

MACNET: Mobile Adhoc Collaborative NETWORKS

Jawwad A. Shamsi, S. Ali Raza, Hassan Farid, Abbas Mehdi, S. Taha Munir
Systems Research Laboratory, Computer Science
FAST- National University of Computer and Emerging Sciences, Karachi, Pakistan

Abstract — *Resource limitation is considered as a major bottleneck in smart phones. This paper aims to address this problem by proposing a collaborative ad-hoc network of smart phones which utilizes trust relationship among peer devices to provide a computationally powerful, resilient, available, and low cost solution. MACNET (Mobile Ad-hoc Collaborative NETWORK) implements a middleware to address the issues of collaboration, trust, and network connectivity. A large number of applications can utilize the MACNET framework for improved connectivity, rapid response, and better communication. The paper elaborates sample applications and addresses the architectural details of the solution. It also explains the implementation details and provides results to demonstrate the effectiveness and benefits.*

Index Terms — Collaborative computing, Smart phones, Resource limitation, Trusted neighbor devices.

I. INTRODUCTION

Smart phones provide interesting paradigm of computing, which consists of numerous benefits and exciting challenges. Compared to conventional computers, smart phones exhibit advantages of mobility, easy interface, and affordable prices. On the other end, they expose challenges of limited resource availability, varying network connectivity, and heterogeneity. This combination of challenges and benefits attracts both the users and the researchers.

While users are eager to utilize the benefits of smart phones through a variety of applications [16]; researchers are focused on enhancing user experience; either by developing efficient and user-attractive applications, or by exploring additional means of enriching smart phones.

One possible way to enrich resource constrained smart phones is using cloud computing through which applications exploit increased availability and higher computing power. A large number of such applications have been discussed and mentioned in the literature [14] [18] [19]. However, they are based on an assumption that a rapid connection to a cloud is available such that offloading to the cloud is beneficial. This assumption is not always true; using a remote cloud can often lead to unnecessary delay, interruption, and unavailability.

This paper is motivated by the above mentioned challenges and considerations. Instead of porting smart applications on distant-clouds, the focus of this research is on building an ad-hoc network of smart phones through which users can collaborate and experience improved performance and enriched behavior.

The main idea is based on this observation that in a closed environment, such as office space or a class room, smart phones exhibit two significant characteristics. 1) They exist in abundance, and 2) a trust relationship exists among them. Motivated by these two observations, this paper proposes MACNET (Mobile Ad-hoc Collaborative Network) - a framework for applications to exploit collaboration in mobile ad-hoc networks of smart phones. The main theme is based on the utilization of available resources and trust relationship, in order to form a collaborative computing platform in which users could either be service providers or beneficiaries.

Through MACNET smart phone users can achieve improved performances. Depending on the type of application, the computing can either be serial or parallel. It may either utilize higher computing power or exploit services from neighboring peers in the system. Examples of such collaborative computing include live broadcast of a video stream through a mobile phone in a bus, audio synthesis of a lecture in a class room, seeking service-opinion from customer's waiting at a doctor's location, and scheduling meeting and avoiding conflicts at an office. In all the examples mentioned above, applications can utilize the MACNET framework for enhanced performance, such as saving network bandwidth and resources, increased computing power, provide data insights in physical neighborhood and rapid response to reach consensus. It should be noted that MACNET does not require a central cloud. Instead, smart phones may collaborate either in a peer-to-peer manner or they may seek services in a centralized mobile phone. Participation in such a network is on volunteer basis, which may be lucrative in order to get mutual benefits.

Development of such a framework is challenging: First and foremost, the framework has to be flexible, adaptable, and comprehensive in order to support a

wide-range of applications which could benefit from a mobile ad-hoc network. Second, considering the wide-range of different types of smart-phones, it should offer efficient mechanisms for service and device discovery. In addition features such as utilization of trust among participants, efficient usage of resources, and device heterogeneity are also significant.

