remain constant, one might expect the number of retransmitting nodes to remain constant as well. Indeed, BRSP and AHBP seem to basically follow this trend. As expected, BRSP constraints the number of re-broadcast nodes regardless of the increase of congestion in the network, thanks to the algorithm to react to the crowding. For the Simple Flooding, however, the number of re-broadcast nodes drops as the network becomes congested. This directly illustrates the effect of collisions and queue overflows in congested networks. On the other hand, the Counter-based and SBA protocols incur the increased number of re-transmissions (per broadcast packet) as the number of broadcast packets originated increase. This increase in re-transmissions is due to the time delay for the clear channel assessment utilized by each of these two protocols. In the Null MAC case of the first scenario, the CAD had little consequence because delivery of packets was instantaneous. In this scenario, the delivery of packets is subject to the delays due to the transmission times, back-offs from failed clear channel assessments, and blocked the interface queues (IFQs). Essentially, higher congestion prohibits redundant packets to be delivered during the CAD; therefore, more nodes try to re-transmit the broadcast packets they received. This also makes the network more congested. It results in an avalanche effect (e.g., the broadcast jumble).

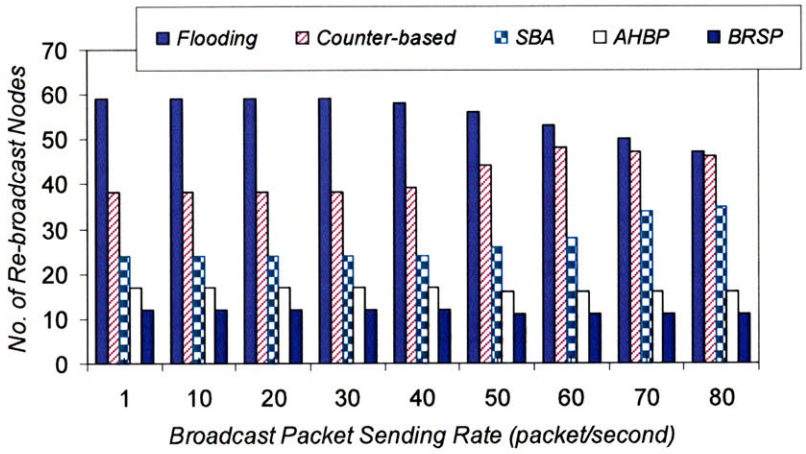


Figure 64. Number of re-broadcast nodes versus Packet sending rate

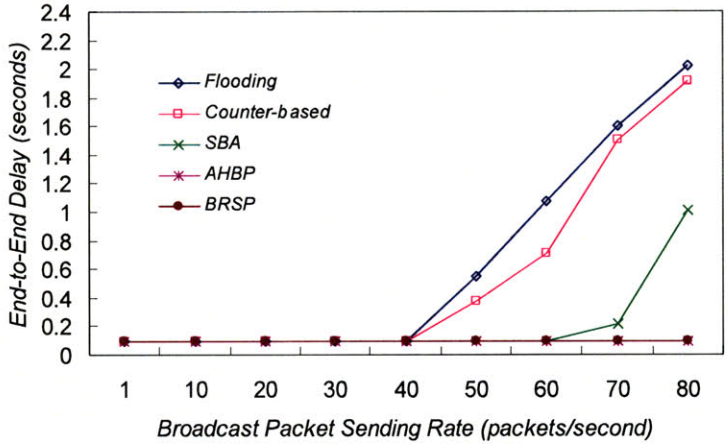


Figure 65. End-to-end delay versus Packet sending rate

Figure 65 verifies that there is a strong correlation between the end-to-end delay and the origination rate of packets in the network. In this case, we define the end-to-end delay as *the time it takes for a destination node to receive a packet generated from a source*. The correlation can be explained by the general congestion effect. Most protocols suffer from keeping the low end-to-end delay. The BRSP, on the other hand, avoids the increase of the end-to-end delay due to congestion control functions of the TCP-like transmission mechanism, the group-based reliable transmission for broadcasts.

Recall that the delay time for channel assessment, CAD is randomly chosen between zero and maximum seconds. We could consider the followings to enhance the performance of the Counter-based scheme and SBA: First, when a network is not congested, low CAD values are desired. Second, when a network is congested, high CAD values are desired. Adapting CAD in this manner maximizes transmission ratio and minimizes end to end delay. To maintain high delivery rates and low end to end delay, we may be able to consider that CAD is adapted to the state of network congestion. However, the adaptation requires a higher number of re-broadcast nodes. In addition, because SBA naturally adapts to dynamic topologies by increasing the number of re-broadcast nodes, the CAD adaptive version of the protocol is unable to cope with a severe network environment like the broadcast jumble network. Therefore, it would not be worthwhile to consider the use of CAD in the BRSP protocol.

