2009

Envisioning Social Computing Applications on Wireless Networks

Siva Gurumurthy
University of Massachusetts Amherst

ENVISIONING SOCIAL COMPUTING APPLICATIONS ON WIRELESS NETWORKS

A Thesis Presented

by

Siva Gurumurthy

Submitted to the Graduate School of the

University of Massachusetts in partial fulfillment

of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL AND COMPUTER ENGINEERING

UNIVERSITY OF MASSACHUSETTS AMHERST

February 2009

Electrical and Computer Engineering

ENVISIONING SOCIAL COMPUTING APPLICATIONS ON WIRELESS NETWORKS

A Thesis Presented

by

SIVA GURUMURTHY

Approved as to style and content by:

____

Aura Ganz, Chair

____

Wayne Burleson, Member

____

Ramgopal R. Mettu, Member

____

Christopher V. Hollot, Department Head
Electrical and Computer Engineering

DEDICATION

To,

My parents and friends

ACKNOWLEDGMENTS

I would like to express my deep appreciation and gratitude to my advisor, Prof. Aura Ganz, for her guidance, suggestion and support throughout my graduate studies. Without her valuable support, I would never have been able to bring my ideas and research to realization. I believe experiences gained under her guidance will be a guiding force in my professional career as well as my personal life.

I would like to thank Prof. Ramgopal R. Mettu and Prof. W. Burleson for being a part of my thesis committee and for guiding me through my thesis.

ABSTRACT

ENVISIONING SOCIAL COMPUTING APPLICATIONS ON WIRELESS NETWORKS

DECEMBER 2008

SIVA GURUMURTHY

B.TECH, INDIAN INSTITUTE OF TECHNOLOGY, GUWAHATI

M.S., UNIVERSITY OF MASSACHUSETTS, AMHERST

Directed by: Professor Aura Ganz

Wireless mobile internet market is still an unprecedented, uncaptured territory for cellular service providers. The shortage and high cost of downlink data bandwidth in a cellular network has remained a huge factor for the slow growth of data services in mobile devices. Although there has been a significant evolution in telephony infrastructures in form of 3G and 4G systems, the potential of high speed ad hoc network for sharing cellular spectrum have not been realized to its full potential. Like (e.g. Verizon) users can share voice minutes with friends, there is a potential for sharing the unutilized cellular bandwidth among friends to increase net data speed. In a scenario like a football stadium where people visit in groups, although a lone phone cannot stream a high quality replay video, unused cellular bandwidth of proximate friend's devices can automatically be used in real time to view the replays. An available secondary ad hoc network such as Wi-Fi or Bluetooth in phone can be used for sharing this cellular bandwidth. Thus, we propose BuddyShare, a novel social-based automatic bandwidth sharing overlay platform on short range ad hoc devices to increase net data speed. The motivation stems from the fact that the location of mobile users tends to be

clustered to form “people hotspots” such as conferences, stadiums, stations, buses and trains. For example, in a scenario like a football stadium where people visit in groups, although a lone phone cannot stream a high quality replay video, unused cellular bandwidth of proximate friends’ devices can automatically be used in real time to view the replays.

Our work creates an overlay on horizontal ad hoc network to enable users to form a group among socially trusted members who can collaboratively share their data connections. Social trust is automatically derived from social relationships obtained by mining mobile-phone behavior pattern. This work aims to improve the overall utilization of the data connection, and increase the data rate of individual users without compromising their privacy and unauthenticated usage. The user privacy is preserved by using the bandwidth resources of only socially trusted member of the user, which also guarantees against unauthenticated exploitation of expensive bandwidth. Our proposed work promises to deliver win-win situation to users, content providers and service providers. The advantages of users are: 1) Increased data rate for the same cost.2) Secure and trusted overlay based communication for sharing resources. The advantages for the service providers are manifold: 1) Customer increase: More customers will avail the data plan due to social influence. 2) Customer retention: [18] Customers part of the social-cum-adhoc network are least likely to leave the network.3) Group subscription: Service provider can get bulk subscriptions as collaborative groups increase data rate.

In this work, we address some key technical issues of developing a socially aware overlay collaborating medium. Some of the addressed functionalities associated with the overlay formations are group discovery, creation, management and actual data distribution. This proposal also accounts the computation of social trustworthiness by using standard social networking analytics. We also account the several key technical challenges associated with management of overlay on mobile nodes and trust computation using abstract social network. In order to verify the usefulness of BuddyShare, we collected realistic datasets from various sources (questionnaires, mobile device logs, social networking portal) and conducted analyses and simulations on it. The analyses concluded that sample users from the dataset shared sufficient social trustworthiness. The real events from the datasets were captured in the simulations. These simulations showed that, by using Bluetooth as a horizontal ad hoc medium, an user can scale his data speed three times on average for sufficient duration per day.

This thesis achieves the following objectives:

1) It presents a comprehensive design for an overlaid social based internet sharing platform called BuddyShare.

2) It presents a social analysis to validate the concept of social trust among users.

3) It delivers a flexible simulation platform to realistically simulate mobile phones with dual interfaces.

4) It presents the results of simulations of real events captured from the device logs of sample users. These results conclude the usefulness of BuddyShare work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

Wireless industry has stepped into its next generation, the service providers are competing to grab the market share by providing "universal" access to wireless services such as internet, email, etc. In particular, the cellular service providers are competing to acquire new users and retain the existing customers. Data plans are new value addition to the cellular network, and providers are coming up with new business models to market their data plans. The data plans are not very popular due to the cost and speed factor associated with the challenges of mobile environment. We present an attractive proposition for increasing data usage among customers by enabling a social based sharing platform on the secondary network medium of mobile phones.

Mobile users pay huge for data access via the backbone network of the service providers. Accessing large files from internet via bottle-neck cellular links at a high speed with low cost is still a challenging problem in the satellite and cellular communications. It has been shown in [10][11] that the data access via cellular infrastructure yields a poor speed not because of shortage of spectrum, but due to inefficient management and allocation of the channel resources. As an instance, [10] shows that usually channels in licensed spectrum are underutilized in current wireless cellular communication and an overlaid secondary Ad hoc network seems to be one of the several approaches to improve the channel utilization. However, under realistic conditions with people carrying mobile phones every where, forming an overlay network with all the physically close people may result in bandwidth theft. So there is a need for forming "socially trustable" (defined by social relationships like friends, family, kith-and-kin etc) closed user groups while forming the overlay network.

Our proposed work creates an overlay on horizontal ad hoc network that will enable users to form group among socially trusted members who can collaboratively share their resources. This work aims to improve the overall utilization of the data connection, and increase the bandwidth of individual users without compromising on their privacy. The user privacy is preserved by using the bandwidth resources of only socially trusted member of the user. This trustworthiness among group members also guarantees against unauthenticated exploitation of expensive bandwidth. Trustworthiness is automatically derived from social network analytics with minimal user intervention.. Our underlying assumption to this work stems from the fact that mobile users tend to form "people hotspots" and most of the bandwidth is purchased as a package and not based on the usage. Thus when the resources are not used but paid for, our overlay protocol scheme enables mechanism to share these resources to "trustworthy" neighbors.

Advantages

Our work promises to deliver win-win situation to service providers, content providers and the customers. The advantages for the service providers are manifold:

- *Customer increase*: More customers will avail the data plan due to social influence,
- *Customer retention*: [18] Customers part of the socio adhoc network are least likely to leave the network.
- *Group subscription:* Service provider can get bulk subscriptions, due to the increased theoretical bandwidth.
- *No infrastructural changes needed*: Low load on the cellular network back end.

Similarly, the customers are benefited in the following ways:

- Increased bandwidth for same cost.
- Secure and trusted overlay based communication for sharing other resources.

A social based overlay on mobile phones presents range of technical challenges ranging from managing wireless networks to using social relationships in forming a mobile P2P platform.. The complete list of technical challenges is presented in the next section.

1.1 Technical Challenges

- The constructed overlay should take care of resource constraints such as bandwidth, power, range and losses associated with the wireless environment. In order to calculate resource cost, the overlay should implement network layer mechanisms to measure the statistics of wireless characteristics. Some recent works [21] [22] have considered physical proximity while constructing an overlay network. Such approaches should be adapted to constraints associated with a short range ad hoc network.
- The overlay should take care of social trust constraints among the nodes. A social trust needs to be automatically computed between every ordered pair of physically reachable nodes. In order to calculate social trust, the nodes have to obtain the social context of the user. While acquiring social context, the mechanism should take care of user privacy, aggregation of available social network data, accounting the missing data and abstraction of incomplete data.
- Social network: Data collection, and anonymising techniques [27][28] need to be suitably chosen for collecting social networking data. If the social network is

provided by the third party, an appropriate privacy preserving scheme need to be devised to exchange information between clients and servers.

- The unstructured topology formation of the overlay network needs to be addressed. The protocols for bootstrapping, peer discovery, peer joins and peer losses needs to be designed and addressed. The node states and their transitions need to be defined.
- Topology maintenance [23]: An overlay adaptation based on mobility needs to be addressed. Dynamic overlay creation incurs an extra overhead of discovering the nodes and sharing keys to establish the overlay periodically. A cross-layer approach is required to reorganize the application level overlay.

Once the overlay is formed, the cellular-bandwidth needs to be managed with resource allocation algorithms. The peers need to schedule their jobs depending on the availability and associated cost. A data delivery protocol based on an ad hoc network needs to be developed. A possible approach is to use transient asynchronous communication. The requestor does not wait for the reply, and the receiver replies to the sender opportunistically. Requestor has an adaptive time out mechanism to eliminate the stale data or lost node. Such opportunistic data delivery mechanism needs to be devised.

1.2 Proposed Methodology

Our work addresses the technical challenges mentioned above in three phases:

First, an overlay construction mechanism is developed. This overlay construction is handled using two process states of a node: Peer discovery, and Group creation. Unlike large scale peer-to-peer networks, BuddyShare relies on data connections of only few physically reachable peers. Hence, peer discovery is similar to node discovery process inherent in ad hoc networks. A cross-layer interaction is used to convey the snapshot of ad hoc network to upper layers for overlay construction. In the Group creation process, the initiator shortlists peers from physically reachable list based on satisfied social trust and resource cost. A group creation protocol establishes overlay links between the initiator and all its peers. This group forms a tree routed at the initiator. Only root can issue the download jobs for its leaf peers. So, a root peer is also termed as leader and leaf peers as slaves, where leader issues download jobs to slaves. A slave of one group can be leader in another group.

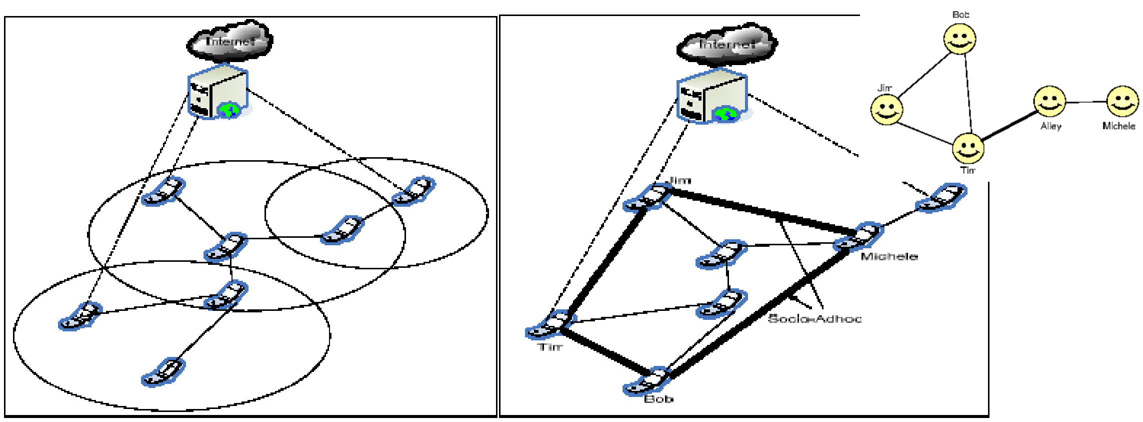

Figure 1.1: Physical topology

Figure 1.2: Socio Adhoc Overlay

Figure 1 : Logical picture of social overlay

Second, a job distribution mechanism is developed which is handled by a Job Arbiter process state in a node. A job is defined as a download request for a file chunk. The jobs are created by packet stripping HTTP byte-range requests at application layer. The jobs contain segmented byte-range, expiration time, ID and other details. These jobs are distributed using input-output queuing mechanism.

Third, an overlay maintenance mechanism is developed. This mechanism is handled by Group maintenance process state of a node. This state defines the protocols for nodes to join and leave. When a node discovers new overlay, it can join as a leaf peer of the overlay tree. All the leaders update their slaves and pending jobs are redistributed. A slave loss is detected by Job Arbiter, when the head of line job does not get completed by the TTL value of the job. A slave loss causes leader to reorganize its group and redistribute the pending jobs. A leader loss is detected while data delivery and all the subsequent jobs from the leader are not served.

All the protocols for overlay formation, maintenance and job distribution are stateful. There are several limitations in ad hoc networks such as delays, losses and mobility, which affect the throughput in an ad hoc network. However, this achievable throughput of an ad-hoc network is still NOT a bottleneck when compared to the capacity of cellular networks assigned to a single user. Even with overheads due to overlay formation, maintenance and job distribution, the vertical data connection can be efficiently shared to increase the net data rate. Apart from the three phases as our contribution, our vision to accumulate social network from mobile devices and use it to compute a social trust while forming overlay links is the first-of-its-kind effort and will open a new bread of distributed applications. Though social based overlay has been recently studied and applied in file sharing P2P systems, there has not been any social based automatic collaboration platform for ad hoc networks to the best of our knowledge. The social factor is essential while forming overlay links as bandwidth and expensive resources are automatically shared only with friends and not with others.

Our work depends on some assumptions and hypothesis. We conducted experiments and garnered datasets from already conducted experiments to validate some of our hypothesis. *Though, we cannot concretely state the hypothesis as always true, our* *analysis on realistic data set showed that it is true and our work is worthwhile in most of cases.*

1.2.1 Experiments

We collected datasets from questionnaires asked to students belonging to different years, classes and cultural backgrounds to understand the need of hour. We also collected the dataset from the reality project undertaken by MIT [9]. These dataset are collected from living samples, where each individual possess a specially designed mobile device to collect the log of their actions pertaining to their interactions and daily activities. Apart from questionnaires and reality data set, we also launched a social networking web portal called Connect during the course of a semester motivating students to discuss the course material through the portal. All these experiments had shown us some significant social characteristics such as socially closer people are often found physically closer as well, and they are willing to share their data connection when they buy internet as a package. Some felt uncomfortable with existing data plans and they really look forward for an improvement in the data plans, and availability of mobile phones everywhere prompts them to check their emails but the speed is often very slow. Our results section has some statistics that infers the conclusion which clearly points the same direction as we are progressing.