MACNET is motivated to solve these problems. It incorporates a powerful middleware, which offers multiple features such as unicast and broadcast communication, storage, service and device discovery, and utilization of trust services. The features are beneficial for both the developers of applications and their users to exploit benefits of collaborative computing in order to enrich smart phones.

This paper describes the design of the MACNET framework and discusses its architecture. In addition, sample applications which can benefit from MACNET are also elaborated. The paper also evaluates effectiveness of MACNET by developing collaborative applications on real smart phones. The experiments confirm the effectiveness and usability of MACNET for a variety of applications.

In the remainder of this paper, we describe benefits and requirements for collaborative computing in smart phones in section II and describe the architecture of MACNET framework in Section III. In section IV, we explain the experimental setup and results. Section V presents significant work by other researchers and section VI concludes the paper.

II. BENEFITS AND REQUIREMENTS FOR COLLABORATIVE COMPUTING

In order to understand the effectiveness of the MACNET framework, it is imperative to realize the need and benefits of collaboration among smart phones. Further, potential applications are also needed to be identified. The purpose of this section is to address these issues. In addition, the section highlights the requirements for extending collaborative environment.

A. Motivation

Collaborative computing refers to group-aware computing in which devices or users collaborate to achieve a common goal. Collaboration enhances the utilization of resource constraints smart phones and leverages their power for improved usage. Collaboration among smart phones is a novel idea presented in this paper. Three important characteristics of smart phones have motivated this idea.

1) Several applications in smart phones are thin client applications, in which processing is handled at a remote server and a thin client is used to facilitate the access to

the user. This is done either because of service dependency or limited resources of smart phones.

2) In a closed environment such as a class room or office space, smart phones exist in abundance and exhibit some level of trust among them.

3) Smart phones can form an ad-hoc network themselves without the need of any additional infrastructure. This extends opportunities for developers to create applications that exploit mobility and abundance of smart phones.

B. Opportunities for Collaborative Computing

Effectiveness of collaborative computing can be well understood using the following two examples which also explain the two modes of collaboration.

1) Centralized Mode

For instance, passengers riding on a bus may be interested in watching a live stream of a soccer match. Instead of having each user tune to the stream individually, a single user in the bus can tune to the stream and provide access to other users. Collaboration in this scenario can reduce stress and congestion on the network and increase availability. Collaboration is handled through a single user such that she is responsible for facilitating services to other peers of the ad-hoc network.

2) Peer-to-Peer Mode

Collaboration can also be extended in a distributed and peer to peer environment, in which each user has an identical role. Example of peer to peer mode includes computationally extensive games where collaboration increases memory availability and increases execution speed.

Collaboration in the above mentioned scenarios can be facilitated through an ad-hoc network which utilizes trust relationship among devices. While both the centralized and the P2P approaches are attractive to users, it may be argued that the centralized approach offer little attraction for the user providing services and yields additional resource constraints on the device which is offering services to all the users. These constraints can be compensated either through monetary means or via a credit based system, which may be agreed on mutual basis.

C. Sample Applications

A few sample applications which can benefit using the two modes of collaboration are described below

1) DNS Resolution:

During a mobile browsing session, a user sends numerous requests to DNS server for domain name resolution. For instance, the domain `www.something.com` needs to be resolved to an IP address for accessing a web page. The DNS records are

stored on global DNS servers across the globe. DNS lookup queries may take excessive time for resolution due to long path to DNS servers and heavy internet traffic.

The stated issue can be easily resolved by obtaining DNS resolution from nearby source. In a collaborative ad-hoc network, any smart phone device can initiate a request for DNS resolution. If any of the neighboring devices have DNS resolution to the requested domain, the device will respond to the query and user will find lesser delay in usual browsing of web pages.

2) *Multimedia Streaming:*

Internet includes a large proportion of multimedia contents. Websites including lives streams, daily news, sports, and entertainment are enriched with multimedia such as audio and video. Downloading these contents require sufficient bandwidth and consumes considerable amount of resources.

On the contrary, smart phones have limited resources and network connectivity. Collaboration in this scenario could be useful in efficient utilization of resources. Instead of having each user download a separate stream from the Internet; users could download it from one of the neighboring device. A collaborative network could also be helpful in mitigating network congestion and enhancing user experience.