5.3.4 Mobility

The third scenario explores the ability of the BRSP protocol to react effectively to host mobility in the network. In this "mobile networks" scenario, we use the requirements and parameters as shown Table 7. Based on the network model of NS2, we also use the random waypoint mobility model [45], which is characterized by *dwelling time*. Each node begins the simulation by remaining stationary for dwelling time seconds. The existing work shows that dwelling times over 20 seconds add significant stability to dynamic networks [46], [47]. Since we prefer to test the protocols without this added stability, we decide to use "one second" dwelling time.

The range of average speeds is distributed uniformly between zero and the maximum speed of 20 meters per second in this simulation. Nodes select a random location in the simulation area and move to that location at a given mean speed. Upon reaching the destination, the nodes pause again for dwelling time seconds, select another destination, and proceed there, repeating this behavior for the duration of the simulation.

Table 7. Simulation requirements and parameters for mobile networks

Parameter	Value
MAC	Null MAC
Broadcast packet sending rate	10 (packets/second)
Number of nodes	60 (nodes)
Mobility model	Random Waypoint model
Dwelling time	0 (second)
Range of average speeds	1 to 20 (meters/second)

The simulations are unrealistic, especially *"at the 20 meter per second"* scenario. It is unlikely to have nodes moving at close to 50 miles per hour in this pattern in a 300x300 meter space. However, the mobility speed can be considered as a proxy for link change rates in general. For example, links may break from nodes moving behind obstructions or vice versa, from interference from other utilizations of the public bandwidth, or even from node devices being turned off. Although we have no supportive data to conclude that these mobile scenarios match real world networks in terms of link change rates, the range of node speeds does provide a diverse range of network conditions.

5.3.4.1 Analysis

In generally, it is expected that our proposed scheme, BRSP would suffer from high mobility. The reason is that frequent host mobility produces outlander nodes that are not exclusionary from a broadcast group and then the BRSP does not deliver the nodes the broadcast packets. We propose improvements on the performance of BRSP in mobile networks through the analysis of the performance of the comparison protocols.

First, the BRSP suffers from a changing topology, as shown in Figure 66, even though showing the performance slightly better than AHPB and AHBP-EX because of its use of local neighbor knowledge. The mobility extension for AHBP (AHBP-EX) marks an improvement over the AHBP; however, it still underperforms the other protocols. As discussed in Section 2.2.5.5, The AHBP-EX requires a node, which receives a packet

from a neighbor not currently listed as an one-hop neighbor, to act as a Broadcast Relay Gateway (BRG). In other words, AHBP-EX handles the case when a neighbor moves inside a node's transmission range between beaconing intervals but does not copy with the situation when a chosen BRG is no longer within the node's transmission range. In results, the outdated two-hop neighbor knowledge corrupts the determination of next-hop re-broadcasting nodes.

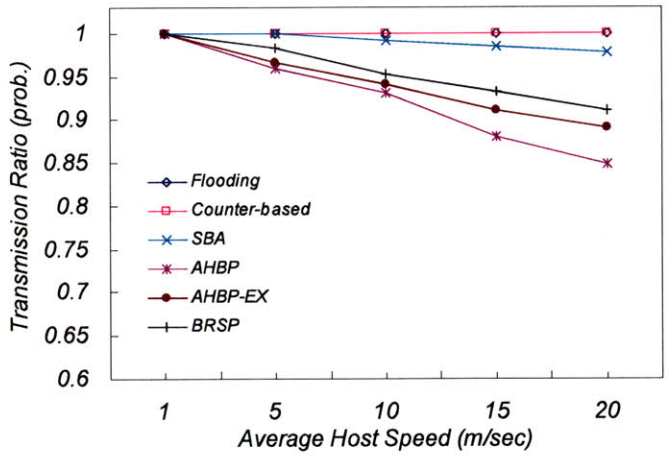


Figure 66. Transmission ratio versus Average host speed

On the other hand, SBA utilizing the local neighbor knowledge is less sensitive to changing topologies than AHBP because of requiring each node to assess its own topology. Recall that if a SBA node receives a broadcast packet from a new neighbor, it is unlikely to be aware of any common one- or two-hop neighbors previously reached; thus the node is more likely to re-transmit the broadcast packets. As shown in Figure 67, SBA naturally adapts to mobility by requiring more nodes to re-broadcast. AHBP and AHBP-EX, on the other hand, have fewer nodes to re-transmit broadcast packets as the node speed increases.

The BRSP also makes use of local neighbor information, but the frequent host mobility prevents the broadcast queues of BRSP from perfectly being synchronized within a virtual ad-hoc cell. That is, in the scenarios with high mobility, there exist many outlander nodes that are not affiliated with a broadcast group. This makes broadcast

packets fail to reach the outlanders. Practically, the outlanders may not have, however, any meaningful effect on the XCast system because they are considered as nodes nonchalant to broadcasts when leaving from the broadcast group. In the case, we need to distinguish the nonchalant from the high mobile.