1.2.2 Simulation

To evaluate BuddyShare performance, we developed a NS-2 [29] based simulation that closely matches a mobile user behavior. The simulated mobile phones have two network interfaces: one for voice and data services in the cellular network (uses ns-2 extension EURANE [43]), and another for short range ad-hoc Bluetooth network (uses

UCBT simulator [30]). An emulated BuddyShare overlay protocol using the sockets provided by [44] is implemented for overlay communication. The overlay communication has been implemented by modifying the Gnutella simulator [45] for wired networks to work in an ad- hoc network as proposed in [46]. It has been modified to form overlay links based on the social trust computed from users' social network. We used another ns-2 extension [47, 48] for getting the accurate web user behavior. The simulated phone picks up a request-response pair from the distribution and schedules a web access during the overlay period. We measure the download rate across different overlay formation algorithms for a period of one month. The simulation results conclude that BuddyShare will enhance download rate by 100% on average by collaborating with friends. We present a scenario in the next section depicting the advantage of socio-adhoc BuddyShare overlay.

1.3 Scenario

Bob, Jim, Tim, his wife Alley, and Alley's friend Michele, are taking off a trip to Mauritius in a cruiser ship. Bob and Tim have Blackberry phones, Alley and Michele have PDA phones, and Jim uses his smart phone to browse internet and add friends in MySpace.com. Bob, Jim and Tim are high school friends, and part of Umass'01 Medical school community in FaceBook.com (Social network of the scenario is shown in Fig. 1). While Bob and Jim are swimming, Tim found an interesting video in youtube.com, he tries to stream them in his Blackberry phone. His limited internet access does not let him stream the video continuously. He subscribes to socio-adhoc service and deploys the application and enters the URL of the stream which he wants to stream.

Bob and Jim have already subscribed to socio-adhoc services, their internet access is shared with Tim and he is watching the video with no intermission.

Michele is working in a cancer research and needs to access large amount of data offline. She requests an offline storage service from socio-adhoc overlay. Bob has a 10 GB of free space donated to the socio-adhoc service, receives a notification in his socio-adhoc application about Michele's request. Bob do not know Michele, but agrees to share his memory as she is his friend's wife's friend. Michele is happy with the space and agrees to share her high speed internet connection to socio-adhoc service.

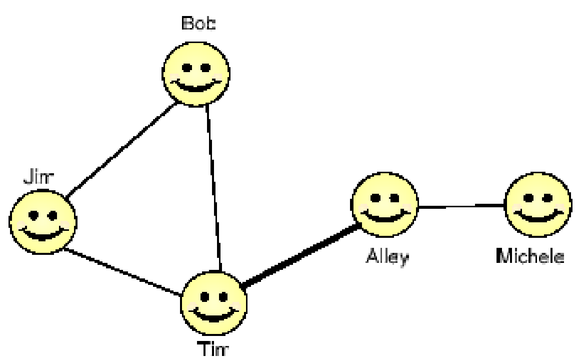

Figure 2: Social network of the scenario

Sharing and donating resources is not a new idea in networks. CPUs and memory are donated for various causes such as Cancer research and SETI. But sharing is purely a social responsibility driven by the trust that one individual binds with another and often people are tentative to share resources outside their social circle. Though the scenario presents an exaggeration, the interesting feature is that it is practically achievable with the existing technologies. It can be achieved by intelligently incorporating social network analytics in existing vertical and horizontal communications.

CHAPTER 2

PREVIOUS WORK ON OVERLAY AND APPLIED SOCIAL NETWORK

In this chapter, we present surveys on the existing works relevant to applied social network, in particular those works that are related to wireless networks that supports our cause. We also present relevant works on algorithms and approaches used to construct an overlay network on mobile ad-hoc network. We also throw some light on existing works on Bluetooth as an important mobile medium to share files in high speed.

An overlaid social based ad hoc network is a concept that borrows trustworthiness estimation from social network and is applied on wireless ad hoc network to form an overlay. In this section, our primary focus is to discuss some works that relate to forming P2P based overlay on wireless networks, and then we discuss some of the existing works that use social networks to enrich some concept of wireless network and related to our work.

Significant works are done on overlaid adhoc network. Overlaid adhoc network has nodes with two interfaces, one for communicating with ad hoc nodes in a horizontal ad hoc network and another for communicating to vertical backend network. Cellular network's licensed spectrum is usually underutilized and as shown in [10], overlaid adhoc secondary networks (ASN) is architected on this unutilized spectrum to improve the channel utilization. The secondary system as claimed could be peer-to-peer communication or wireless internet access on the left over spectrum. Spectrum is shared in this system. The primary system, which is owner of the cellular spectrum, needs to authorize and approve the use of secondary system to utilize when the channel is free. Our work draws parallel with the discussed concept; however it does not need primary system's authorization to improve the bandwidth. The adhoc network does not need to be constructed in the left over spectrum, as the overlay is a user driven grouping on personal area network where each individual cannot increase their bandwidth beyond the allocated bandwidth of service provider. Hence, more than one user needs to open up parallel connection to increase the bandwidth and form an overlay to redistribute the downloaded data to get a better overall throughput without taking help of service provider.

Work done in [32] deals with forming and maintaining overlay over wireless adhoc network. The overlay network is constructed with only two constraints: transmission range and strong connectivity of all the nodes as long as there is a spanning tree. This work supports the hypothesis that overlay can be maintained for a period of time as long as they are reachable using the wireless links. They motivate the idea of overlay on multi-hop adhoc network. Most of the overlay networks discussed is not convincingly trustworthy; in fact they guarantees only connectivity. Thus we need to incorporate social trustworthiness in forming an overlay on ad hoc network. So far, we have seen some works on overlay construction of wireless networks. Apart from initial overlay formation, the downloaded data needs to be routed back to the requestor node. Since the underlying topology may change by the time independent data chunks are downloaded, thus overlay needs to be reconstructed and needs to be maintained while the underlying physical network change. Overlay multicast protocol [21] builds a virtual mesh spanning all members of a multicast group. The virtual topology gradually adapts to the change in underlying network topology in a fully distributed manner. A source-based Steiner tree algorithm is used for constructing the multicast tree which progressively adapts to latest local topology. This work highlights the fact that while constructing virtual overlay, there could be different unicast physical tunnels sharing physical links, which results in redundant traffic on physical links. These existing literatures can be efficiently used to build an overlay that is aware of the physical wireless medium.

Let us discuss some of the works that relates to social aspects of the human holding the mobile device. The work in [20] presents a connection establishment mechanism in an adhoc network between devices by exploiting some kind of social trust between the humans who hold these devices. Social trust is established between devices based on the addresses in the device's address book. They build adhoc network only between these trusted members. While constructing ad hoc network, it uses a secure communication by means of exchanging hash value of a person's id (email address or phone number) along with the keys. Upon receiving the key and hash value pair, device on the other end searches all the addresses for the correct hash value. This work also discusses ways to prevent impersonation, Diffie-Hellman keys are exchanged in a secured channel. This method models an adhoc network connection establishment using a distributed approach, however for an application where expensive resources such as bandwidth and memory is shared, an address book based approach will not be sufficiently socially trust worthy. Moreover, the aforementioned work is a connection establishment mechanism between the devices, supplementing the Bluetooth authentication mechanism. Our work is not just setting up a trustworthy connection automatically between devices, but to create a group of physically reachable devices which are social trustworthy to each other. As the connection establishment is one-to-one, it does not use relay nodes to establish connection between more than 2 hop neighbors. Though the address book serves as one source of social relationship, it may contain addresses of several acquaintances who may not be willing to share their bandwidth. However under absence of any other external social networking information, address book serves as a good substitute. Thus our work will use a hybrid approach which will use both distributed approach and a centralized approach, whenever there is an availability of centralized social network data. Our hybrid approach uses social network analytics to decide between distributed and centralized approach in forming an overlay network.

Next, we discuss some of the works that uses social aspects in regards to wireless network. [35] models the mobility of an user by observing their social behavior. It uses the results of social network theory to effectively model a near-actual mobility distribution of mobile users in an ad-hoc environment on the basis of which, the ad-hoc protocols are formulated. This work tries to predict the mobility of humans in an Adhoc network. The fundamental idea is that by identifying a set of social clusters and frequency of hopping between these social clusters, it would model the movement behavior of humans. Another work done [31] in a delay-tolerant MANET routing, uses social network analysis to compute the next hop neighbor by estimating the neighbors centrality to the target. The node forwards the packet to the next hop neighbor in order to route to the destination. This work works on the hypothesis that a person belonging two groups could be used as a medium to transfer message from one to another, and social networking measures such as betweenness and centrality helps finding the correct person.

2.1 Building on Prior Works

In this section, we show how the proposed approach complements and builds on prior works.

The presence of multiple network interfaces in mobile devices has prompted considerable research in increasing WWAN bandwidth by sharing through secondary collaborating mediums. PTCP[49] proposes a transport level aggregating mechanism by inverse-multiplexing or striping packets across multiple interfaces. It deals with addressing problems such as packet reordering associated with TCP in wireless collaborating medium. This work assumes that all the WWAN links are attached to same device and hence do not consider the issues related with maintenance of an ad hoc group. MAR[50] proposes a bandwidth aggregating router that collects the bandwidth of all the unutilized channels and pools it together to increase the net bandwidth. It dynamically shifts load from poor quality to better channels. This work is constrained by the presence of an aggregate router which has simplified ad hoc group maintenance. Another work, MOPED [51] explores the collaboration of multiple devices. It has transport level bandwidth aggregating mechanism which distinguishes losses due to wireless and congestion. It also addresses the mobility of group users. However, MOPED does not account for the cost of internet by assuming a single owner for all the devices. Thus MAR and MOPED does not have a good incentive and coordination mechanism addressing economic considerations such as monetary cost of resources.

The works such as COMBINE [53] and AGORA [52] addresses the economic considerations while forming a collaborative group. AGORA uses an open marketplace environment where collaborating users buy and sell bandwidth.

COMBINE uses an incentive mechanism based on unified monetary and energy costs for collaboration. It uses an energy cognizant accounting scheme for maintaining the peer relationship. Even at the worst mobility scenario, these works reported a gain by a factor of two to there by exploiting network diversity. Our work, BuddyShare also exploits the network diversity by application-level stripping for sharing the Internet. It further maintains the peer relationship based on social relationships of the collaborating users. Since the Internet bandwidth is an expensive resource and if a user subscribes to unlimited data plans, it is likely that users may not be willing to share it with strangers. Hence, along with sharing bandwidth, our work answers with whom we share these resources, by forming a social based overlay group.

With respect to considering social relationships while forming an overlay network, there are some relevant works which incorporate social network aspects in P2P overlay formation. The work introduced in [55] aims to construct an overlay topology by mimicking social phenomena to reduce the search latency. It aims to reduce the number of overhead messages while querying for an interested object, which is expected to be found in socially closer peer. The peers are connected based on social relationships inferred using the access similarity in preferences. These inferred relationships are not friend or trust relationships intended to form overlay links in order to share the Internet resources. In [54] the authors consider friend relationships obtained from Orkut social networking portal. It constructs an overlay on top of a social network and shows sizeable improvements in lookups and round trip delays. This work evaluates the Orkut's suitability for overlay networks by studying the clustering behavior of its graph structure and socializing patterns of its members. We extend this idea of forming a social based overlay to the mobile world, and sharing the Internet connection, which has an additional constraint of physical presence of friends within the ad-hoc range.

BuddyShare builds on features of other prior works. To form overlay on short range ad hoc network, an adapted Gnutella for mobile environment in [46] is modified and used. An overlay link is established only when both the nodes agree on mutual trust computed from available social relationships information. As in [37], social trust is computed based on connectivity in social network. The social relationships are obtained in distributed manner by using local address book contacts or call details and further exchanging them with ad hoc neighbors [56].

CHAPTER 3

3. SOCIO-ADHOC NETWORK

Socio-Adhoc network is an overlay network formed over mobile ad-hoc network of cellular devices. Each of these devices runs an application level middleware called socio-adhoc BuddyShare to run the service. This socio-adhoc middleware discovers all the adhoc neighbors and all the physically reachable nodes from the initiator node. It is important to note that, this is an active discovery and only happens when the initiator starts the service. This middleware computes trust of the nodes, creates overlay group and displays the potential group members to the device owner for an optional secondary authentication. Since middleware constructs this overlay using social network analytics, by default, all the overlay members will be part of the overlay unless the owner decides to block any.

Once overlay is created, various services are offered by the middleware to the owners. In this work, we will focus on bandwidth accelerator service. It scales up the user's data rate. Initiator submits its current pending job to all the members of group. Job here refers to download job. The initiator of the overlay group is adjudged as the leader of the group. In most cases, it is the leader who requests the service. In case members need service, they create their own group, which does not contain other group leaders. Leader breaks up its job and redistribute to every group member. All the group member replies before the group expiration time, assigned by the leader based on heuristics. The completed job is then independently delivered by the members to the job owners. This process discussed is an informal discussion of the actions carried out by the middleware over a wireless PAN medium.

We have discussed some of the issues while constructing socio-adhoc overlay network in the first chapter. Now, let us discuss ways to address them by building this socio-adhoc middleware. In the middleware, all the communication that happens between the adhoc nodes follows a Socio-Adhoc (BuddyShare) communication protocol.

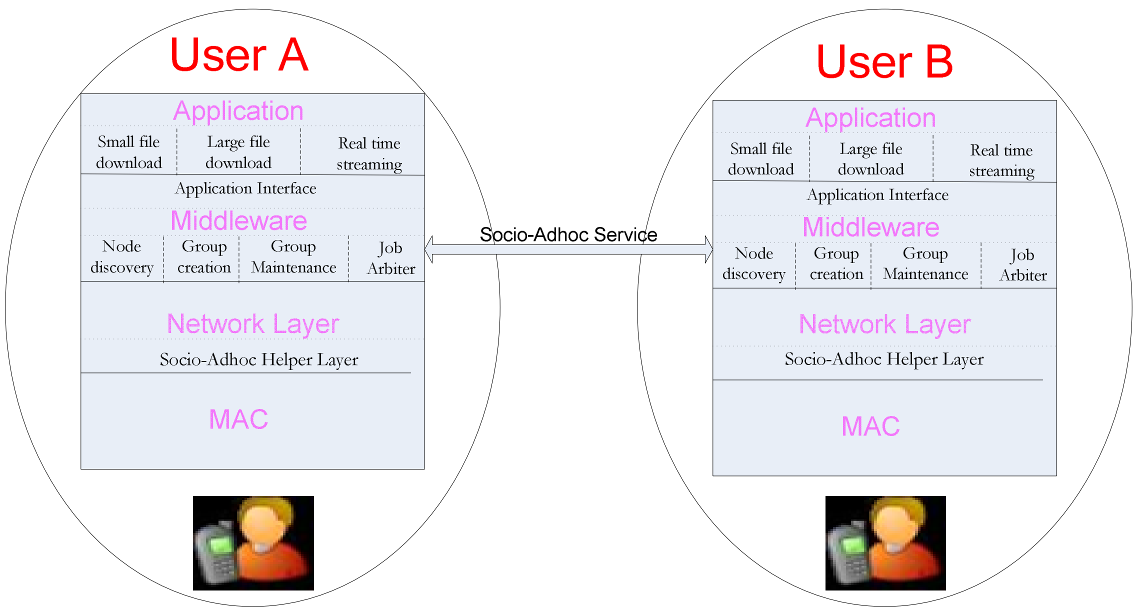

Figure 3: Protocol Stack

BuddyShare protocol stack is shown in Figure 3. It includes middleware for an application level overlay and it consists of four process states. The overlay construction is handled using two process states of a node: Peer discovery, and Group creation. Unlike large scale peer-to-peer networks, BuddyShare relies on connections with only a few physically reachable peers. Hence, peer discovery is similar to node discovery process inherent in ad-hoc network. A helper layer at network level gathers all the physically reachable nodes while constructing the adhoc network (like AODV, DSDV) and informs periodically (using cross-layer interaction) for overlay reorganization.