3) *Social Gaming:*

Social gaming has been an evolving technology. From socially connected games such as Facebook to Massively Multiplayer Online Role Playing Games (MMORPG), social games are attractive to a large community.

Collaborative computing can also provide opportunity for a completely new paradigm for social games on mobile devices. Developers using the collaborative framework can develop ad-hoc social games and exploit local and closed social networks. Such games have the potential to be highly capable and revolutionary as they can be played without any dependency on the Wide Area Network. Collaborative games will also eliminate the requirement of having a dedicated server.

4) *Collaborative Meeting Planner*

Collaborative computing in smart phones can also be used for online voting. Further, it can also be helpful in reaching consensus or planning a meeting. Conventionally, a calendar tool or e-mail is used to communicate and reach consensus. However, this may cause longer delays in reaching a consensus. It will also require a connection to the Internet in order to interact via an Internet-based calendar service. To encounter the problem, collaborative computing can be used as a meeting planner.

The above mentioned sample applications highlight the effectiveness of collaborative computing which can enhanced user experience and yield efficient resource utilization.

D. Requirements for Collaborative Computing

There are several challenges and requirements which must be fulfilled in order to promote collaborative computing among smart phones.

1) *Trust Relationship*

Collaboration must utilize trust among peers in order to assign priority and facilitate communication among them.

2) *Service Discovery*

A service discovery mechanism is needed in order to identify potential applications and resources for collaborative communication. Without service discovery all nodes will remain disconnected in an apparently connected network.

3) *Device Churn*

The device discovery mechanism must be able to facilitate discovery of new devices. Similarly, the framework should be able to tolerate departure of devices present in the network.

4) *Group Communication*

For a collaborative computing framework, it is imperative to have support for group communication. Features such as Unicast, Broadcast, and Multicast must be supported.

5) *Efficient utilization of resources*

Smart phones have limited resources. The collaborative network must be able to utilize these resources efficiently. For instance, socket communication should not overly consume resources. Collaboration should not overwhelmingly consume resources of devices such as power and memory.

6) *Flexible Application Development*

The framework should facilitate development of applications which can utilize collaborative computing in order to achieve high performance.

7) *Security and Privacy*

Collaboration must not violate security or privacy requirements of a user. That is, confidential information or private data should not be shared.

MACNET is focused on meeting these challenges. In the next section, details of facilitating collaboration are discussed. This section covers the details regarding preparation of your manuscript for submission, the submission procedure, review process and copyright information.

III. MOBILE AD-HOC COLLABORATIVE NETWORKS

MACNET is motivated to facilitate collaborative computing among smart phones. It is a rich application layer framework for mobile ad hoc devices, which provides methods and an API for extending this collaboration. This section describes the architecture of the MACNET framework and explains its operation and implementation.

Collaboration is achieved through an ad-hoc network. Request for forming an ad-hoc network can be initiated either by a user seeking services from a collaborator or by a user who is willing to offer services to other peers. In either case, the MACNET framework supports formation of application-specific overlay network over an ad-hoc network. In that, the MACNET framework broadcasts the request to all the devices that are operational in its network vicinity. Other MACNET users may accept the request based on their comfort and level of trust with the initiator.

Users may also join an existing network by initiating a broadcast query. Therefore, it is possible that a user may join two or more application-specific overlay networks at a time.

A. The MACNET Architecture

Fig. 1 illustrates the architecture of the MACNET framework. It consists of two layers: 1) A middleware layer which provides all the features of MACNET and also facilitates collaboration among the devices; and 2) an application layer, which consists of applications that are built on top of the middleware and utilize MACNET to achieve enhanced performance. The middleware layer consists of set of libraries that are built on top of native Java communication libraries. It extends communication capabilities and allows abstracted functionalities such as message queuing and group messaging and communication with trusted nodes. Following services are supported by the middleware layer.

- Device and Service discovery
- Communication among devices
- Utilization of Trust
- Sharing of resources

The middleware consists of two sub-layers: The Service Layer and the Storage layer.