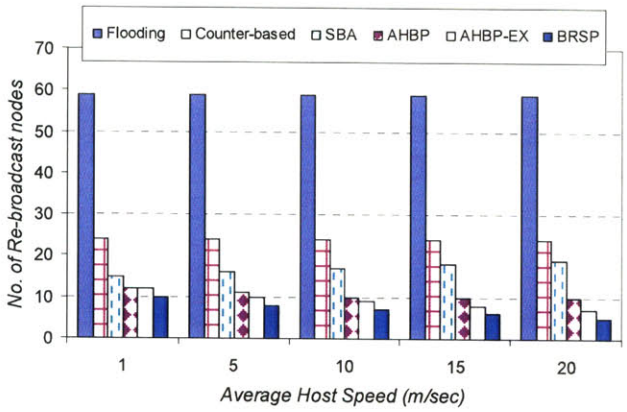


Figure 67. Number of re-broadcast nodes versus Average host speed

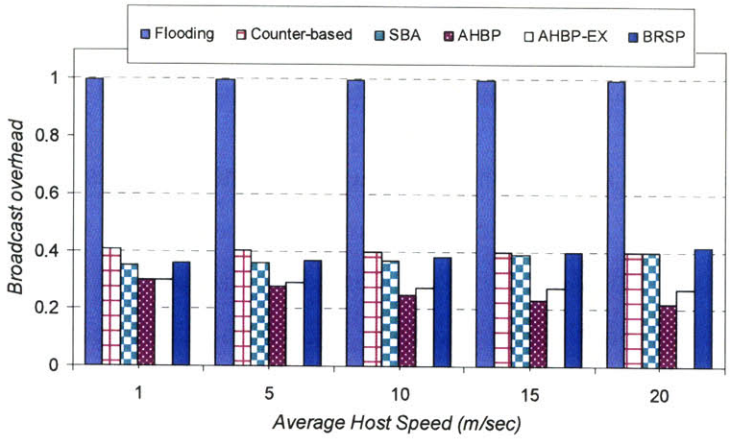


Figure 68. Broadcast overhead versus Average host speed

We evaluate the total broadcast transmission overhead of BRSP and compare it both to the protocols that require periodic beaconing and to the protocols that do not. Note that unlike the experiment with OLPC XOs, in this test, we re-define the "total broadcast overhead" as *how many control packets per broadcast packet are transferred.* Figure 68 illustrates that *"at the one second beaconing period,"* AHBP remains the least

transmission expensive while SBA becomes as even expensive as the Counter-based scheme at high mobility. The BRSP suffers slightly further than SBA. With respect to the overhead of BRSP, as discussed in Section 4.2.3, additionally it uses "Queue update" messages to synchronize the broadcast queues and to maintain the virtual broadcast queue. Since the messages are broadcast, the BRSP results in the added overhead even if the messages are just exchanged on demand, without having a period. This makes the performance of the BRSP degraded more than that of other neighbor knowledge schemes.

From the results above, we can be aware of that the BRSP should use control packets (e.g., beacon signals) judiciously; otherwise the cost of control packets outweighs the benefits of having fewer re-broadcasting nodes. As an example, one viable adaptation technique to correct for high mobility in AHBP-EX is to use shorter intervals between beacons, "Hello" packets. Figure 69 demonstrates that shortening the beaconing interval from 5 second to 0.2 seconds allows AHBP-EX to maintain a high transmission ratio with a changing topology. However, it leads to significant overhead in the broadcast packet transmission as demonstrated in Figure 70.

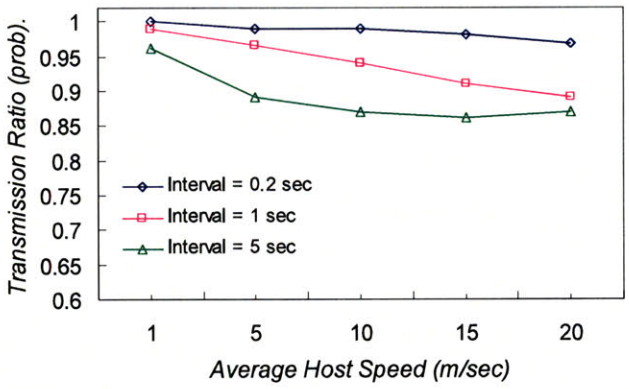


Figure 69. AHBP-EX: Sensitivity of transmission ratio to beacon interval

Figure 70 also shows the cost of the shorter interval in terms of broadcast overhead (*total packets transmitted per node per broadcast)* and compares the costs to that incurred by the Counter-based scheme. The shortest interval (0.2 seconds) is required for AHBP-EX to have a transmission ratio above 95 percent, but the total transmissions needed are

nearly twice that required by the Counter-based scheme. Furthermore, while a node only transmits a beaconing packet per update interval, it will receive a beaconing packet from each of its neighbors. The reception overhead is high, especially in dense networks.