In Group creation process, the initiator shortlists peers from the physically reachable list based on satisfied wireless resources and social trust cost. A group creation protocol establishes the overlay links between the initiator and all its peers. This group forms a tree routed at the initiator. Only root can issue the download jobs for its leaf peers. So, a root peer is also termed as leader and leaf peers as slaves, where leader issues download jobs to slaves. A slave of one group can be leader in another group.

A job distribution mechanism is handled by a Job Arbiter process state. A job is defined as a download request for a file chunk. The jobs are created by packet stripping HTTP byte-range requests at application layer. The jobs contain segmented byte-range, expiration time, ID and other details. These jobs are distributed using input-output queuing mechanism.

An overlay maintenance mechanism is handled by the Group maintenance process state of a node. This state defines the protocols for nodes to join and leave. When a node discovers a new overlay, it can join as a leaf peer of the overlay tree. All the leaders update their neighbors list and pending jobs are redistributed. A slave loss is detected by the Job Arbiter, when the head of the line job does not get completed by the TTL value of the job. A slave loss causes the leader to reorganize its group and redistribute the jobs assigned for the lost node. A leader loss is detected while data delivery and all the subsequent jobs from the leader are not served.

All the protocols for overlay formation, maintenance and job distribution are stateful. There are several limitations in an ad- hoc network such as delays, losses and mobility, which affect the throughput in an ad-hoc network. However, this achievable throughput of an ad-hoc network is still not a bottleneck when compared to the individual capacity of a cellular link. Thus, even with overheads due to overlay formation, maintenance and job distribution, the vertical data connection can be efficiently shared to increase the net data rate.

3.1 Group Discovery: Node Discovery

The group discovery process requires discovering an ad-hoc network node and discovering an overlay peer. Ad hoc nodes are an autonomous router, which discover other nodes on-demand by the needs of the application layer requests. These nodes have built-in discovery procedures to construct routing tables. For example, AODV discovers on-demand and OLSR discovers proactively. A cross-layer interaction allows efficient peer discovery in the overlay network. An overlay peer exploits the routing protocol in use, to perform peer discovery in conjunction with route discovery at the network layer. A new type of message that allows node to broadcast optional information (OI), and enables discovery of all the sensed nodes. A cross layer interaction can be achieved by a vertical stack [46] component using publish-subscribe framework which enables interactions other than standard layer interfaces. An overlay peer injects its control messages along with routing control packets for discovery requests, or discovery replies. Using the shared data structures such as routing table or bridge table in case of Bluetooth, the middleware layer gathers all the peers with in reachable range along with their physical distance.

As found in other cross-layer approaches, two events are described to broadcast and receive messages. Messages contain peer info such as peer ID, leader ID, and group information. The helper routing agent subscribes, and receives notification from middleware layer. These events inform agent to advertise peer information along with route discovery packet used for routing. The receiving event is subscribed by the middleware. When routing agent receives messages pertaining to peer information, it notifies the middleware. These two events are supported by helper agent present in network layer which also notifies ad hoc network reorganization to middleware. It impacts peer joins, leaves and overlay reorganizations.

The helper agent maintains a list of all reachable nodes within certain number of hops. It also computes wireless-resource cost of nodes in the list depending on number of physical hops, signal strength, battery strength and observed round trip delay. It opportunistically updates this list when the routing agent gets any new information such as a new route or loss of an old route. The resource cost is a variable and may vary on different network conditions. Thus the helper agent conveys the snapshot of the network along with the resource cost periodically to the middleware. This cost is accounted for creation of efficient overlay group which is aware of constraints in the physical wireless network.

3.2 Constructing social overlay tree

In this section, we study the Group shortlist mechanism which uses trust and resource cost to shortlist overlay members. Once the shortlist is created, a Group creation protocol establishes the overlay links between these overlay members and the leader.

3.2.1 Group Shortlist

From a list of physically reachable nodes, a shortlist is created using trustworthy cost and resource cost. A resource cost (RCOST) is computed by taking a linear combination of number of hops, battery strength, signal strength and average observed Round Trip delay. A high resource cost is incurred for a furthest ad-hoc node with bad link or low power. A trustworthy cost is computed using the Compute-Trust (CT) algorithm. It uses socio-grams which contain "actors" as nodes and "relationships" as edges to compute trust.

A trust between two people is computed by using the direct trust probability and based on connectivity between two people in the social network. The direct trust probability T_{AB} between a pair A and B is computed using the following equation:

$$T_{AB} = (1 - \beta)\frac{m_{AB}}{m_{A,\max}} + (\beta)\frac{R_{AB}}{10}$$

R_{AB} is the explicit trust rating in a scale of 10 assigned by user A to B . For example www.orkut.com has three stars to rate trustworthiness of a friend. To counter the inaccuracy in rating assessment, the number of social messages exchanged is used as a measure of trust. The ratio $\frac{m_{AB}}{m_{A,\max}}$ indicates a relative number of messages exchanged between A and B to the maximum number of messages between A and any node. This factor intuitively indicates how close A and B is with respect to A 's closest person. β is the confidence in explicit trust rating.

The calculated trust probability from the equation is only based on direct interactions. Two persons may be involved in indirect trust. This is significant as the social networking sources are usually not complete. Our indirect trust calculation is formulated in similar line of Rounding Algorithm used in [37]. This algorithm assigns continuous {0, 1} trust values to different nodes in the network with respect to randomly chosen sources. In our case, we assign the direct trust probabilities computed from equation 1 to the nodes with respect to a random source. The trust ratings at the each edge are averaged and rounded bottom up from the sink to the source. A simple Breadth First Search (BFS) is used to calculate the trust index. The trust relationship factor between the source and sink is proven in [37] to converge to the actual social trust. We assume there are no misbehaving nodes.

We have validated that this way of calculating trust have good convergence for internet sharing application using data collected from survey. The details are part of future work. Once trust (CT) cost and resource cost (RCOST) is computed, a shortlist is chosen from the list of reachable nodes by using an aggregated cost between an initiator and node is computed using the following equation:

$$Cost(<init,node>) = \alpha * {}^{1}\!/_{RCOST(<init,node>)} + (1-\alpha) * CT(<init,node>)$$

α is a parameter that chooses between social trustworthiness and resource constraints. This parameter is set by the internet accessing application, for e.g. a secure email should set α as zero, whereas a video streaming should set α as one.

An important requirement of the trust computation is to have an access to the social network data. In case of BuddyShare network, the social network can be either collected locally from mobile devices in distributed fashion, or from a centralized repository.

3.2.1.1 Centralized Approach

In centralized approach, Compute-Trust algorithm is best to be implemented in centralized server. It has access to social network and can save processing computations on mobile devices. The mobile devices provide the list of ad hoc neighbors list to server. The server computes the trust using equation 1 by providing values for R_{AB}, and ratio $\frac{m_{AB}}{m_{A,\max}}$. The values are obtained from various social networking data. For e.g. in case of social networking server, ratings can be explicit user ratings and ratio $\frac{m_{AB}}{m_{A,\max}}$ can be heuristically determined by collecting the total number of personal or public interactions shared between two. See Figure 4 for the centralized architecture.

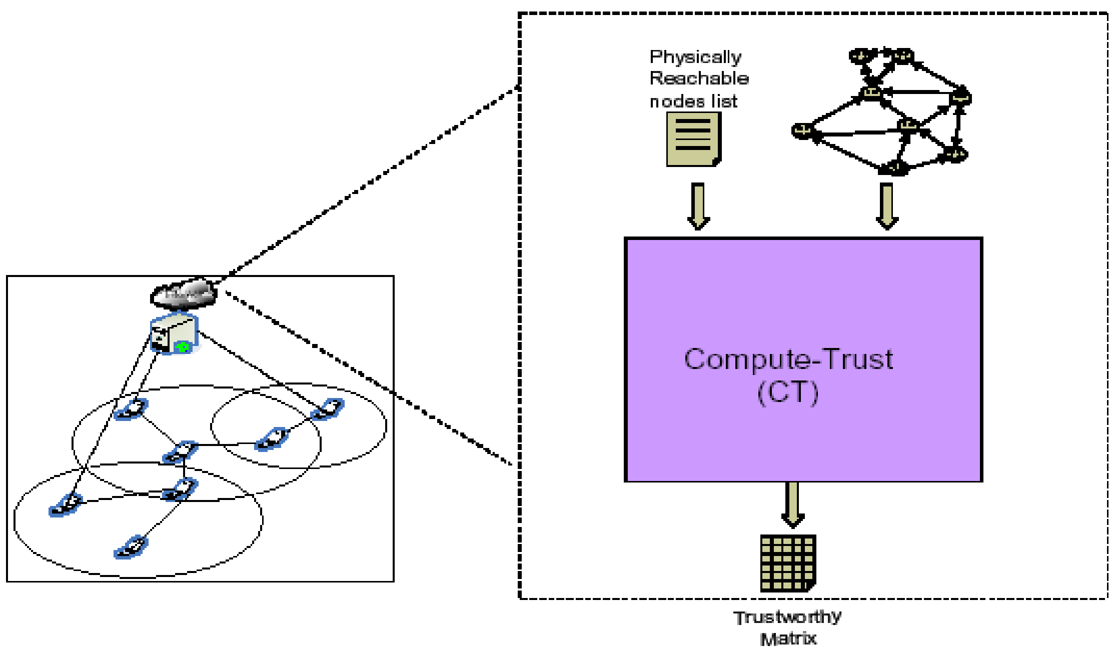

Figure 4: Centralized trust computation

3.2.1.2 Distributed Approach

Distributed approach makes a local decision for social trustworthiness. It uses address book based approach [20] or past interaction based approach. Address book based approach requires user intervention to rate the trustworthiness of each of the person in the address list. This trustworthiness rating is assumed as final. It can also use other local factors such as number of calls made to the particular addressee, number of missed call records of addressee, and number of call records of addressee present in received calls. Alternatively, the devices can exchange the address book while trying to pair up during group creation phase. The number of common neighbors could be used as another factor for the trust rating. This is direct trust assignment of only those are present in address book, which may cause to miss on several potential trustworthy nodes that are physically reachable. The exact trustworthy cost function follows as similar formulation as cost used in centralized case, which has complete social network information. Equation 1 uses m_{AB} as the number of messages exchanged in social network, in here this number is a linear combination of factors such as dialed calls, received calls, and missed calls. The absolute value of direct trust rating by the user in address book is correlated to the value of R_{AB}. If such rating system is not supported by the address book, relative number of common neighbors correlates to a (0,1) value of R_{AB}.

An alternative distributed approach is to use past interactions between the devices. An example interaction could be Bluetooth interaction. If a device has been found regularly in an interaction with another device, it could correlate to some kind of social relationship between the two owners. Intuitively, frequent physical interactions give a broad correlation between the physically closer people and the actual social network. Some sociological works [38] have supported the intuitions. Thus, a node can compute the trust rating based on the frequency of interaction. Every time a node interacts with another, a plugin layer records the interaction and exchanges the past interaction records with the neighbors. Each node can construct a two hop social network with respect to the initiator and use compute trust function on this graph to infer trust between the physically reachable nodes. The cost function computed in this case is similar to that computed in centralized approach using complete social network.

Work in [35] uses locally computed social networking features such as betweenness and node similarity to see a good correlation between the centrally computed betweenness and node similarity. It uses these features to judge which next hop neighbor is the best suited carrier to route the packet to destination. Nodes may exchange the nodes they encountered previously to compute these features. In the context of our work, such local computation augments the computation of trustworthy rating. The egocentric betweenness and other measures can replace as ratings and thus could be used in overlay creation.

3.2.2 Group Creation

An overlay group is created by the initiator with short listed nodes. A group is collection of nodes which are socially trustworthy with respect to the initiator. The lists of steps for group creation are:

- A Group-Creation message is sent to all the nodes in the short list. It contains Group ID, same as node ID of the leader, assigned peer address, and group expiration time.
- On receiving Group-Creation message, the nodes compute their estimated aggregated cost. On a satisfied cost and availability, the nodes send Group-Accept message. The nodes maintain the state of the group until the group expires.
- On receiving Group-Accept message, the leader updates its neighbor list.

We do not discuss security issues such as replay attacks, or spoofing in this work. We assume no misbehaving nodes. The messages are sent on sockets based on TCP connection between the nodes. Hence there is no message loss or inconsistency in the states. The stale states are removed by an expiration timer set by the node. Next we discuss several features of the group maintenance such as joining, leaving or reorganization of a group.

3.3 Group Maintenance

A peer node discovery is carried out in conjunction with the ad-hoc discovery process. A peer can join any other group. The role of the peer is to either serve as a source or a requestor. A peer cannot join another group as a *leader*. A *leader* has to create its own group. A peer can however join as a source or *slave*. The steps are: A *slave* peer multicasts Group-Discover message to all nodes in the shortlist. If the receiving peer is another *slave* peer, it forwards it to its leader. The receiving leader then initiates a group creation process with the newly discovered node.

Leaving group: A node may leave the group under various circumstances. It may leave by explicitly notifying the leader of the group. A Group-Terminate message is sent by the leaving node to the leader. The leader removes the state and reorganizes its overlay neighbors. It also redistributes the jobs meant for the node to other remaining nodes. The nodes can also suddenly vanish from the overlay network due to the unpredictable characteristics of the network. The leader maintains a list of active jobs at the overlay neighbors for a download. The job expiration time at the leader, notifies a neighbor loss. The node after a loss may later reenter and reply with the finished job. If the finished job is in the valid interval of time, the loss of the node will not be detected as the node as served its group correctly. If the node realizes that it has unfinished jobs whose finishing time has gone past, the node clears all its state and joins the overlay as a new peer.

The overlay reorganization takes place whenever a helper layer detects a change in the physical network. Several overlay adaptations [46, 53] for physical network change have been proposed in the past. Usually, the overlay links adapt to the physically closer node. In our case, the overlay reorganizes based on the aggregated cost which is periodically computed based on the updates provided by the helper layer. If the cost of an existing neighbor reduces below the previously computed cost, no additional jobs are assigned.