1) Service Layer

Service layer is the core of the middleware. It provides important services such as service discovery,

communication among peers, and utilization of trust.

Device and Service Discovery: The device and service discovery mechanisms provide facilitation to add new devices and services in the ad-hoc network.

The discovery mechanism is simplified and eliminates the need of remembering shared URIs. Instead of sharing WSDL files like in traditional SOA, the mechanism is encapsulated in middleware for ease of development. The discovery process starts with discovering all nodes available in the network. Then they are filtered to mobile devices running MACNET framework. These filtered devices expose services which are then discovered by other nodes in the network based on trust preferences.

Communication: Features for both sending and receiving of messages are part of the communication infrastructure. Point-to-point communication is supported through unicast; whereas, groupware communication is supported through multicast and broadcast. Messages are received based on the trust level between sender and receiver. A priority queue is maintained which prioritizes data and messages among applications. The queuing service is a necessary feature to support device unavailability and incorporate churn. In case of unavailability of a receiving node, messages are stored in a queue on temporary basis. If the receiving node enters the network within a specified time, the messages are resent from the queue. This feature significantly improves quality of user experience in an ad-hoc network.

Trust: Social noise is an overwhelming drawback of electronic social networks. Purging of unwanted information and filtering to gain social network value is vital. Considering this, a trust model is included in the middleware which utilizes application's feedback and user preferences to allow or block application specific traffic and communication between nodes. This ensures that the device ecosystem co-exists with human social network. A reinforcement learning based, generic trust model based on Q-learning [22] has been implemented for this purpose.

Applications may use it to give periodic or occasional feedback to middleware for trust shaping with other nodes. Fig. 2 shows different levels of trust supported in the system. In that, a value can go as low as -3 denoting a malicious user, or to +3 for an extremely trusted entity. Strangers are considered as neutral.

Sharing: For sharing of files, MACNET implements

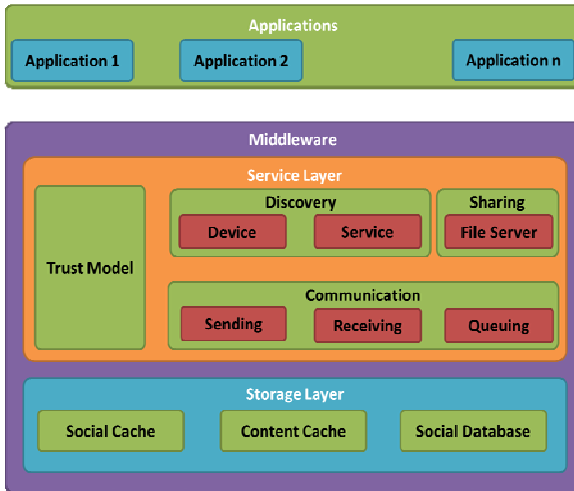


Figure 1 - MACNET's Internal Architecture

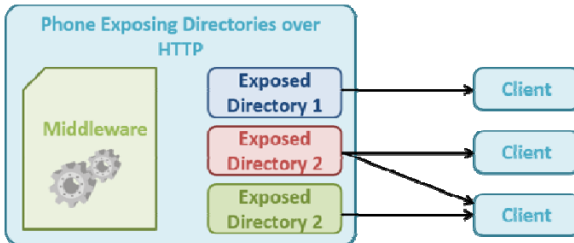


Figure 3- MACNET's File Sharing Mechanism

an efficient file server as a service to developers. Using this service, applications can expose directories for remote devices to access. This feature facilitates collaboration among applications that utilize sharing of data. The file server also promotes privacy, by restricting access according to the permissions of the file owner. Fig. 3 illustrates the concept. File sharing is implemented over HTTP. When a node intends to share a file, it calls expose directory function of the MACNET framework. The framework exposes the directory over HTTP on some random port. The application shares the port with other nodes, which they will use to explore and download files over the HTTP protocol. When a node wants to terminate the simultaneous sharing of multiple files session, it requests MACNET which closes the exposed directory. Exposing directory allow sharing files over HTTP permits browser-based access for many collaborative applications.