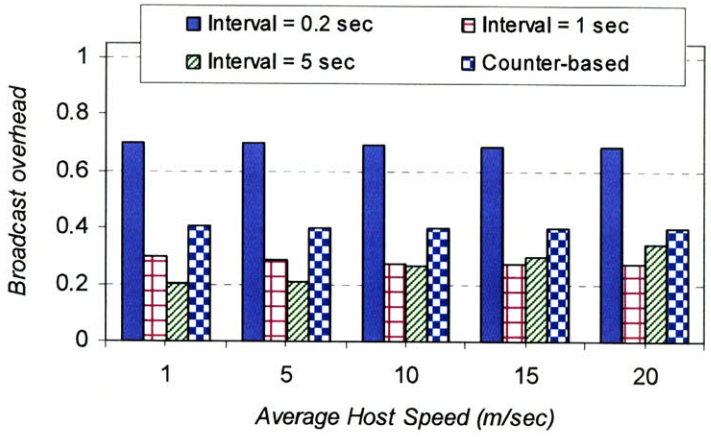


Figure 70. AHBP-EX: Sensitivity of broadcast overhead to beacon interval

After all, we need to carefully consider the tradeoff between the transmission ratio and the packet overhead per a broadcast, even both in sparse and dense networks. In other words, since overhead of the beaconing packets was augmented over a moderately low number of broadcast packets, it is recommended to lengthen the interval of the beacon signals. A higher traffic rate, while hiding the beacon packet cost in a larger number of total transmissions, may allow the interval of beacon signal to be shortened. However, in the high traffic rate, practically the overhead is troublesomeness: decreasing the beacon signal interval still incurs node resource costs that outweigh the performance gains.

In BRSP, the interval of the beaconing remains constantly at five seconds. Instead, as discussed in Section 4.2.7, it applies "mobility notification" method to intelligently deal with the high mobile surroundings. In other words, it compensates the transmission ratio in mobile networks as timely providing broadcast nodes with the information of changing topology caused by the host mobility. This makes it possible for the BRSP to adapt to the mobile environments. In the simulation, however, the mobility notification was not

implemented due to the decision time to send the mobility messages. The applicability remains still a further study.

5.3.5 Combination

In the previous three scenarios, we scrutinized particular network conditions by varying node density, congestion, and host mobility. In order to isolate the effects of those changes, we only varied one parameter in each scenario. The downside is that the behavior at a set of constant parameters may be different than the behavior at another set of constant parameters. Moreover, we risk missing the combined effects of mobility, congestion, and node density.

Table 8. Simulation parameters in the combined network scenario

Trial	1	2	3	4	5	6
No. of nodes	30	40	50	60	70	80
Avg. Speed (m/sec)	1	3	7	11	15	20
Packet sending rate	5	20	35	50	65	80

We attempt to address those concerns. To perform a complete scenario like a real network, we simulate all combinations of node density, traffic source rates, and average node speed. In other words, we evaluate the BRSP protocol as the severity of the network condition increases. We designed the six trials so that Trial 1 takes a combination of the least severe conditions and Trial six takes a combination of the most severe conditions, similar to the simulation mode in [42]. As shown in Table 8, we aggregate parameters into six trials to give an overview of performance at combined conditions. We maintain this trend of increasing severity along the x-axis. This aggregation of the three previous scenarios provides several bits of important information. Among others things, it demonstrates how our proposed protocol reacts in real networks. It allows us to gauge relative importance of node density, mobility, and congestion to the overall performance of the BRSP protocol. It also illustrates the general limits of each protocol for a given network environment.

5.3.5.1 Analysis

As the severity of the network environment increases, it is expected that each protocol has a "breaking point" in terms of its ability to deliver packets. Figure 71 demonstrates it in terms of the transmission ratios for each protocol according to each trial. The BRSP scheme is the highest performer through the whole trials: Even if suffering from the high mobility (e.g., trials 4 to 6), the reliable transmission protocol of BRSP results in maintaining the good transmission ratio in the average. The Simple Flooding breaks first, after Trial 2. The Counter-based breaks second after Trial 3, and the Neighbor Knowledge protocols, the SBA and the AHBP-EX break last after Trial 5 (e.g., 0.95 percent below). Especially, in the SBA scheme, after the trial 5, congestion appears to catastrophically interfere with its ability to deliver packets. This shows that the simple adaptation related to the interval of beacon messages was not sufficient to handle the congestion caused by high packet origination rate coupled with high node count. A more robust adaptation, which uses longer CAD for example, could be possible and be an appropriate topic for future research. The changing topology causes AHBP-EX to be the worst performer through the first three trials. Except fro BRSP, however, AHBP-EX degrades the most gracefully, and in Trial 5 it outperforms all other protocols by roughly over 20 percent.