3.4 Job Arbiter

The *job arbiter* mimics the concept of input-output queuing used in routers. Instead of implementing the exact Input-Output queue and applying the bipartite matching algorithm to match the input with output queue, we slightly modify the design to schedule or reschedule the jobs to the overlay nodes. Every node maintains an input queue for each *leader* and an output queue for each of its *slave*. It also has an extra output queue for jobs to its 3G interface. A job encapsulates Http byte-range request, job ID, its leader ID, expiration time depending on the type of request and the master job ID. A download initiating node segments the request into equal sized blocks and distributes to all its output queues. The extra output queue corresponding to 3G interface schedules the job for download. The other output queues sends it to the input queue of the corresponding neighbor. On completion of a job, a neighbor replies the downloaded data opportunistically to the original requestor of the job. In special cases like multicast, a data delivery tree is constructed to deliver to all the required leaders.

The node also sends the Job-Complete message to its immediate leader. The Job-Complete message is propagated all the way back to the job owner in the same path.

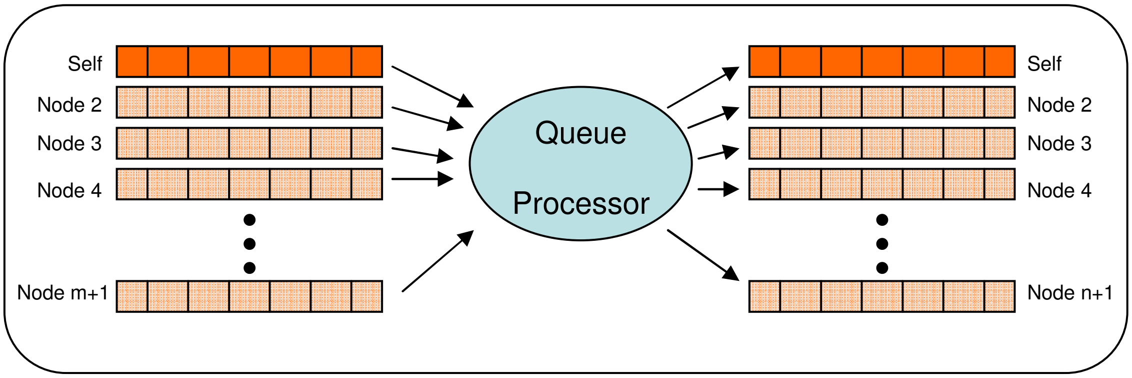

Figure 5: Job Arbiter

At the leader, the Job-Complete receive action dequeues the job and the next job in output queue is scheduled. If an output queue becomes empty, the jobs are dequeued from the input queues in round-robin fashion and refilled in the empty output queue. When there is no job in input queue, the jobs are removed from other longest output queue. Thus no job remains blocked in output queue due to a loss of node. Even when all the neighbors are down, the output queue leading to 3G interface will be downloading and completing the jobs. If a node is lost, the corresponding output queue times out and the jobs are rescheduled to other queues. The jobs are sorted based on expiration time and stored in the queues in same order. The expiration time of a job depends on the type of request. A streaming request has a short expiration whereas a file download has higher expiration time.

All the protocols are stateful in nature. Since the overlay is expected to be small in nature, the statefulness of the protocol does not cause any issues such as scalability.

CHAPTER 4

DATA COLLECTION

Much of the descriptive and analytic usefulness of social network analysis is based on the unique relational data collected in various formats, typically through survey instruments. In their simplest form, these tools typically ask respondents about their ties to a listing of certain individuals, organizations, or groups. Relational data collected from these instruments is usually converted into a matrix or list format, and used as input for any of the several existing software packages designed specifically for analyzing social network data.

The biggest challenge of this application is to make the social networking data of a person available to the socio-adhoc service. We have collected social network data in following ways:

- Address book based social network inference
- Mining the Call Detail Records (CDRs)[12]
- Social networking portals [40]
- Direct questionnaires
- Mobile device interaction logs [9]
- Randomly generated small world network that obeys certain properties.[39][41]

In our distributed approach, we use the address book as the source of social network. Every contact present in address book is a one hop neighbor in the social network. As discussed previously, trust information is derived from the local information such as call records and common neighbors. Since it is difficult to infer real “social” relationship from the address books we also follow a centralized approach to obtain the social network data.

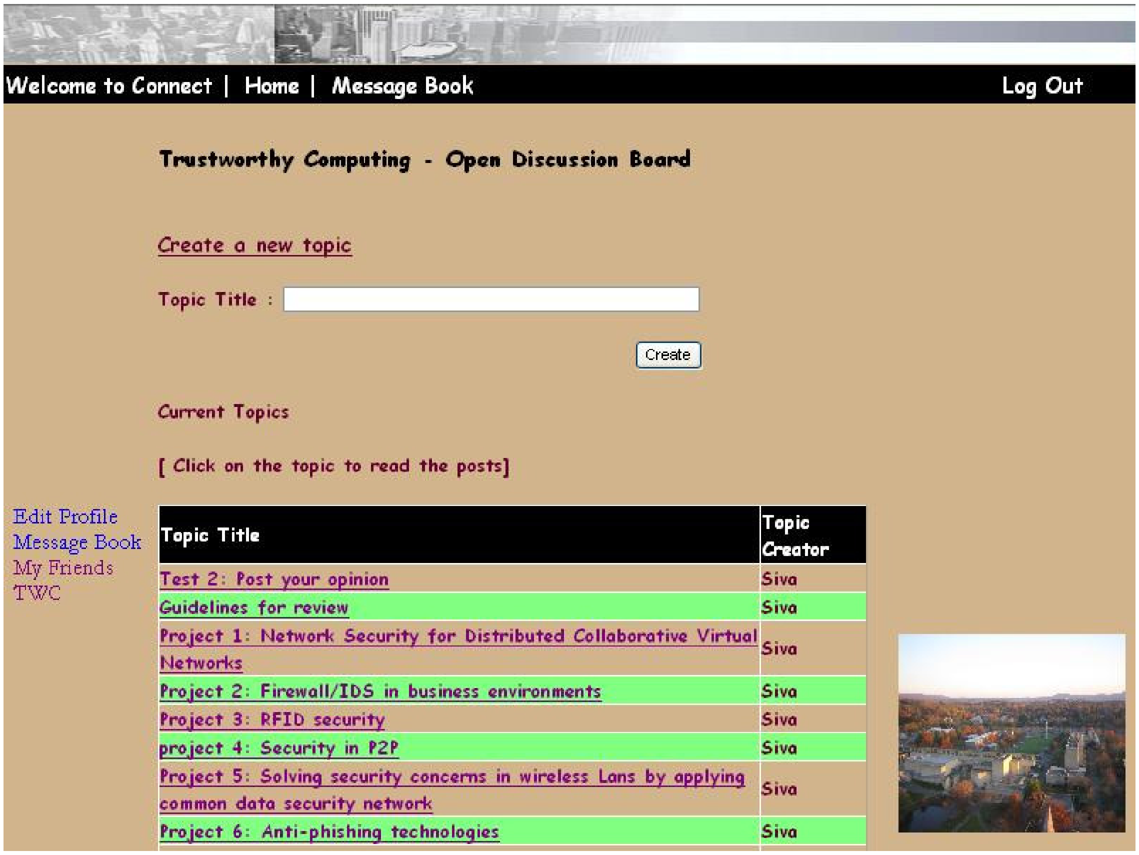

Figure 6: Connect: Social Networking Portal

In cellular network, base station is useful resource to extract social network. [12] shows how to extract the social network from the call graphs. These graphs are generated from the calls made by one to another. A social network graph is created by adding an edge between the caller and callee. The weight could be computed from frequency and duration of call length extracted from the CDRs. These call graphs were shown to obey all the properties of a social network and we have already discussed how to compute trust from such centralized social network resource.

We have built a social networking portal called connect (See Figure 6) [40] for students of UMASS. Connect is an online discussion forum for courses such as network security, wireless networks, and advanced wireless networks. Students participate in connect to share the knowledge, review the peer projects to avail extra credits. Started few months ago, connect has already tasted its success among students community of UMASS. Connect has features to add friends, add communities, discuss topics in forums. Trustworthy rating is computed from metrics such as personal messages, and common friends. While collecting the data, we have used anonymization techniques [28] to conceal the identity of the actual user. More data can be collected by collaborated with other social network portals.

Direct questionnaires are the best way to get an individual perspective of several individuals. Questionnaires such as "friends", "best of friends", time spent with friends, willingness to share internet connection, duration for which friends are physically closer, etc give a better intuitive idea of evaluating our work as well as to deduce the accurate social network useful for our work. These questionnaires were given to the students belonging to different age groups, cultures to get a wide variety of opinions. The students were never asked to use the names, instead a random number representing the names are used. An undergoing work[42] uses questionnaires to find both physical proximity and social proximity and uses it to evaluate the correctness of self reported data. Our work uses this as an analysis mechanism to study the usefulness.

Some of the questionnaires as mention in their work were: Dyadic Questions: *Estimate Your Average Proximity with Each Person* Rate as: 5 - at least 4-8 hours per day, 4 -at least 2-4 hours per day, 3 - at least 2 hrs - 30 minutes per day, 2 - at least 10 – 30 minutes per day, 1 - at least 5 minutes, 0 – 0-5 minutes (default) *Is this Person a Part of Your Close Circle of Friends?* Yes / No (default)
Some of the questionnaires used in our work were: *Quality time per day you spend with people you listed as friends* a) > 7 hours b) 4-6 hours c) 1-4 hours d) <30 minutes *Physical space you share with people you listed as friends* a) <15 feet radii (room) b) 20-50 feet radii (class) c) Don't meet often

An alternative way of deducing social networks is using device logs of mobile users. Reality mining experiments [9] conducted in MIT uses 94 subjects using their mobile phones preinstalled with several pieces of context software that record and send the researcher data on call logs. Bluetooth devices in proximity, cell tower IDs, application usage, and phone status. These subjects were observed via mobile phones over the course of nine months, representing 330 thousand hours of data. Subjects included students and faculty from a major research institution. A Bluetooth device conducts a discovery scan and others within the range respond. A persons MAC address is logged in as discovered person nearby. The interaction log generated over a period of time serves as an input to construct interaction network. Several short interactions occurring only few times do not qualify as a physical interaction. Also the interaction network cannot exactly specify the exact relationships in social network. However, these logs could be used for understanding some of the social behavior of the subjects. These data were analyzed to obtain the social networks as presented in [42]. Our distributed approach uses the similar ways to record logs and infer the social network. We will later see the data used in reality mining experiments were used to infer some useful patterns that support the successes of the socio-adhoc network.

Finally, the social networks are proved to obey certain properties such as power law distribution of degree [14], or distribution of path lengths [16]. Watts in [6] showed that these properties match with small world networks. These small world networks follow two important properties: high clustering coefficient, and logarithmic distribution of path length with the number of nodes. These two properties characterize random graph generator as shown by [39]. These generators can be used to generate social network and by assigning random trust assignment similar to [37], one can compute the trustworthy rating to test some of the hypothesis.

CHAPTER 5

MOTIVATIONAL STUDY: SOCIAL BEHAVIOR ANALYSIS

In this chapter, we will discuss some preliminary results to support the use of our system. Specifically, we will analyse some of the interaction logs obtained from reality mining experiment dataset [9] and social networks from different sources [40]. Our main aim is to study certain features of social network that support trustworthiness among mobile users to form a collaborative overlay network. Let us first discuss certain features of social network that are significant to form overlay network:

Clustering Coefficient: Clustering coefficient [39] of an actor in social network is defined as the measure of connectivity among neighbors. It quantifies how close the vertex and its neighbors are from being a clique. It is computed as ratio of total number of edges among the neighbors to maximum possible edges among them. It also is alternatively computed as ratio of total number of triangles involving the subject node to maximum possible triangles involving the subject node.

The clustering coefficient of graph is average of all the clustering coefficients of graph. In our work, clustering coefficient of the social network gives a hint of connectivity of the network. If a node has high clustering coefficient, its neighbors are well connected thus even if few social network neighbors are found near the vicinity, it is enough to form a well connected overlay network. This is because, when the initiator's neighbors are well socially connected, even if one of them is found near the vicinity the neighbor's overlay network, this neighbor can lead the initiator to more common neighbors in its vicinity as they are well connected among themselves. As opposed to this, if the clustering coefficient is less, then a node has only option of creating many overlay edges to reach its socially close neighbors. On studying the general distribution of clustering coefficient of a graph and finding the mean of graph, a system can decide whether to choose distributed approach or centralized approach to compute trustworthiness. As discussed earlier, distributed approach do not make expensive query to centralized social networking server, instead it computes the group without any centralized server. If a graph has high clustering coefficient, it is better to use distributed approach as overlay connectivity may not significantly change in centralized approach.

Importance: Importance of a node can be defined as how important a node is to the network. It is subjective to the application, and can be estimated collectively with several factors such as PageRank [43], hubs and authorities [15], Indegree, centrality measures etc. All these features conclude how important a node is in the social network or how well connected a node is in the social network network. On observing the pattern of distribution of these measures in dataset, the system can identify bottleneck nodes a priori. Most of the small world networks follow power law [16, 14] in these distributions. The power law distribution implies that there are fewer important nodes and larger unimportant nodes. However, in some of the dataset that considers only small social networks such as a college students' social network, the distributions was not exactly power law supporting our cause as there were almost equal important nodes as unimportant nodes.

Proximity: Our work strongly believes in hypothesis that socially closer people are often physically closer as well. Thus it is important to survey or analyze the distribution of the degree of physical proximity in a network. An interaction network as discussed early can be the best resource for proximity information. Proximity has been studied in dyadic questionnaires and has been used as an interaction network in an undergoing work [42]. In the context of our work, proximity is an important feature to study because if friends are often found in physical proximity, then the overlay network is useful in increasing the bandwidth of a person's data connection.

As discussed in previous chapter, we have collected data sets from disparate sources such as social networking portal, questionnaires, call graph and Bluetooth interactions. We have observed the discussed features by conducting Social Network Analytic experiments on the following four datasets:

1) ***Questionnaires based Grad Graph of Fall semester***
2) ***Social networking portal connect based Grad Graph of Spring semester***
3) ***Bluetooth interaction Graph from reality mining project***
4) ***Call graph observed by mobile phones used in reality mining project. (under progress)***

5.1 Questionnaires based graph

This graph was generated from the dataset collected through questionnaires. The graph was visualized and analyzed using social network analysis tool called PAJEK[9]. It shows a dense connectivity as shown in the figure 7. The list of metrics obtained through the social network analysis tool PAJEK are :

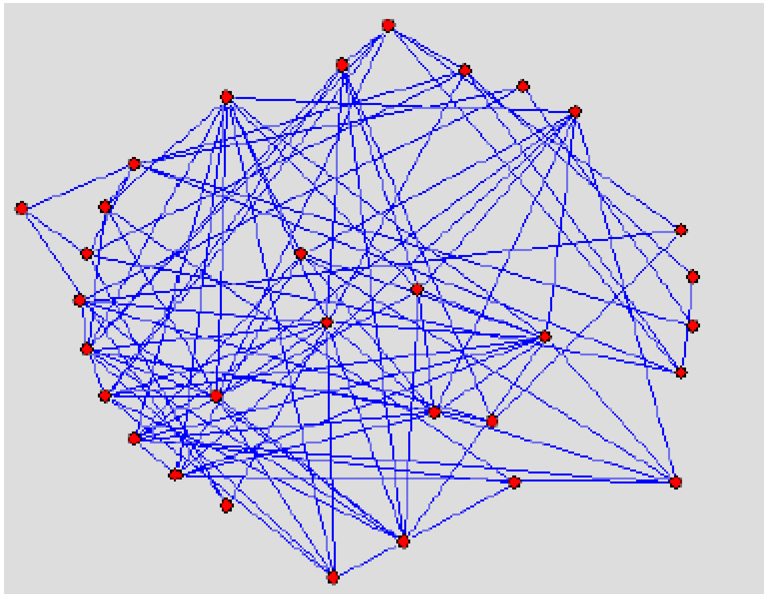

Figure 7: Questionnaire based graph

5.1.1 Clustering coefficient distribution:

. The distribution of clustering coefficients versus number of nodes is shown in the Fig 8.