1) Storage Layer

The purpose of the storage layer is to facilitate,

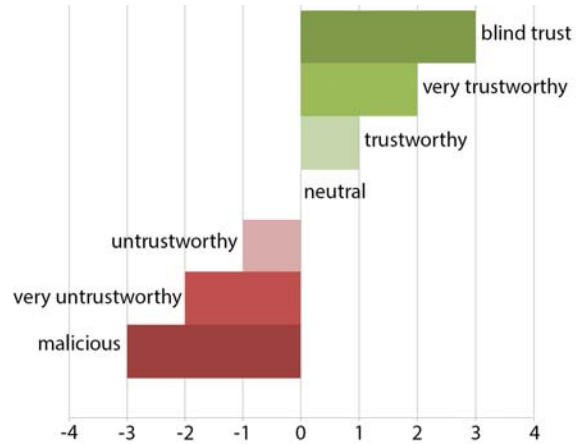


Figure 2 - Scale of Trust Relationship

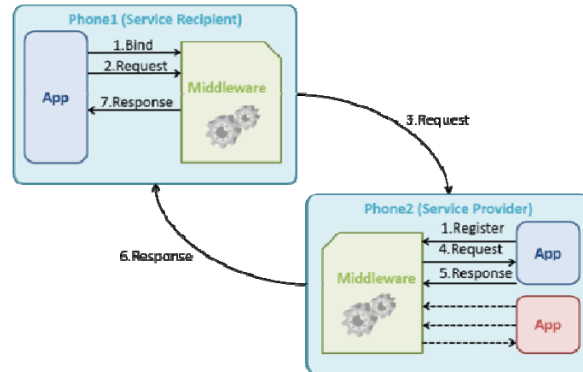


Figure 4- Flow of communication between applications

collaboration by maintaining data that is useful for social interaction. For this purpose, caching of social contacts and contents is maintained. In addition, a social database is also used.

Social Database: The social database maintains a list of all nodes that ever came in contact with a user's device. It helps in recognition of users and serves basis for the trust model.

Social Cache: A social cache is maintained to include a list of most recent contacts. The difference between social database and social cache can be understood through a small example. For instance, a person may have around 150 social contacts, but at any particular instance in an ad-hoc network she may have very low percentage of her contacts available. The social cache helps to keep data of only those contacts in memory which are active. This saves memory and reduces battery consumption

Content Cache: The content cache stores the actual data which is shared among devices [23]. The contents could include files, multimedia contents, streaming contents, documents, and pictures, etc. The data is

stored in the cache and is replaced after certain time period. Content cache allows scalable sharing of contents. In that, popular contents are replicated and stored in a cache, thereby decreasing load on the primary source.

B. Operation of MACNET

Fig. 4 explains the communication sequence for MACNET operations. For better understanding, the sample application of DNS resolution (mentioned in section II.C) is considered.

Suppose there are two mobile devices: 1) a service provider, and 2) a service requestor. The applications on both devices are built using communication features of the MACNET middleware. Each device embodies the middleware which acts as communication channel between other connected devices in the ad-hoc network. The middleware manages device discovery routines and stays updated with all connected devices in the ad-hoc network. The following sequence of steps describes the scenario.

1) Bind: The application at the service requestor end initiates and registers itself with the middleware.

2) Request: The application initiates a request for DNS lookup to the middleware.

3) Request: The middleware at the service requestor communicates the requests with all available and trusted nodes in the network. Middleware can also keep the messages in cache. This is useful if an application would like the message to be delivered to any device that may later join the ad-hoc network. For instance, service announcement messages may benefit from this scenario. The request is transmitted in an obscure manner such that the formatted packets that can only be read by middleware deployed on other devices.

4) The request is received by the middleware at the service provider end. Upon receipt, the listening middleware reads the message and identifies the request. It then checks its local registry if the requested service or application is bind to it. Upon verification, it forwards the request message to destined application.