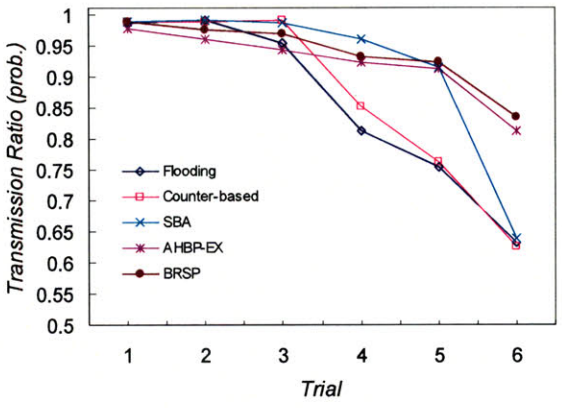


Figure 71. Transmission ratio according to increasing severity of network conditions

The number of re-broadcast nodes for each trial is shown in Figure 72. All protocols experience an increase in the number of re-broadcast nodes as the severity of the network environment increases. As expected, the neighbor knowledge methods including the BRSP have the fewest number of retransmitting nodes. In fact, the Counter-based scheme appears to mimic the behavior of Flooding after Trial 3 in terms of both the number of re-broadcast nodes and the transmission ratio provided.

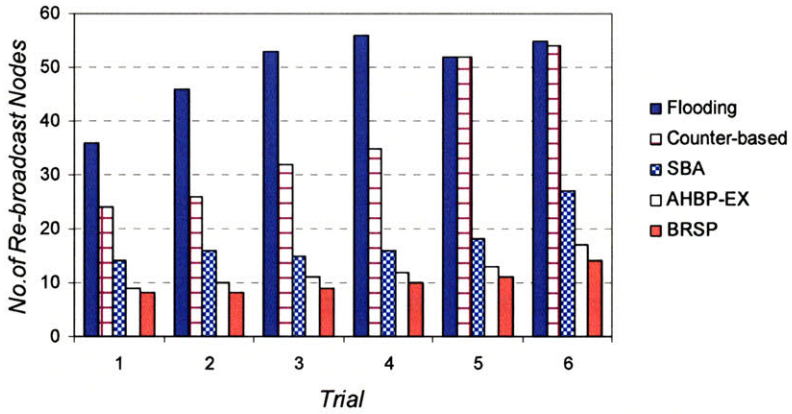


Figure 72. Number of re-broadcast nodes according to the increasing severity of network

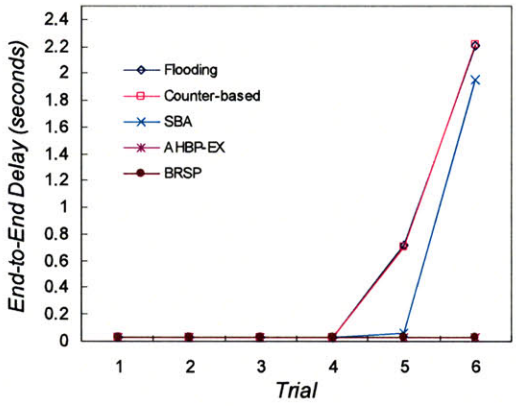


Figure 73. End-to-end delay as severity of network conditions increases

As the last experiment, we performed a test to see how the end-to-end delay is as network severity increases. The end-to-end delay results follow the trends of the transmission ratio. That is, the BRSP and AHBP-EX show constant end-to-end delays, respectively.

All in all, in the combined network scenarios, our proposed scheme, BRSP outperforms the remaining protocols except for the high mobility situation. That is, those experiments show that the BRSP could be applied to real networks that do not require high mobility.

5.3.6 Lessons learned

Our performance evaluation of the BRSP and the comparative broadcast protocols shows the followings:

1. In viewpoints of reliability and scalability, the Neighbor Knowledge methods outperform the Counter-based scheme. As the number of nodes in the wireless ad-hoc network is increased, the performance the Counter-based scheme is disproportionately degraded while the Neighbor Knowledge methods approximate the MCDS closely. Utilizing the more intelligent neighbor knowledge method, the BRSP requires the least number of rebroadcast nodes. That is, in terms of number of re-broadcast nodes, it is the most scalable method in density networks. Through the experiments to adapt threshold values of the Counter-based scheme, we anticipate that the scalability of BRSP protocol would be further improved by choosing the proper threshold value dependent on network density.

2. In the congestive situations, we conclude that in general, the Neighbor-knowledge methods outperform the Counter-based methods. In addition to the neighbor-knowledge, the BRSP making intelligently use of congestion control became the best performer at the congestive networks. On the other hand, the protocols that utilize a CAD, such as the Counter-based scheme and SBA, fail to operate efficiently in congested networks, because of the inability to minimize the number of re-broadcast nodes and the inherent redundancy of re-broadcasting nodes, respectively. Based on the experiments, we could estimate that the performance of the Counter-based protocol could be improved by adapting it to a node's neighbor count and local congestion level. As well, we anticipate that use of the reliable transmission mechanism for broadcasts would have the protocols much reliably work in the congested situations. In results, the improvements would make

the scheme deliver the higher number of packets in congested networks. The extension, however, negate the inherent simplicity of the protocol, which is its most attractive feature. The improved performance corresponds to higher algorithmic cost and to processing cost of control signals reacting efficiently to the jungle.