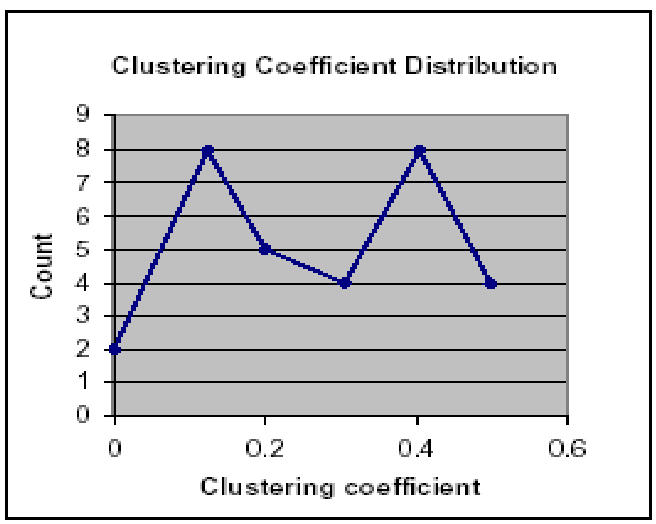

Figure 8: Clustering coefficient distribution of questionnaire based graph

High clustering coefficient implies high connectivity among the neighbors and high probability of reaching more neighbors via few physically close neighbors. This forms a platform for deciding between the distributed approach and centralized approach. Any initiator node first knows its "social" quality by means of querying the social network server or by past experiences. So one estimates its clustering coefficient and decides between centralized and distributed. For e.g. In our scenario of Grad students, *if threshold is set as 0.4 then 12 members will follow distributed approach and rest 19 members will choose centralized approach*, our objective is to minimize the number of nodes that uses centralized approach still guaranteeing a good connectivity .

5.1.2. Importance and other features

The non power law distribution of authority weights of small sized questionnaire based graph is showing in Figure 9.

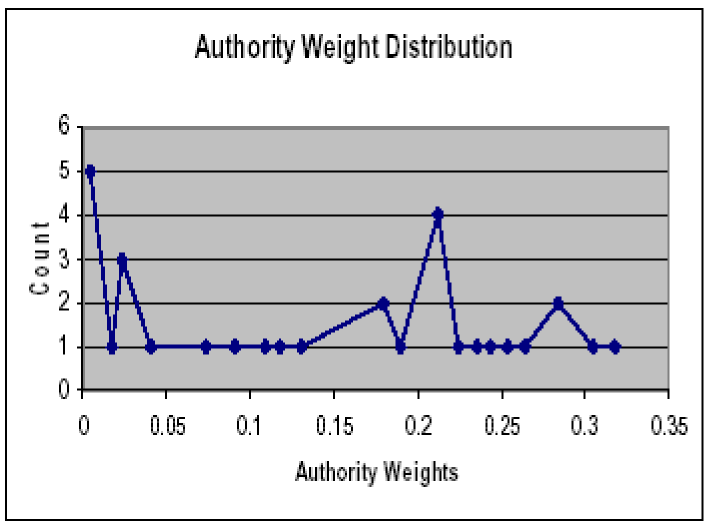

Figure 9: Authority weight distribution

Authority weight defines the importance of a node in the context of group of nodes that connect to a node. The most popular nodes in the network will attract more nodes to join the network and thus increase the connectivity. For bandwidth hungry applications, the connectivity with in few hops is an important factor to increase the throughput**.** The higher Authority weight peer is Bottleneck peer, which has high requests. The general distribution should be high. ***More than 12 students tend to have authority weight greater than 0.2, higher bottleneck nodes.***

On similar notes, high degree of the nodes also describes popularity of nodes. The degree distribution of the social graph is shown in Figure 10.1. We have also estimated the path length distribution shown in Figure 10.2 to estimate the quality of our application among the group of students. It is clearly visible that *around 50% of possible pairs are closer than 2 hops away in a social network graph, thus these people form a closed user group and we found from the survey under given conditions 80% of the students were willing to share the connection.*

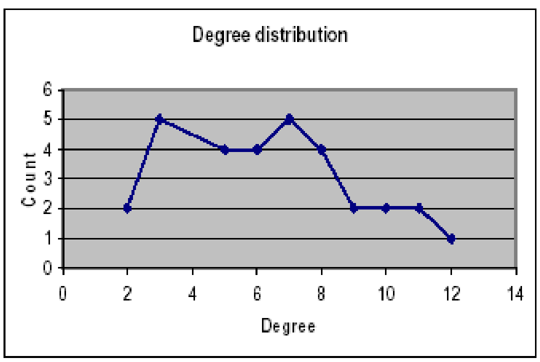

Figure 10.1: Degree distribution

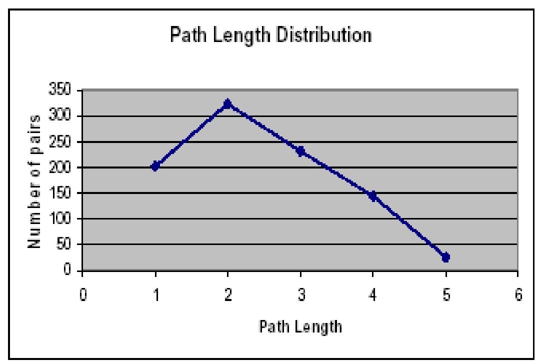

Figure 10.2: Path length distribution

Path length distribution indicates the optimal path length to judge the social trustworthiness. Smaller the average path length, the general trust factor of the network is high.

5.2 Connect based social network graph

This is graph generated from social networking portal Connect. The data is first anonymized and then using tools such as PAJEK, the features were analysed.

5.2.1. Clustering coefficient distribution

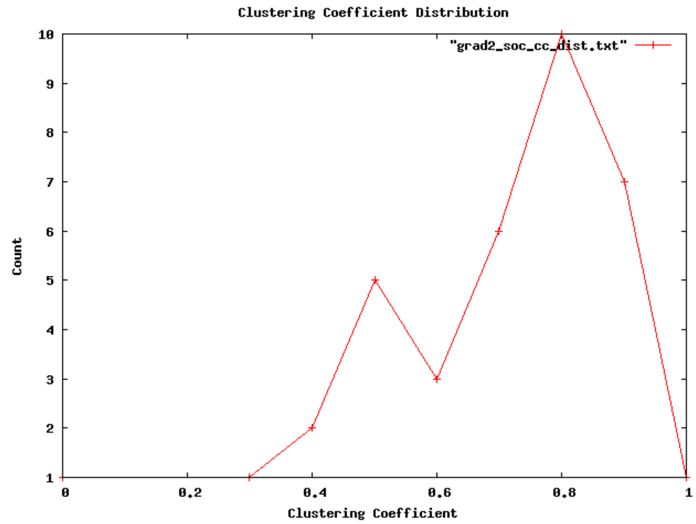

Figure 11: CC distribution of connect graph

If the threshold is set as 0.8, then 18 members will follow distributed approach and rest 17 will follow centralized approach. Like in previous case if threshold is set as 0.4, we have all the nodes following distributed approach. All the nodes are well connected among themselves.

5.2.2. Importance and other features related to path

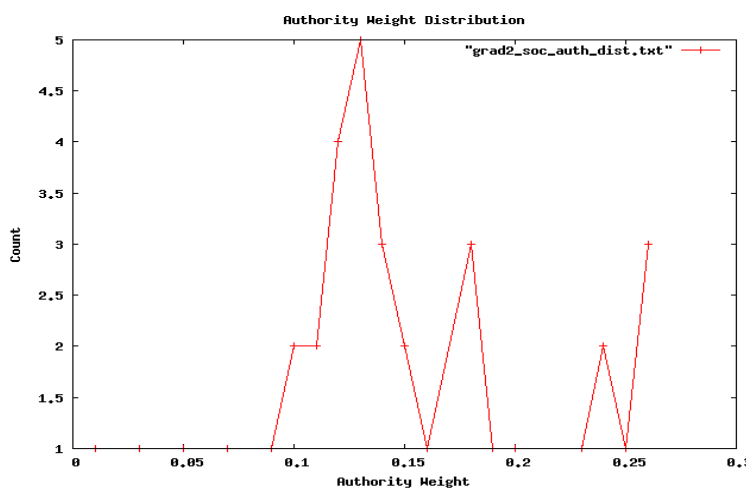

Figure 12: Authority weight distribution of connect graph

The distribution of the authority weight of the connect graph is shown in Fig. 12. *Around 50% of people have high authority weights, thus the number of bottleneck peers* *are high.* The degree which marks the connectivity of a single node appears to follow much of a normal distribution than a power law distribution as in case of small world social networks [39].

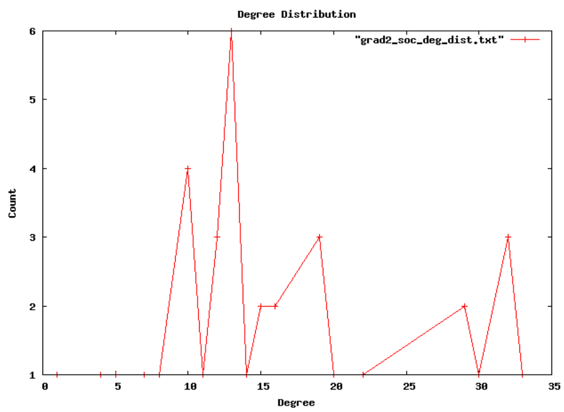

Figure 13.1: Degree Distribution

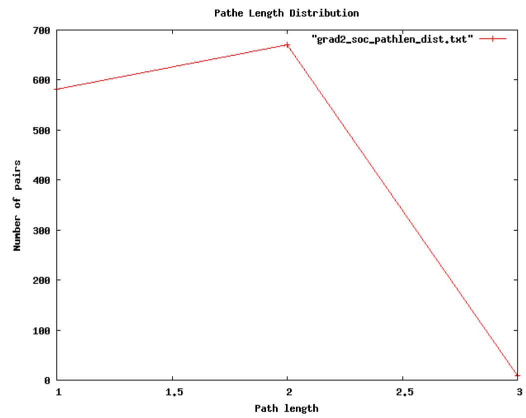

Figure 13.2 Path length distribution

Higher degree implies large number of requests. Hence, number of high degree nodes should be high in the network. And around 50% of possible pairs are closer than 1 hop away in a social network graph thus forming a closed user group. And almost every one is within 1 hop away.

5.3 Evaluating social factors on Bluetooth interaction network

In the previous chapter, we discussed about the reality mining project that has specialized mobile device to record interactions. Whenever a device comes under the vicinity of other device, it stores the MAC address of the device and all the recognized base stations. The base stations are optionally labeled by the user as “home”, “office” etc. These labels are used to obtain the context and location of the user interactions. We mined this dataset for conducting various experiments not just limited to social networking features. We focused on extracting the distribution of proximity and the size of average simultaneous interactions in order to verify the possibility of a useful overlay network. We also carried out the same set of social networking analysis to this interaction network. As suggested before, by intuition an interaction network is likely to correlate to social network as supported by some of the sociological works[38]. If two members share same physical space for a long time at non-office hours and weekends, it correlates to social relationship at more than 90% of time as shown in [c8]

5.3.1. Clustering coefficient

Similar to other networks, we first study the clustering coefficient distribution which is shown in figure 14.

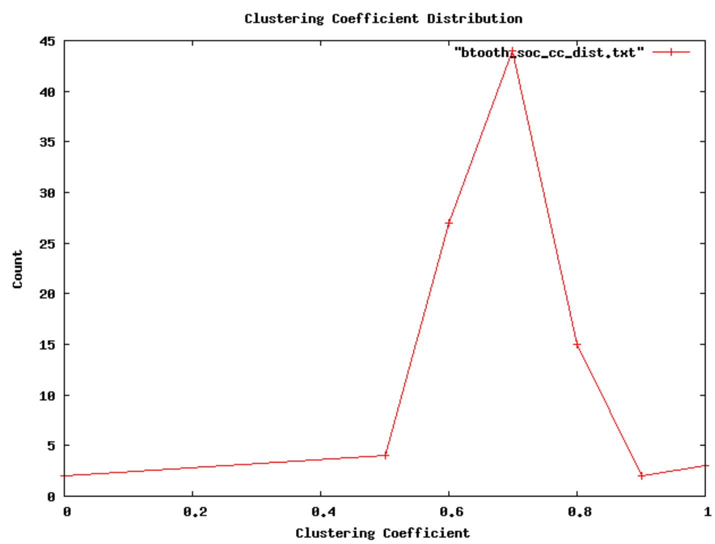

Figure 14: CC Distribution of social network based on physical interaction

If the threshold is set as 0.7, then 63 members will follow distributed approach and rest 34 will follow centralized approach. The distribution is normal as opposed to power law in the existing small. This network is of special interest as most of the nodes have high clustering coefficient. This network entirely depends on the set of people chosen to participate. In some sense, all these people chosen already fall under same umbrella as they have decided to use the specially designed mobile phone. Thus the physical interactions among these set of people is expected to be high, which is evident from clustering coefficient distribution.

5.3.2. Importance and other path related features

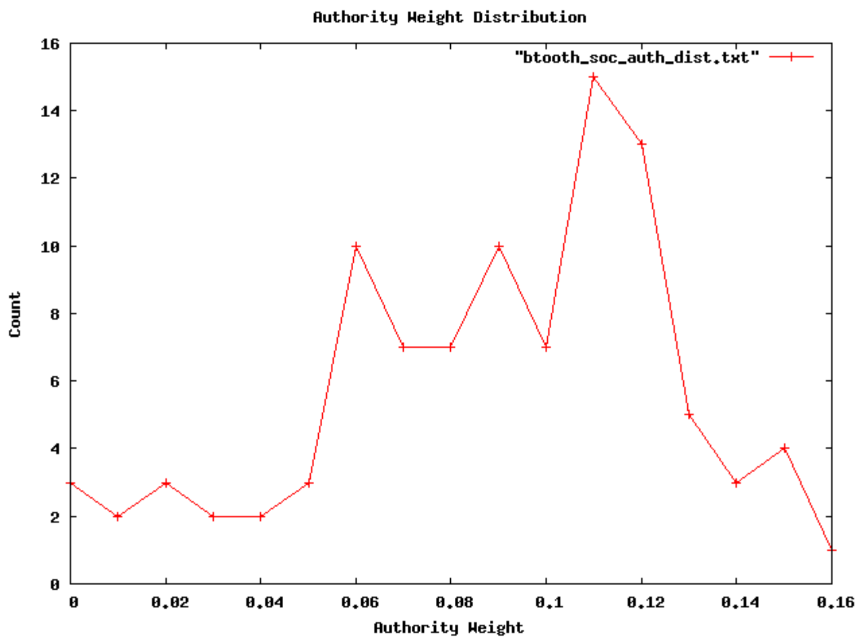

Figure 15: Authority weight distribution

Around 50% of people have high authority weights, thus the number of bottleneck peers are high. The authority weights are high because of the original dense connectivity of the graph.