5) The service provider application on the device responds to DNS resolution request. It sends the response to the middleware, in order to be delivered to the requested device.

6) The middleware at the service requestor device, can filter DNS resolution responses based on trust model and deliver the response from most authentic device to the application which requested the service.

C. Strategies for Performance Enhancements

MACNET incorporates several features in order to enhance performance.

1) Multi-threaded and Asynchronous Framework

Considering that the devices in the ad-hoc network may experience variable latency, all the functions in the framework use asynchronous call backs. That is an application does not wait for network response.

In addition, the framework is properly threaded in order to handle multiple on-device function calls simultaneously.

2) Power, Memory, and I/O Optimizations

Since mobile devices have resource constraints, special care was taken during development of middleware. All allocations are carefully made to reduce memory overheads. Service objects for example are initiated when only application requests them, and kept in memory only until at least a single application is using them.

The trust model requires persistent storage in the form of database. Certain applications might update trust frequently causing lots of I/O in SD Card. To counter this, data of nodes in the network is loaded in memory. Updates from memory are written back when application is closed or a node leaves the network. To conserve battery power, device discovery and service discovery are done intelligently. Instead of periodic broadcast of messages from each node for discovering devices and services, each device sends the discovery once. New nodes in network announce their presence to the whole network.

3) Device Churn

To incorporate device churn or detect devices which may leave abruptly, the MACNET middleware keep a check on all the nodes for last active communication. For devices from which communication is being made after a long delay, rediscovery mechanism is launched.

In the next section, experimental details of MACNET are discussed.

IV. EXPERIMENTS AND RESULTS

It is pertinent to assess the performance of the MACNET through experiments. For this purpose, an experimental test-bed containing ten smart phones is setup. The four sample applications mentioned in section II C are used. The setup consists of a dedicated link with 2Mbps Internet link capacity for uploading and downloading. We setup an ad-hoc network of smart phones to be used for evaluation of MACNET. The test

TABLE I
EXPERIMENTAL CONFIGURATION

PARAMETER	VALUE
Device Count	10
Processor	Single Core, 830 Mhz
Wi-Fi (For testing)	802.11 b/g/n
RAM	290 MB