3. The Neighbor Knowledge methods such as AHBP and AHBP-EX that do not use local information to determine whether to re-broadcast have difficulty in mobile networks. Nevertheless the use of local neighbor knowledge, our proposed scheme, BRSP suffers from the dynamic host mobility. The reason is that the highly dynamic movements of nodes cause creation of many outlander nodes that are not affiliated with a broadcast group. In the end, this leads the broadcast transmissions to fail to operate correctly. In addition, in such the mobile environments, the broadcast overhead caused by the control packets used to create or maintain the viral broadcast queue within a viral ad-hoc cell aggravates the network conditions.

4. BRSP provides the best performance in the most severe network environment. Unfortunately, its sensitivity to node mobility produces the lowest transmission ratio in networks where the dynamic changes of the topology often occur.

Based on our performance evaluation, we conclude that BRSP is recommended for static topologies or extremely congested networks. Also, our evaluation of broadcast protocols provides other directions for valuable future research as well. We demonstrate that BRSP approximates MCDS within 10 percent in dense networks. However, the fact that the protocol all rely on extremely accurate neighbor knowledge information (via the broadcast queue) makes them unattractive in dynamic networks. Therefore, further work in algorithmic optimizations should take a back seat to research in making these protocols effective in mobile networks. As suggested in Section 4.2.7, "mobility notification" method to intelligently deal with the high mobility would be a good example for the future research on the BRSP.

CHAPTER 6. CONCLUSION

6.1 Contributions

The system described in this thesis touched on two different areas, each of which is aimed at enhancing the event awareness and the group communications in local life through the use of mobile devices. The first area we have tackled is about creating a cognitive platform for social engagements mainly driven by locally created events. The other is to invent a core engine to technically support the platform in the wireless/mobile infrastructure-free milieu.

For the cognitive platform, we first defined *"Social Event Network"* as a space, where people are seasonably aware of what is happening around them, and then designed architecture of the platform supporting the event-driven social networking. Specifically, the platform is wrought by a personal and group-wise broadcast system, named *"XCast,"* in the wireless/mobile P2P network. In other words, the social events are created through the personalized yet group-wise broadcast operations. The platform consists of personal broadcasters, broadcast groups, broadcast resource schedulers, event collection servers, and an event mining tool, called *"Social Event Navigator"*: Each personal broadcaster senses others and builds a broadcast group to share information they want. The broadcast resource scheduler makes it possible for the broadcasters to efficiently share data without any undesirable effect. These components are required for event creation. The event collection server then plays a role of gathering the events created by the interaction of the components above and of running the *Social Event Navigator.* Finally, the event navigator provides the users with intelligence to gaze or search locally created events in the real-time way.

In addition to the platform, we implemented the XCast system on the state-of-art mobile device, OLPC XO for wireless mesh networking. The system includes two participatory applications for the multimedia contents sharing in cohesive and orderly manners.

As a core engine to support the cognitive platform, we invented a protocol to make it possible for broadcast operations to reliably work in the wireless ad-hoc/mesh networks. We call it *"Broadcast Resource Scheduler Protocol (BRSP)."* Its ultimate goal is to transform a wireless ad-hoc/mesh network into a cognitive network. To achieve this, BRSP has four management functions such as *Group Management, Broadcast Queue Management, Routing Management,* and *Mobility Management.* First, the Group Management encourages the personal broadcast nodes to build a broadcast group as a unit of broadcast transmission. This is a group enabling all nodes to make cognizant of each other. Next, the Broadcast Queue Management helps share all the status of broadcast transmissions in the group and synchronizes those information, in order to schedule broadcast operations without any packet collision or drop. Then, the BRSP protocol uses a group-based broadcast transmission method with acknowledgement to make the broadcast sending more reliable in the wireless ad-hoc/mesh networks. If the target destination nodes exist beyond its own broadcast group, the Routing Management starts to run a cognitive routing protocol, dubbed *"Two-Zone"* routing protocol. The protocol is used to support the multi-hop routing and then to reduce the baleful effects on the packet collision and the overhead caused by the re-transmission of broadcast packets in the multi-hop routing. Finally, the Mobility Management uses a Mobility Notification/Detection method to face the disruptions of the broadcast group resulted by the system's breakdown or host mobility.

To evaluate the performances of the BRSP protocol, we had two different experiments: One was to investigate the performances in real-testbed constituting of OLPC XOs, and the other was to use a simulated testbed to compare the existing reliable broadcast protocol with the BRSP. In the former, we examined the effects on the broadcast packet collision and the overhead caused by the blind broadcasting and the broadcast jumble. In the test, the BRSP outperformed the legacy 802.11 broadcast protocol installed on the OLPC XO laptop. In the simulation tests, we built a large-scale wireless ad-hoc network consisting of 100 nodes, and delved the performance parameters such reliability, scalability, the overhead, congestions, and mobility. The BRSP protocol also largely outperformed other protocols in all the performance tests except for the mobility scenario.