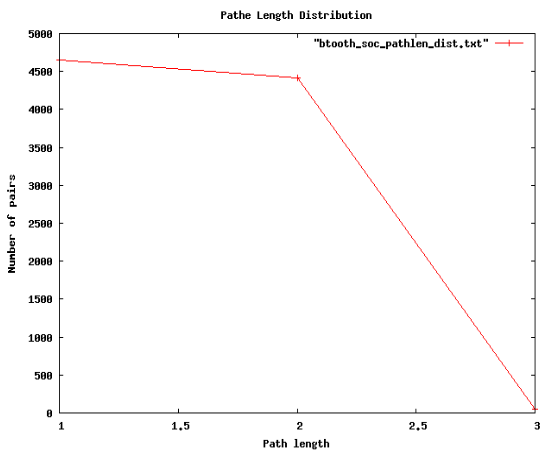

Figure 16. 1: Path length distribution

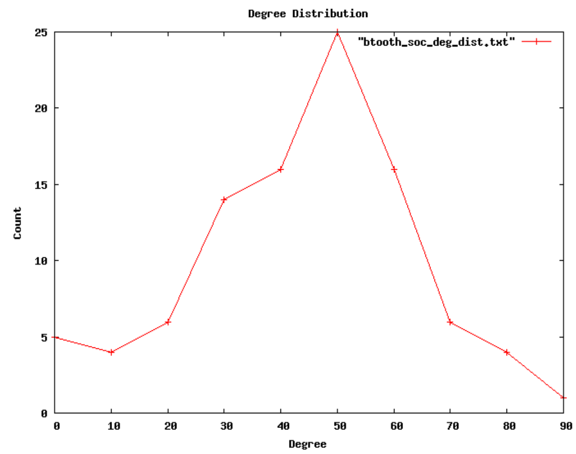

Figure 16.2 Degree distribution

\Around 50% of possible pairs are closer than 1 hop away in a social network graph thus forming a closed user group. And almost every one is within 1 hop away.

The other factors computed from these graphs are:

- *Number of unreachable pairs: 192*
- *Average distance among reachable pairs:* ***1.49496****, this validates that this interaction network is a dense network and correlates to the social network graph obtained from connect site.*
- *The most distant vertices: v1 (1) and v52 (52). Distance is 3.*

CHAPTER 6

MOTIVATIONAL STUDY: ANALYZING REAL USER'S BEHAVIOR

BuddyShare aims to increase the Internet bandwidth for data services by using nearby friends' mobile phones. This system viability depends on the following factors: how often friends come together and for how long they stay together. To understand the behavior of real mobile users in terms of their geographical proximity and social relationship with each other, we analyze the logs of mobile phone users obtained by the Reality project [9] conducted at MIT.

The Reality project consisted of ninety four Nokia 6600 smart phones pre-installed with context software. The users included faculty and students at MIT from different fields. The context software monitors the user behavior and updates a centralized server every day. The information collected includes call logs, Bluetooth devices in proximity, cell tower IDs, application usage, and phone status (such as charging and idle). The study generated data collected over the course of nine months and represent approximately 330,000 hours of data on users' location, communication and device usage behavior.

The Bluetooth logs of dataset are useful for our work since they include various interesting behavior such as: frequency by which people come close and how long they stay together. A lengthier interaction implies lengthier ad-hoc link duration and lesser overlay re-organization. The duration of ad-hoc links is an important factor for a stable overlay network and efficient data exchange. The interaction time distribution is depicted in Figure 17. An average interaction lasts 56 minutes, which is sufficient link duration for data exchange and countering frequent overlay re-organization due to the mobility factor prevalent in an ad-hoc environment.

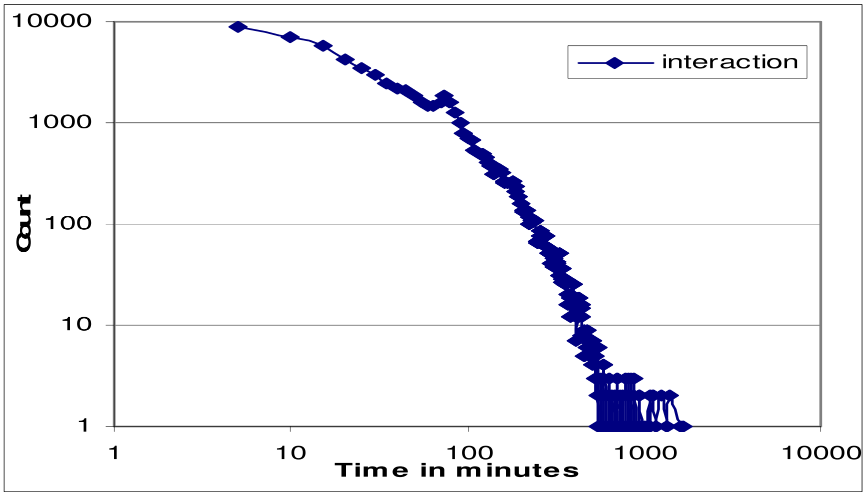

Figure 17: Interaction time distribution (Mean 56 mins)

BuddyShare requires the presence of social relationships among the sample users to form an overlay network. Since a mobile device only knows about the calls made, received or devices sensed, it can only use this information for computing the social trust. A frequent voice call between two people can be correlated to a social relationship. Hence, the call detail records in these datasets are very useful in extracting social relationships of the mobile users. We create a call graph where people are represented by "nodes" and sufficient voice call minutes between them are represented by "edges". We have two simple variations of this call graph to get the social trust relationship.

- *Strongly-social 1-hop call-graph* based: We have a threshold on the product of number of calls and the duration of a call. Every voice call with a target node which crosses this threshold is qualified as trustworthy node.

- *Relaxed-social 2-hop call-graph* based: The sample set just contains 94 people. This people could be socially connected through people not in the sample (For e.g. Friend of a friend). Hence we included those persons not in sample set but involved in voice call with at least two people from sample. The node representing this person introduces new paths between nodes in *1-hop call-graph*. In future sections, we show that extra paths can affect the trust calculation. This graph is prefixed *relaxed-social* as it also includes the indirect inferences.

We used the Reality project dataset to find out the "potential" overlays. The social relationships are available from the two described call graphs and proximity behavior is available from the Bluetooth logs. By studying the features of users who shared physical space within two physical hops for a long time, we constructed the overlay links virtually on satisfied social trust. For evaluation purposes, if two users have an edge in *1-hop* or *2-hop call graph*, we qualified them to have sufficient social trust.

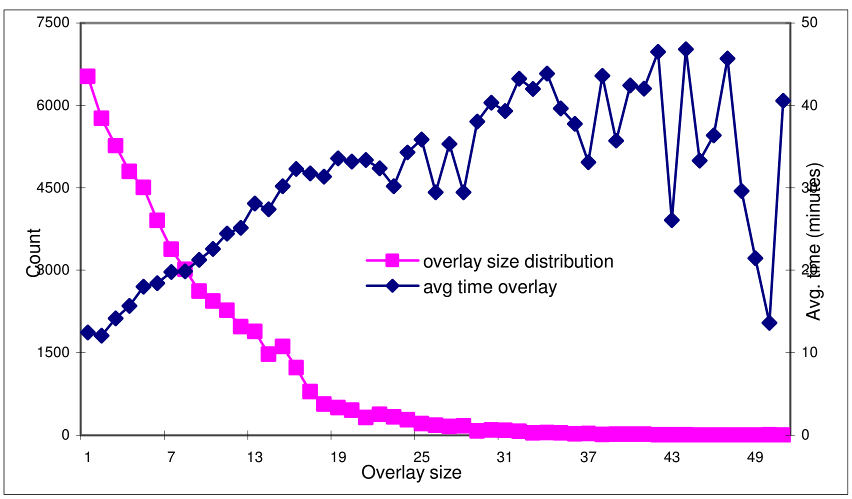

Fig 18: Overlay size distribution (mean overlay size: 3.5) and average overlay duration

We obtained the virtual BuddyShare overlay sizes and existence times at different scenarios for a period of a year. Figure 18 shows the distribution of the overlay size in primary y axis and distribution of average existence time of the overlay in the secondary y axis. For example, an overlay of size 19 could be found in 300 occasions over a period of a year with an average existence time of 32 minutes. This graph concludes significant existence time of overlays with a mean overlay size of 3.5 which provides us a good motivation to study the technologies to achieve the BuddyShare system. In the next chapter, we will simulate the BuddyShare system using the same dataset and show its advantages.

CHAPTER 7

SIMULATION

Even though a real test bed is more accurate than a simulation, the cost factor, scalability and versatility issues have encouraged us to build an ns2-based simulator. Our simulator (see Figure 19) is designed to closely approximate the real mobile phone with dual network interfaces: UMTS and Bluetooth. There are two separate stacks for overlay communication, and Http Communication as shown in Figure 19. We classify every component related to forming an overlay over Bluetooth ad hoc network into one stack. The other stack contains all components for client-server communication to a HTTP server using cellular network.

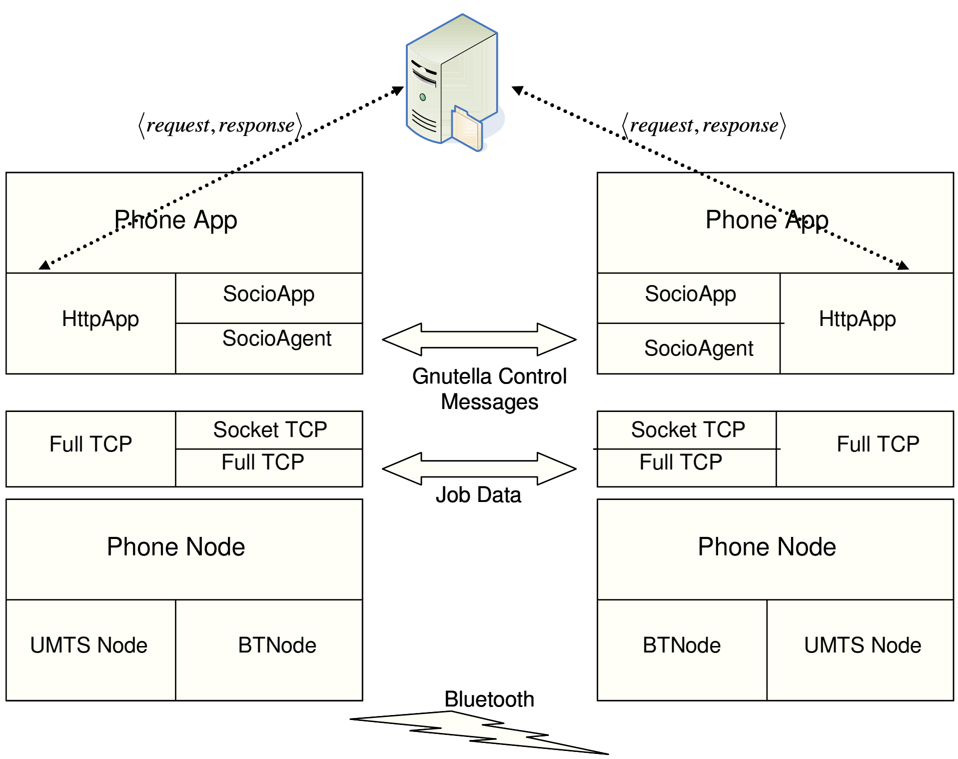

Figure 19 : Simulation Architecture

7.1 Nodes, Links and Lower Layer components

We introduce a phone node that contains network stacks of both UMTS (Universal Mobile Telecommunications System) and Bluetooth. The phone node is a wrapper for UMTS node from EURANE (Enhanced UMTS Radio Access Network Extensions for NS-2) [29] and Bluetooth node from UCBT [30] to present a unified interface to the simulator application.

A UMTS node in EURANE can be of three types namely Radio Network controllers (RNC), Base Station (BS) and User Equipment (UE). Each UMTS node contains several network interface (NIF) stacks, major components being Radio Link Controller (RLC), MAC and physical objects. The transport Channels such as (Dedicated Channel) DCH are connected to the physical objects. We use the UE node as one of the two interfaces of the phone node. We also use the RNC and BS node for forming a cellular network with the mobile phone for voice and data services. In all the simulation scenarios, an HTTP server application is attached to a NS node. This ns-2 node connects to UMTS network through gateway nodes GGSN and NGSN of core network using wired links. The gateway nodes, in turn connect to RNC node using wired links. The RNC connects to base station node via an Iub interface. The base station and UE communicate using RLC protocol in Acknowledge Mode (AM), where in an Automatic Repeat reQuest (ARQ) mechanism is used for error correction. The data generated at UE is carried over air interface with DCH transport channels, are mapped into different physical channels such as Dedicate Physical Data Channel (DPDCH) at physical layer. The configurations for all the UMTS nodes, links, channels and layers are listed in Table 1.

Table 1: UMTS Configuration

Channel connection:	DCH
LL Type:	RLC/AM
wired links and Iub	622Mbit
DCH Bit rate	200kbps
BS TTI:	10ms
Downlink TTI	20ms
Uplink TTI	20ms
Downlink bandwidth	200kbps
Uplink bandwidth	64kbps
Traffic	Http
TCP type	FullTCP

A Bluetooth node (BTNode) of UCBT is similar to wireless (WNode) node available in NS2. It consists of following components implemented with respect to ad hoc network formation: 1) Bluetooth Radio for air interface. 2) Baseband for connection establishment in piconet. 3) Link Manager Protocol (LMP) for establishing link setup. 4) Logical Link Control and Adaptation Protocol (L2CAP) for providing connectionless and connection services for upper layer protocol. L2CAP adapts the upper layer protocol to the baseband layer. 5) Bluetooth Network Emulation Protocol (BNEP) for delivering the network packets on top of L2CAP. It acts as Bluetooth MAC for upper network layer (IP) protocols. A BTNode part of phone forms piconet with other BTNode of phones using LMP to establish Bluetooth links. The BNEP connections are formed among these BTNodes for exchanging the network layer data. The configurations for all the Bluetooth nodes, links, channels and layers are listed in Table 2.