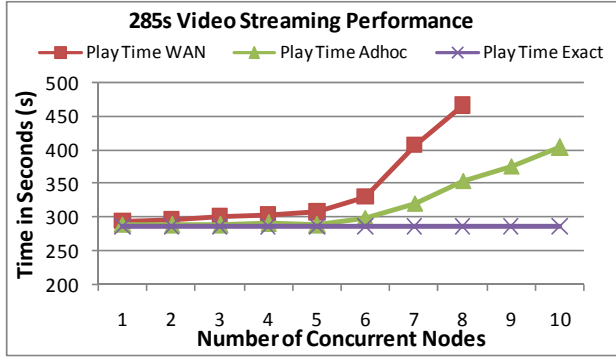


Figure 5 - Comparison of Video Streaming

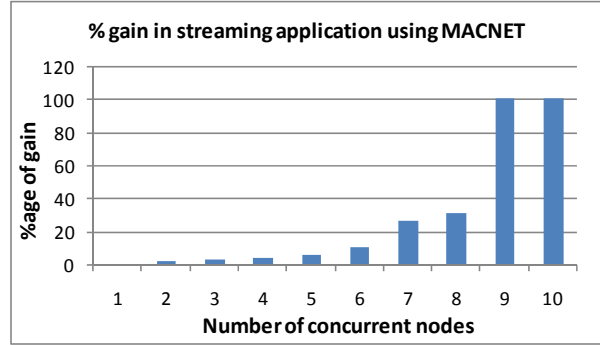


Figure 6. Gain in Percentage

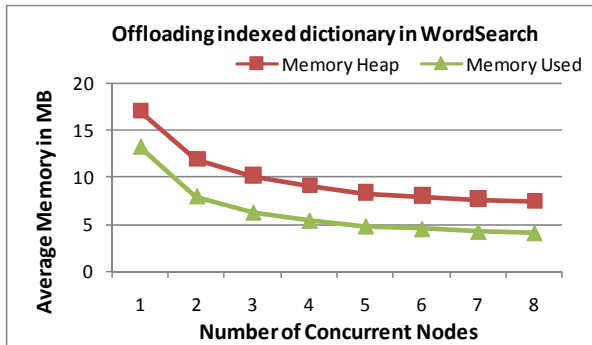


Figure 7. Social Game: Comparison of Memory Requirements

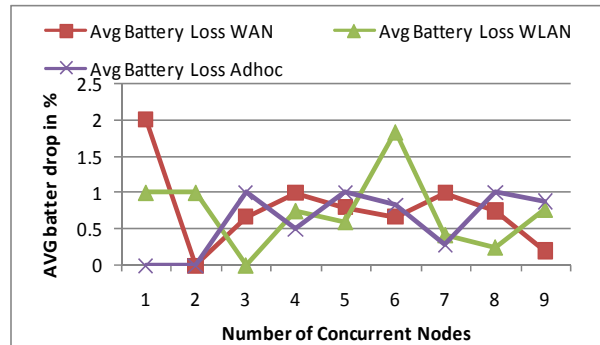


Figure 8. Average Battery Consumption

configuration environment is summarized in table I.

A. Need for collaborative computing on Smart Phones

Our initial experiments were motivated to establish the need of collaborative communication among smart phones. We sent a series of ping messages from LAN and WLAN to an external static IP on the Internet. The LAN environment included PCs and the WLAN environment consists of smart phones. We observed that that smart phones using WLAN exhibit higher fluctuations in latency as compared to LAN. It was also noted that higher fluctuations exist when smart phones communicate via the Internet. For communication among the smart phones, the latency of communication is much smaller and smoother. This set of experiment confirmed that when smart devices connect to a remote location, they are subject to high variability in network latency. Further, this observation states the need of an ad-hoc network which can facilitate collaborative computing.

B. Gain in Performance using the Streaming Application

As a next step, effectiveness of the MACNET framework was evaluated for streaming applications.

The experiment consists of playing a video stream using both the Internet connection and the ad-hoc network utilizing the MACNET framework.

For both the setups, effect on latency was noted while the number of smart phone users is gradually increased. Fig. 5 shows the playtime for a 285 seconds video. The purple line shows the actual playtime of the video. The green line shows the playtime when MACNET is used for ad-hoc network. The red line is the playtime when each user is directly downloading from the Internet. It should be noted that there is a massive difference in the latency. The difference increases with the increase in number of users, which confirms the scalability and effectiveness of the MACNET framework. During the experiments, it was also noted that the stream was unplayable when more than eight nodes tried to access the stream via the 2Mbps Internet link (each user is downloading independently). In comparison, MACNET was able to support the required number of ten users for accessing the stream. The effectiveness of the MACNET framework can also be understood via Fig. 6 which shows percentage gain of MACNET over WAN, where percentage is measured in terms of decrease in latency for playtime.

C. Gain in Performance using the social game

The efficacy of MACNET was also evaluated for the

social gaming application discussed in section II C. For this purpose, a popular word game, “Word Search” was developed using the MACNET framework. In this game, players join in a group playing session where a ‘random’ matrix of letters is generated at runtime. The task of searching all possible combination of valid words in matrix is a costly operation, where each possible word is matched with index of words in the English dictionary (with 25,000 words) in order to check the validity of the proposed word. Using the MACNET framework, the task of word verification was distributed among mobile devices. This provided an opportunity for resource constrained devices to participate in the game. Further, the distributed computation model yielded reduced resource consumption on the devices. Without the MACNET framework, a dedicated server and internet connection would be required or only a limited number of smart phones would have been able to participate due to computational complexity of game.

The dictionary game was deployed with increasing number of nodes, while the memory usage was noted. Fig. 7 shows the amount of total and heap memory used when game is being played with varying number of users. It should be noted that for one user, the complete dictionary is residing on the device, whereas with increasing the number of devices, the dictionary is spanned over multiple devices. MACNET enables better resource usage for the devices as more memory is available for other tasks.