6.2 Future Work

To make a system cognitive requires as many sensing processes as we are socialized with someone. Even if we designed and developed various functions and protocols to support the cognitive platform, some core functions and modules still remain outstanding. In the section, we describe issues we need to figure out and the future work.

First, to create a cognitive framework for the Social Event Network, we need to complete the functions of the event collection server. The part we need to still figure out of the server's functionalities is about the decision of event location. That is, it makes use of Triangular Address Sharing method, which finds the geographical location of an event, given host's IP address. In general, that method would not be applicable. For the reason, IP addresses are allocated arbitrarily, as there is no inherent connection between an IP address and its physical location, and there is no reliable method to do the trick. Some helpful experimental work, for example, utilizing a DNS Resource Record [49] [50], has been performed, however. The trials provide some feasible room we could explore for that issue.

Next, the issue we need to delve is the "mobility support" of the BRSP protocol. This is a very crucial part to decide the applicability of BRSP protocol in mobile ad-hoc/mesh networks. To achieve this, we have designed and implemented the Mobility Notification/Detection module of Mobility Management using three control messages. However, the notification scheme, called "NTL/CTL mobility pair" was fully not implemented due to the difficulty to measure the dynamic behaviors of mobility and so the unintelligible decision time to send NTL, as discussed in Section 4.2.7. One possible approach is to predict the mobility dynamics. It is still unstable and tricky to use that prediction.

Finally, we need to further scrutinize the applicability of participatory applications and the efficiency of the BRSP functions we have implemented. That is, we leave the user experience tests of the applications-by COUHES (Committee on the Use of Humans as

Experimental Subjects)-a future work. Also, to examine the efficacy of our implemented protocol, the large scale experiments with OLPC XOs remain a future work.

-

BIBLIOGRAPHY

[1] XCast website, http://www.media.mit.edu/~starsu/xcast.

[2] M. Stephens, "The History of News - 3rd Ed" Oxford University Press, New York, 2007.

[3] "Deadly Rampage at Virginia Tech" diagram of Norris Hall, The New York Times, April 22, 2007.

[4] S. Ni, Y. Tseng, Y. Chen, and J. Sheu. "The broadcast storm problem in a mobile ad hoc network," In Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), pages 151–162, 1999.

[5] J. Broth, D. B. Johnson, and D. A. Maltz. "The dynamic source routing protocol for mobile ad hoc networks," 1998. Internet Draft.

[6] Z. J. Haas and M. R. Pearlman. "The zone routing protocol (ZRP) for ad hoc networks," 1998. Internet Draft.

[7] M. Jiang, J. Li, and Y. C. Tay. "Cluster based routing protocol (CBRP) functional specification," 1998. Internet Draft.

[8] C. E. Perkins and E. M. Royer. "Ad hoc on demand distance vector (AODV) routing," 1998. Internet Draft.

[9] "Channel Deployment Issues for 2.4 GHz 802.11 WLANs," Cisco Systems, Inc. white paper, February 07.

[10] "802.11a: A Very High-Speed, Highly Scalable Wireless LAN Standard," Proxim Wireless Network, white paper, 2003.

[11] "Practical Considerations for Deploying 802.11n," Siemens Enterprise Communications, white paper, February 2008

[12] G. Anastasi, M. Conti and E. Gregori, "IEEE 802.11 ad hoc networks: Protocols, performance and open issues, in Mobile Ad hoc networking," S. Basagni, M. Conti, S. Giordano and I. Stojmenovic (eds.), IEEE Press and John Wiley and Sons, Inc., New York, 2003.

[13] Z.J. Haas and J. Deng, "Dual busy tone multiple access (DBTMA)-A multiple access control scheme for ad hoc networks," IEEE Transactions on Communication, vol. 50, no. 6, pp. 975–985, June 2002.

[14] J. L. Sobrinho, R. de Haan, J. M. Brázio, "Why RTS-CTS is not your Ideal Wireless LAN Multiple Access Protocol," in Proc. IEEE WCNC 2005, New Orleans, Louisiana, March 2005 .

[15] J. Peng. "A New Scheme to Avoid Collisions for Broadcast Packets in Wireless LANs," Communications Letters, IEEE, Vol 11, Issue 9, pp. 762-764, September 2007

[16] K. Tang and M. Gerla, "MAC reliable broadcast in ad hoc networks," Military Communications Conference, MILCOM, pp. 1008–1013, Oct. 2001.

[17] M. Sun, L. Huang, A. Arora, and Ten-Hwang Lai, "Reliable MAC layer multicast in IEEE 802.11 wireless networks," International Conference on Parallel Processing, pp. 527–536, Aug. 2002.

[18] C. Chiu, E. Wu, and G. Chen, "A reliable and efficient MAC layer broadcast (multicast) protocol for mobile ad hoc networks," Global Telecommunications Conference, pp. 2802–2807, Dec. 2004.

[19] W. Si and C. Li, "RMAC: a reliable multicast MAC protocol for wireless ad hoc networks," in Proc. IEEE ICPP 2004.

[20] OLPC: http://www.laptop.org.