Table 2: Bluetooth Configurations

Packet type	DH5 or 3-DH5
Inquiry and Page time	(3,2) sec
Link type	ACL, Asymmetric
Air data rate	1 or 2Mbps

7.2 Higher layer stack components for overlay communication.

At the BNEP layer of the Bluetooth stack, we have implemented a helper layer called Socio-Adhoc helper layer. It periodically `monitors the bridge table or routing table for any updates sent by the ad-hoc routing agents such as AODV. Cross layer interactions are used to communicate these updates to the higher layer. It also returns the breadth-first list of all physically reachable nodes along with its associated resource cost as computed based on equation 2. We used AODV routing agent for forming the ad-hoc network. We used FullTCP[44] as a transport layer agent, which enables bidirectional data transfer found efficient in a client-server or P2P based communication. Currently ns-2 does not support application payloads which are required to send and receive application- level control messages. Hence, we used another ns-2 extension called GnutellaSim[45], packet level P2P simulation framework that simulates Gnutella protocol on a wired network. GnutellaSim also has a socket framework which allows P2P protocol to be emulated on the simulated sockets. Gnutella framework has been implemented in the following layered architecture: Application layer, Protocol layer and Socket Adaptation layer. An application layer implementation called GnutellaApp class maintains peer relationship and acts as a direct interface for users. Another implementation called GnutellaAgent class at protocol layer implements Gnutella protocol. NSSocket class at socket layer provides socket interfaces to TCP implementation.

Unlike Gnutella overlay, BuddyShare overlay are formed on lesser number of nodes and has to adapt to the ad-hoc network. Thus, we have simplified the Gnutella to work as a BuddyShare by incorporating the following changes:

1) All overlay nodes run SocioApp (as shown in Figure 17), a middleware application that implements all the discussed processes in Chapter 3: Group discovery, Group creation, Group maintenance and Job Arbiter. SocioApp is an extension of GnutellaApp that maintains the peer relationships. GnutellaApp establishes peer relationships to every peer registered in bootserver until its maximum limit. SocioApp modifies this way of establishing relationship. The peer relationships are established only if the aggregated cost validates a node as socially trustworthy and has good wireless connectivity. It follows the Group creation protocol. SocioApp is used by a wrapper application called PhoneApp for download job distribution.

2) SocioApp uses SocioAgent to send and receive control messages. SocioAgent is an extension of GnutellaAgent which performs protocol dependent message transmission and processing. The Gnutella control messages such as PING, PONG, CONNECT, GNUTELLA OK, and GNUTELLA REJ have been retained in implementation of overlay protocol. The extra messages used are JOBSEGMENT and JOBCOMPLETE. The role of JOBSEGMENT is to assign a segmented job to neighbor peers. Once the job is completed, the Job-Complete signal is sent back to the initiator.

3) We have disabled the procedures for lookup, file search at SocioApp and implemented the protocol for job distribution in *Job Arbiter* for concurrent download.

The downloaded data file is sent independent of the overlay network, directly to the receiver. A data sender application opens up a new TCP connection with receiver directly and sends the data. The socket layer is not modified.

7.3 Higher layer stack components for HTTP communication

An UE node of UMTS network is used as a primary interface of the phone node. The Http server resides in the Internet and the Http client resides on UE of UMTS network. The client server communication is carried on FullTCP which approximates the real world web access behavior [48]. In order to accurately model the web traffic in a mobile phone, we have used another ns-2 extension called PackMime [48]. It supports web traffic generation in ns-2. PackMimeHTTP is the class that drives the generation of HTTP traffic. Each PackMimeHTTP object controls the operation of two types of applications: *Server* and *Client*. The server and client applications are connected to a TCP Agent (FullTCP). FullTCP agents are attached to the network interface stacks of UE node. We have not modified the server application which resides on a node in internet. We have modified the HTTP client which additionally recognizes the presence of BuddyShare overlay for data distribution. Every HTTP request-response pair generated at a client is sent to SocioApp for segmenting the jobs using HTTP byte-range requests.

7.4 Coordination between stacks

The interaction between the Bluetooth stack and UMTS stack is achieved through a class called PhoneApp. The HTTP client simulates the web user behavior which generates HTTP requests based on size and inter-arrival distribution functions as described in [47]. This behavior is modeled for general Internet users and not for mobile users. However, these traffic patterns are a good approximation to the mobile internet user. The HTTP client in UMTS stack has two roles in coordination: First, it passes every HTTP request to PhoneApp for job distribution. Second, it reads the job queue of PhoneApp for downloading data from the internet server. PhoneApp maintains a priority queue of active jobs for HTTP client to download. This queue is populated by the *Job Arbiter* process of SocioApp application whenever it reads a job from output queue. When a job data is downloaded by the HTTP client, the job is dequeued and the downloaded data is sent directly to the owner through separate TCP connection. The PhoneApp also signals SocioApp to send a Job-Complete message to its immediate leader.

7.5 Simulation description and results

The simulations are done in two phases. One in which we used customized scenarios to evaluate the behavior of BuddyShare overlay on different parameters such as: 1) Request and Response size, 2) Number of overlay nodes, and 3) Job size.

The first phase simulation evaluates the following overlay behavior: 1) Achieved download completion time. 2) Achieved goodput, number of bytes received over total time under different conditions. In the second phase of simulation, we apply the BuddyShare simulator to the scenarios extracted from the reality dataset and show the enhancement whenever socially related friends are near. The configuration for parameters is given in Table 1, 2 and 3.

Table 3: Simulation Parameters

Active phones	2-10
Area:	15x15 square meters.
Base stations	1
Download initiators	1
Response size mean	22.9 KB
Number of scenarios:	741 in 30 days
Simulation length:	500 sec per scenario

FullTCP is used as the transport agent for both client-server communication in UMTS and P2P communication in Bluetooth. Hence, we first observe the throughput performance of FullTCP in an UMTS network as depicted in Figure 20 for DCH data rate of 200Kbps. Since FullTcp uses a bidirectional data transfer, the throughput is calculated as sum of total file size received at both the ends over the total time. As shown in the Figure 20, the throughput saturates around 184Kbps with increasing file size. A smaller file size gives lesser throughput due to the slow start phase of congestion window algorithm.

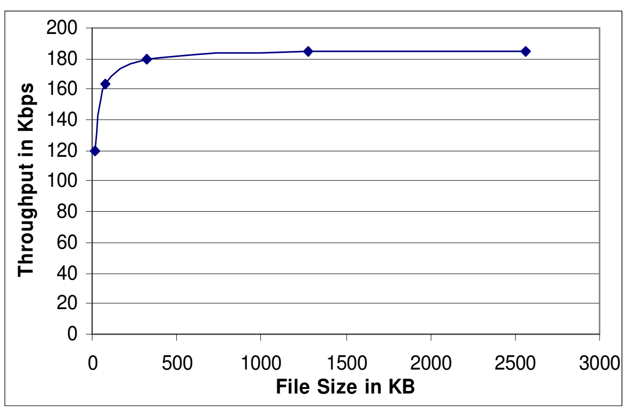

Figure 20: FullTCP Throughput in UMTS

Next, we study the performance of FullTCP in Bluetooth network (see Figure 21). In order to obtain the FullTCP performance, we created a scenario only with two nodes. The nodes formed BNEP connections. The gross air data rate in this case was 1Mbps.

No loss model was used. One node acted as a sender of a file and other as a receiver. The throughput saturates around 500kbps as shown in Figure 21, which is higher than 184kbps.

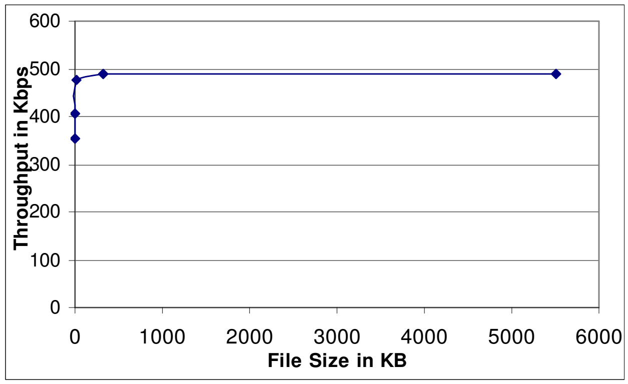

Figure 21: FullTCP in Bluetooth

Next, we study the performance of the overlay by distributing the download job. We created a scenario of piconet of 8 phones with 2Mbps air data rate and each phone has 200Kbps DCH connection. The other parameters are listed in table 1-3. We assume every user is socially trustworthy, thus overlay of 8 peers is formed. Only one download requestor at a time is considered and the download request is segmented in equal sized fragments and sent to the overlay neighbors. The overlay neighbors download in parallel and send data in the ad-hoc network.

The performance of download time based on varying number of overlay neighbors for different file sizes is shown in Figure 22. It can be observed that download time decreases exponentially for higher file sizes. This shows the BuddyShare overlay significantly reduces the download time for larger file size with increasing number of collaborators. To understand how overlay increases the net download rate, the download rate v/s the number of peers is plotted as shown in Figure 23. Download rate scales linearly for higher file sizes. For example, a user collaborating with two friends and downloading files of size greater than 1MB will get a download rate of 420Kbps on average because of three parallel connections. If the social overlay is not used then one would get only around 184kbps. The download rate will eventually flatten with increasing number of peers as active peers increase load on the Bluetooth network.

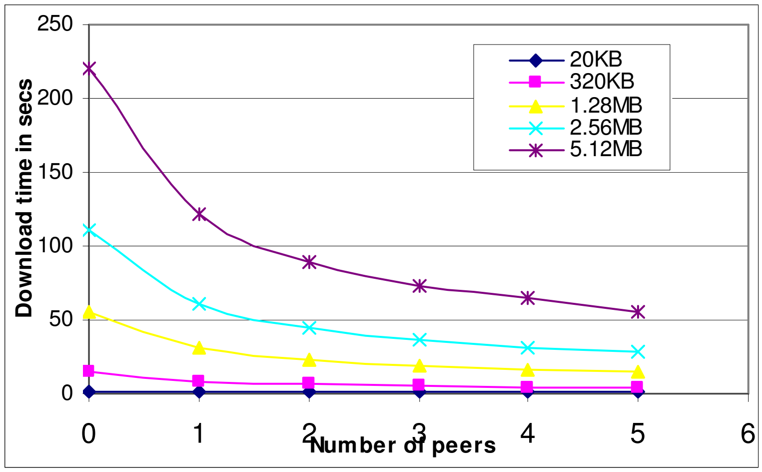

Figure 22: Download Completion time

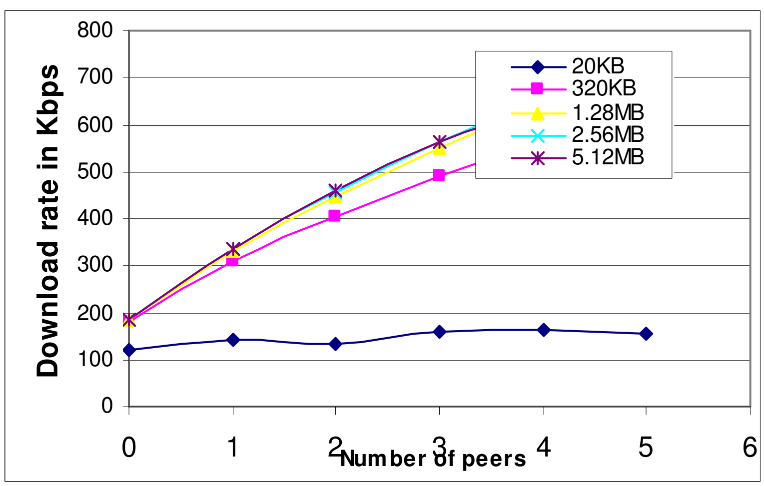

Figure 23: Download rate

As observed from all the Figures 21-23, the data rate is high for a higher file size than a lower file size. A larger file download causes TCP to increase the congestion window to its maximum and the ad hoc data rate grows to around 500kbps and cellular data rate to 184Kbps as shown in Figure 20 and 21. Whereas a smaller file size gets distributed at relatively slower rate due to header overhead and slow growth of the congestion window. Thus the smaller files do not get downloaded at the same rate. This factor affects the performance of job segmentation mechanism of a BuddyShare overlay.

In job segmentation mechanism, a download request is segmented into equal sized fragments. Since fragment or job segment size affects the performance of overlay, the download time based on different job sizes is plotted in Figure 24. The optimal block size is around 64KB. A very small fragment size will affect the TCP performance because of connection overheads and small congestion window. On the other hand, a large fragment will take a long time to do download from internet. This keeps the ad hoc network idle and receiver in waiting for relatively longer time which increases the download completion time. Hence both smaller and higher job size does not have best performance as seen in the Figure 24.

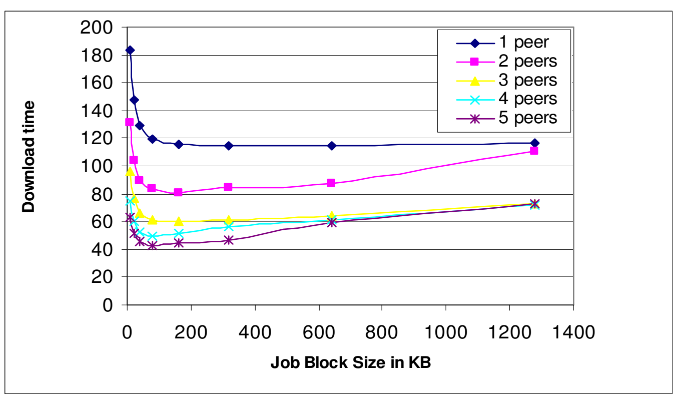

Figure 24: Download time on different job sizes

In order to realistically evaluate the usefulness of BuddyShare, we simulate it on scenarios extracted from the logs of mobile users from the Reality Dataset. We picked 30 continuous days of data randomly from one year of data. The simulator program, attached a phone node to each of the 97 participants. The nodes are programmed to perform operations such as calling a person, moving, sensing a device as the participants performed in those 30 days. The actions are obtained from the call logs and Bluetooth logs. Though there was not any interaction of downloading web data, we made an assumption that a user with maximum number of neighbors downloads a file whenever s/he detects a presence of BuddyShare overlay. We limited all the scenarios to only a maximum of 8 nodes, as we considered only Piconet and not Scatternet. The nodes perform overlay formations. We have used the web traffic generation studied in [47] to get the distribution of file size. Since the mobile phone is smaller in size, the mean of file size was chosen around 22.9 KB. The simulator picks up a file size from the distribution and starts the download process for the highest degree node for each scenario. The other configuration parameters for the simulation scenarios are shown in Table 1-3.

Four different ways of forming the overlay are evaluated for performance: 1) S1: It forms an overlay with any nearby node until 2 hops. No social or resource constraints. 2) S2: It forms an overlay with those nodes within 2 physical hops with sufficient aggregate cost computed from equation 2. The trust part of aggregated cost used in Group-shortlist process is based on *relaxed-social 2-hop call graph* (as discussed in Chapter 6, page 59). 3) S3: It forms an overlay with any nearby node with just 1 physical hop away. It uses *strongly-social 1-hop call-graph* for computing aggregate cost. 4) S4: It forms no overlay.

Every scenario per day is averaged and the average download rate achieved per day is shown in different walls of Figure 25. As can be seen, S1 formed without any constraints has the best performance in terms of achieved average throughput per day. Wall S2 builds the overlay network on certain social relationship and it almost matches the performance of S1. This infers that socially close people come often geographically close validating our hypothesis. Wall S3 uses a stronger social relationship also performs better than S4. S4 does not have any overlay, thus it has the lowest download rate. This graph concludes that any of the BuddyShare overlay scales twice as much on average in a university environment. For now, it cannot be generalized for other environments.

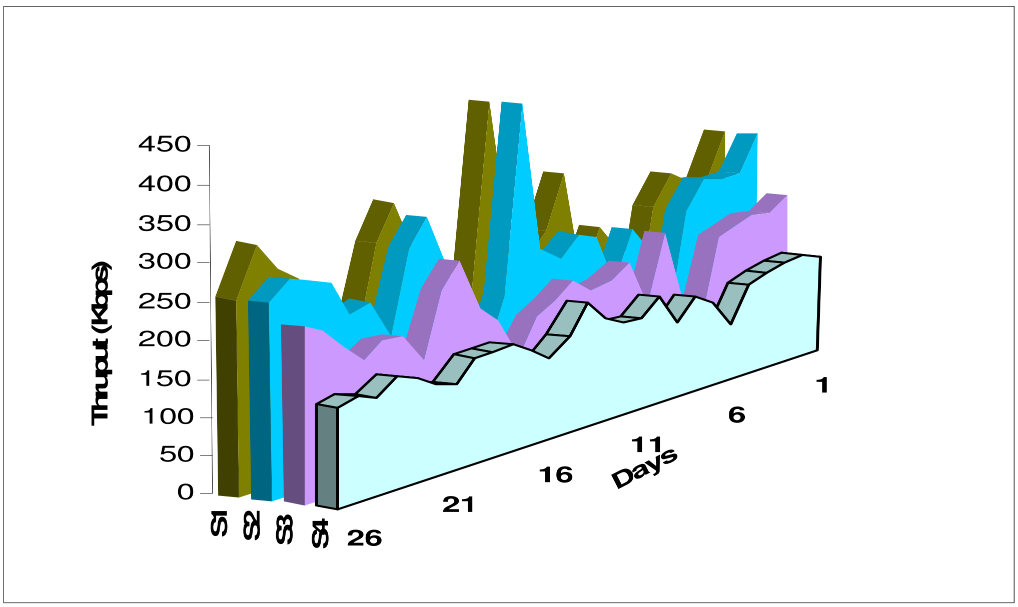

Figure 25: Throughput of different overlays at different points in time

CHAPTER 8

CONCLUSION

In this thesis work, we presented research on overlay networks augmented with concepts from social networks that will aid an active trustworthy collaboration among mobile users to increase the net data speed. This concept not only improves the data speed, it brings humans one step closer to collaborate and share the resources for mutual benefits. Our vision shapes out in a form of creating an overlay network on mobile devices which takes care of social aspect of human beings for collaborating with other devices.

Henceforth, we presented an overlaid mobile Internet sharing system called BuddyShare. It uses the social context of the user from mobile phones to share the Internet with friends automatically. We validated this idea by analyzing logs of real mobile users. The analysis showed that on average around 3.5 friends are around for a long time. This acted as the motivation of the BuddyShare system. BuddyShare undergoes four process states to create, maintain and distribute jobs in overlay network. It maintains the social trustworthiness by only creating overlay links with satisfied social and resource cost.

In order to validate the hypotheses, we collected various datasets pertaining to university settting. We analyzed user behavior in this university setting to study the potential of this collaborative overlay to share internet. The social behaviors of sample set collected from university environment found to support the cause of mobile sharing medium.

Finally, we constructed an ns-2 based flexible simulation framework for mobile phones that have dual interfaces and can participate in 3G and Bluetooth network. We simulated the real event traces on this framework and concluded the usefulness of BuddyShare as it scaled the data speed of an user by atleast 100% on average.

BIBLIOGRAPHY

[1] Marsh, S. "Formalising trust as a computational concept". PhD thesis, Department of Mathematics and Computer Science, University of Stirling, 1994

[2] Castelfranchi, C. and Falcone, R. "Social trust: A cognitive approach". In *Trust and Deception in Virtual Societies*. C. Casteleranchi and Y.-H. Tan, Eds. Kluwer Academic Publishers, Dordrecht, Holland, 2002.

[3] Richardson, M., Agrawal, R., AND Domingos, P. "Trust management for the semantic Web", *Proceedings of the 2nd International Semantic Web Conference*. Sanibel Island, FL. 2003.

[4] Ziegler, C. N. AND Lausen, G. "Spreading activation models for trust propagation". *Proceedings of the IEEE International Conference on* E-*Technology,* E-*Commerce, and* E-*Service*, Taipei, Taiwan. 2004

[5] Josang, A.. "The right type of trust for distributed systems". *In Proceedings of the 1996 New Security Paradigms Workshop*. 1996.

[6] Watts, D. "Small Worlds: The Dynamics of Networks Between Rider and Randomness". Princeton University Press, Princeton, NJ, 1999

[7] "BLUETOOTH SIG ADOPTS NEW CORE VERSION 2.1 + EDR", Bluetooth SIG Group, July 2007. URL:http://www.bluetooth.com/Bluetooth/Press/SIG/BLUETOOTH_SIG_ADOPTS_NEW_CORE_VERSION_21__EDR.htm

[8] Dave Wisely, Philip Eardley, Louise Burness, "IP For 3G: Networking Technologies for Mobile Communications", John Wiley and Sons, page 10-12, 2002, ISBN 0471486973.

[9] N. Eagle, A. Pentland. "Reality Mining: Sensing Complex Social Systems", *Personal and Ubiquitous Computing*, 10(4): 255-268, 2006. URL:http://reality.media.mit.edu/dataset.php

[10] Papadimitratos, P. Sankaranarayanan, S. Mishra, A., "A bandwidth sharing approach to improve licensed spectrum utilization", *Communications Magazine. IEEE*, vol 43. issue 12, supl.10- supl.14, 2005

[11] J. Mitola, III, "Cognitive Radio for Flexible Mobile Multimedia Communications," *IEEE Int'l. Wksp. Mobile Multimedia Commun.*, Nov. 1999, pp. 3–10., 1999

[12] Amit Nanavati, Gautam Das, Siva Gurumurty, Dipanjan Chakraborty, Koustuv Dasgupta, Sougata Mukherjea, Anupam Joshi. "On the Structural Properties of Massive Telecom Call Graphs: Findings and Implications"**.** In *Proceedings of ACM Conference on Information and Knowledge Management* (CIKM), Arlington, VA, USA, Nov 6-11, 2006

[13] Culotta, A., R. Bekkerman and A. McCallum: "Extracting Social Networks and Contact Information from Email and the Web"., *Proceedings of the First Conference on Email and Anti-Spam (CEAS)*, (2004).

[14] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On Power-law Relationships of the Internet Topology. In Proceedings of ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pages 251–262, 1999.

[15] J. Kleinberg. Authoritative Sources in a Hyperlinked Environment. In *Journal of ACM*, volume 46, 1999.

[16] M. E. J. Newman. The Structure and Function of Complex Networks. *SIAM Review*, 45:167, 2003.

[17] W. de Nooy, A. Mrvar, and Batagelj. V. *Exploratory Social Network Analysis with Pajek*. Cambridge University Press, 2005.

[18] Wai Ho Au Chan and K. C. C. Xin Yao. A Novel Evolutionary Data Mining Algorithm with applications to Churn Prediction. *IEEE Transaction on Evolutionary Computation*, 7(6):532– 545, Dec 2003.

[19] Y. L. Sun, W. Yu, Z. Han, K.J., and R. Liu. "Information theoretic framework of trust modeling and evaluation for adhoc networks". *IEEE Journal on Selected Areas in Communications*.

[20] G. Ananthanarayanan , R. Venkatesan, P. Naldurg , S. Blagsvedt, and A. Hemakumar, *"*SPACE: Secure Protocol for Address-Book based Connection Establishment*"*, *ACM HotNets*, Irvine, CA, USA, Nov 2006.

[21] Chao Gui Mohapatra, P. "Efficient overlay multicast for mobile ad hoc networks", *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, 1118- 1123, vol.2, 2003.

[22] Zahn, T.; Winter, R.; Schiller, J. "Simple, efficient peer-to-peer overlay clustering in mobile, ad hoc networks", Proceedings. *12th IEEE International Conference on Networks, 2004. (ICON 2004)*, 520-524, vol 2, 2004

[23] Stanze, O., Zitterbart, M. "On Demand Overlay multicast in mobile ad hoc neworks", *Wireless Communications and Networking Conference, 2005 IEEE*,2155-2161, Vol 4, 2005.

[24] C. E. Perkins, E. M. Belding-Royer and S. R. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", RFC 3541, *Internet Engineering Task Force*, July 2003.

[25] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)", RFC 3626, *Internet Engineering Task Force*, October 2003.

[26] Toshiaki Osada, Gen Kitagata, Debasish Chakraborty, Takuo Suganuma, Norio Shiratori, "An Effective Application-Level QoS Routing Scheme for MANETs", *Proceedings of the 2007 international symposium on applications and the internet,* page 18, 2007.

[27] D. Agrawal. C. Aggarwal. "On the design and quantification of privacy preserving data mining algorithms". *Proc. PODS*, 2001.

[28] Hay, M., G. Miklau, D. Jensen, P. Weis, and S. Srivastava (2007). "Anonymizing social networks". *University of Massachusetts Amherst, Technical Report 07-19*, 2007. URL: http://kdl.cs.umass.edu/papers/hay-et-al-tr0719.pdf.

[29] The network simulator - (ns-2). http://www.isi.edu/nsnam/ns

[30] UCBT simulator. https://www.ececs.uc.edu/ cdmc/ucbt/

[31] E. Daly and M Haahr (2007), **"**Social network analysis for routing in disconnected delay-tolerant MANETs**",** *MobiHoc '07,* ACM Press, 32-40. URL:http://reality.media.mit.edu/pdfs/Daly.pdf

[32] C. Scheideler., "Overlay networks for wireless ad hoc networks"**,** In the *Proceedings of the IMA Workshop on Wirless Communications*, 2005..

[33] S. Basagni and C. Petrioli. A scatternet formation protocol for ad hoc networks of bluetooth devices. In *IEEE Vehicular Technology Conference(VTC)*.

[34] Wang, B., Bodily, J., Gupta, S.K.S. "Supporting Persistent Social Groups in Ubiquitous Computing Environments Using Context-Aware Ephemeral Group Service". In Proceedings of the *Second IEEE International Conference on Pervasive Computing and Communications (PerCom'04)*.IEEE Computer Society Press, 287-296, 2004. URL: http://citeseer.ist.psu.edu/wang04supporting.html

[35] Mirco Musolesi, Stephen Hailes and Cecilia Mascolo, "An Ad Hoc Mobility Model Founded on Social Network Theory", In Proceedings of the *7th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2004)*. Venezia, Italy. October 2004. URL:http://www.cs.ucl.ac.uk/staff/c.mascolo/www/mswim04.pdf

[36] Sewook Jung, Alexander Chang, Mario Gerla. "Peer to peer video streaming in Bluetooth overlays", In the Journal of *Multimedia Tools and Applications*, 2007.

[37] Jennifer Golbeck and James Hendler, "Inferring binary trust relationships in Web-based social networks", ACM Trans. Inter. Tech. 2006, pp 497—529.

[38] Carol M. Werner, Barbara B. Brown, Irwin Atman, Brenda Staples, " Close Relationships in their Physical and Social Contexts: A transactional Perspective", Journal of *Social and Personal Relationships*, Vol. 9, No. 3, 411-431, 1992.

[39] Watts, D. J., AND Strogatz, S. H. "Collective dynamics of 'small-world' networks". *Nature 393*, 6684 , June 1998,

[40] Connect: A social networking portal for students. http://dvd1.ecs.umass.edu/connect

[41] Newman, M. E. J. "Clustering and preferential attachment in growing networks". *Phys. Rev. E, 64(025102)* (2001). *CS-200006, Duke University* (2000).

[42] Eagle N, A. Pentland, and D. Lazer (2007), "Inferring Social Network Structure using Mobile Phone Data", *PNAS*, (in submission).

[43] EURANE User Guide (Release 1.6), SEACORN Project, 2005. Available via http://www.ti-wmc.nl/eurane/.

[44] Enhancement to FullTcpAgent and Sockets in ns2: http://www.cc.gatech.edu/computing/compass/gnutella/tcp.html

[45] Packet-level Peer-to-Peer Simulation Framework and GnutellaSim http://www.cc.gatech.edu/computing/compass/gnutella/gnusim.html

[46] M. Conti, E. Gregori, and G. Turi. A cross layer optimization of gnutella for mobile ad hoc networks. MobiHoc '05, pages 343–354, ACM Press.

[47] Stochastic Models for Generating Synthetic HTTP Source Traffic. J. Cao, W.S. Cleveland, Y. Gao, K. Jeffay, F.D. Smith, and M.C. Weigle. Proceedings of IEEE INFOCOM Hong Kong, March 2004.

[48] Web Traffic Generation in NS-2 with PackMime-HTTP: http://dirt.cs.unc.edu/packmime/.

[49] H.-Y. Hsieh and R. Sivakumar.2002. "A transport layer approach for achieving aggregate bandwidth on mutli-homed mobile hosts". MOBICOM, pages 83–94, Sept.

[50] R. Chakravorty, J. Chesterfield, P. Rodriguez, I. Pratt, S. Banerjee, "MAR: A Commuter Router Infrastructure for the Mobile Internet". ACM Mobisys, Boston, June 2004.

[51] C. Carter and R. Kravets. "User Devices Cooperating to Support Resource Aggregation.WMCSA '02, pages 59–69. IEEE, June 2002.

[52] AGORA: Open market architecture for collaborative mobile internet: http://pages.cs.wisc.edu/~suman/projects/agora/

[53] G. Ananthanarayanan, V. N. Padmanabhan, L. Ravindranath, and C. A. Thekkath, 2007. " COMBINE: Leveraging the Power of Wireless Peers through Collaborative Downloading", ACM MobiSys, San Juan, Puerto Rico.

[54] Z. Anwar, W. Yurick, V. Pandey, A. Shankar, I. Gupta, R.H. Campbell. 2005. “Leveraging Social Network Infrastructure to Improve Peer-to-Peer Overlay Performance: Results from Orkut. CoRR abs/cs/0509095.

[55] P. Androutsos, D. Androutsos, and A. Venetsanopoulos. 2006. “Small world distributed access of multimedia data: an indexing system that mimics social acquaintance networks,” Signal Processing Magazine, IEEE, vol.23, no.2pp, pp. 142– 153.

[56] R. Chakravorty, J. Chesterfield, P. Rodriguez, I. Pratt, S. Banerjee, “MAR: A Commuter Router Infrastructure for the Mobile Internet”. ACM Mobisys, Boston, June 2004.