[21] OLPC Wiki: http://wiki.laptop.org/go/The_OLPC_Wiki.

[22] VIA mini-ITX board: http://www.via.com.tw/en/products/mainboards/.

[23] Madwifi Wireless Card: http://madwifi.org/wiki/Compatibility/Netgear.

[24] Silver, H Ward, "Amateur Radio for Dummies," Indianapolis: Wiley Publishing. ISBN 0764559877. OCLC 55092631, April, 2004

[25] YouTube website, http://www.youtube.com.

[26] FluidVoice website, http://www.media.mit.edu/~kwan/Research/FV/fv.pdf.

[27] Podcast website, http://www.apple.com/itunes/store/podcasts.html.

[28] WikiCity, Rome: http://senseable.mit.edu/wikicity/rome.

[29] Nokia Lifeblog: http://www.nokia.com/lifeblog.

[30] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath, "Flooding for reliable multicast in multi-hop ad hoc networks," In Proceedings of the International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication (DIALM), pages 64–71, 1999.

[31] J. Jetcheva, Y. Hu, D. Maltz, and D. Johnson, "A simple protocol for multicast and broadcast in mobile ad hoc networks," Internet Draft: draft-ietf-manetsimple-mbcast-01.txt, July 2001.

[32] W. Peng and X. Lu, "On the reduction of broadcast redundancy in mobile ad hoc networks," In Proceedings of MOBIHOC, 2000.

[33] H. Lim and C. Kim, "Multicast tree construction and flooding in wireless ad hoc networks," In Proceedings of the ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM), 2000.

[34] L. Lovasz, "On the ratio of optimal integral and fractional covers," Discrete Mathematics, 1975.

[35] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying: An efficient technique for flooding in mobile wireless networks," Technical Report 3898, INRIA - Rapport de recherche, 2000.

[36] T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot, "Optimized link state routing protocol," Internet Draft: draft-ietfmanet-olsr-06.txt, September 2001.

[37] W. Peng and X. Lu, "AHBP: An efficient broadcast protocol for mobile ad hoc networks," Journal of Science and Technology - Beijing, China, 2002.

[38] W. Peng and X. Lu, "Efficient broadcast in mobile ad hoc networks using connected dominating sets," Journal of Software - Beijing, China, 1999.

[39] J. Sucec and I. Marsic, "An efficient distributed network-wide broadcast algorithm for mobile ad hoc networks," CAIP Technical Report 248 – Rutgers University, September 2000.

[40] I. Stojmenovic, M. Seddigh, and J. Zunic, "Internal node based broadcasting in wireless networks," In Proceedings of the Hawaii International Conference on System Sciences (HICSS), 2001.

[41] J. Wu and H. Li, "On calculating connected dominating sets for efficient routing in ad hoc wireless networks," In Proceedings of the International Workshop on Discrete Algorithms and methods for Mobile Computing and Communication (DIAL-M), pages 7–14, 1999.

[42] Toilers Laboratory: http://toilers.mines.edu/Public/CodeList.

[43] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," In Proceedings of European Symposium on Algorithms (ESA), 1996.

[44] A. Levitin, "Introduction to the Design and Analysis of Algorithms," Second Edition, Villanova University, ISBN-13: 9780321358288, Addison-Wesley, 2007

[45] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, "Multi-hop wireless ad hoc network routing protocols," In Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), pages 85–97, 1998.

[46] J. Boleng, "Normalizing mobility characteristics and enabling adaptive protocols for ad hoc networks," In Proceedings of the IEEE Local and Metropolitan Area Networks Workshop (LANMAN), pages 9–12, 2001.

[47] T. Camp, J. Boleng, B. Williams, L. Wilcox, and W. Navidi, "Performance evaluation of two location based routing protocols," In Proceedings of INFOCOM, 2002.

[48] Center for Future Civic Media: http://civic.mit.edu/

[49] C. Davis, P. Vixie, T. Goodwin, and I. Dickinson, "A Means for Expressing Location Information in the Domain Name System", RFC 1876, January 1996.

[50] C. Farrell, M. Schulze, B. Pleitner, and D. Baldoni, "DNS Encoding of Geographical Location", RFC 1712, Curtin University of Technology, October 1994.

[51] OhmyNews: http://english.ohmynews.com/

[52] Bookstein, Abraham, "Informetric distributions, part I: Unified overview," Journal of the American Society for Information Science 41, page 368–75, 1990.

[53] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a Fair Queuing Algorithm," Internetworking: Research and Experience, Vol. 1, No. 1, pp.3-26, 1990.

[54] I. Stoica, S. Shenker, H. Zhang, "Core-Stateless Fair Queuing: Achieving approximately fair bandwidth allocations in high speed network," in Proc. ACM Sigcomm, Sept 1998, Vancouver, Canada.

[55] Roomba Platform: http://www.irobot.com/sp.cfm?pageid=305

[56] Google MAP API: http://code.google.com/apis/maps.

[57] Gstreamer library: http://gstreamer.freedesktop.org.