D. Effect of Collaboration on Power Consumption

An important question about MACNET is to assess its effect on the battery life. For the streaming application experiment discussed above, battery consumption was observed for LAN, WAN, and the ad-hoc network. It was noted that the battery consumption is sporadic and it is inconclusive to observe any major difference on the battery life. Fig. 8 shows the comparison of the three schemes.

E. Effect of Collaboration on Scalability

Fig. 5 shows the effect of collaboration on the scalability of the system. While the WAN application was not able to support more than eight smart phones through the dedicated Internet connection, the MACNET framework was able to provide access to stream to all the ten devices with much lower latency. Even for other applications such as consensus based meeting planner, the MACNET framework was able to facilitate collaboration among a large number of devices.

The above mentioned results confirm the effectiveness of the MACNET framework in proving

low-latency and high performance solution with enhanced user experience. The MACNET framework provides higher efficiency, increases availability, promotes scalability, and facilitates improved utilization of resources such as memory and storage.

V. RELATED WORK

MACNET is an ad-hoc network for collaborative computing for smart phones. Table II compares the uniqueness of our approach with existing solutions.

A. Collaborative Computing

Maestro [2] is a middleware support tool for distributed multimedia and collaborative computing applications. It is based on group-based collaboration architecture, where users join groups and utilize available services. In comparison to Maestro, MACNET is focused on smart phones. A distinct feature is utilization of trust relationship among mobile phone users. This empowers smart phones for a variety of collaborative computing applications. It also enables visibility of all services in the network.

uBeeMe [26] is a mobile collaborative computing platform which focuses on peer to peer mode of collaboration. Unlike uBeeMe, MACNET is a more flexible and powerful tool which facilitates mobile application development for both centralized and peer to peer modes.

MACNET is also inspired by the work on crowd computing [13]. There are many applications that utilize power, wisdom and mobility of crowd. Applications such as minimal route prediction based on feedback from users in different areas on traffic and other conditions exist. The MACNET middleware provides great opportunities for developers and researchers to build applications using power of crowd. MACNET provides necessary foundation of communication and trust relationship. These two features can empower applications for a large number of users in a social network. One such application ‘Word Search’ has been discussed in detail in this paper. Users can play game in multiplayer environment and offload their computational and memory needs on other nodes.

Zhefan Jiang and Sangwook Kim proposed trust model ubiquitous ad-hoc networks [5]. MACNET incorporates a Q-learning algorithm - a reinforcement learning method, for the trust model. This helps mobile devices to become mature in assessing the trust scenario and opting for better path to service by accumulating each experience into a quantifiable value. Phone to phone ad-hoc communication for gaming has also been explored [10]. MACNET is inspired by this work; however, it is not restricted to social games. The

Table II - Comparison of MACNET with other related projects

CHARACTERISTICS	UBEEME [26]	MAESTRO [2]	VIMOWARE[6]	HAGGLE [21]	CUCKOO [7]	MOBICLIQUE [17]	MACNET
Ad hoc network	✓	✓	✓	✓	✓	□	✓
Distributed Computing	✓	□	✓	□	□	✓	✓
Collaboration	✓	✓	✓	✓	✓	□	✓
Overlay Networks/Groups	□	✓	✓	□	✓	□	✓
Middleware / Framework	✓	✓	✓	✓	✓	✓	✓
Device Discovery	□	□	□	✓	✓	□	✓
File Sharing	□	□	□	✓	□	□	✓
Asynchronous Callbacks	□	✓	□	□	□	□	✓
Peer-2-Peer Architecture	✓	□	□	✓	□	□	✓
Adaptive Trust Model	□	□	□	□	□	□	✓
Security	□	✓	□	□	□	□	□
Energy Optimization	□	□	□	□	□	□	✓

MACNET middleware introduces a generic framework which can be used for a variety of collaborative computing tasks. Similarly, Scohlten et al. [20] introduced decision support system using decentralized collaboration among smart phones. In comparison, MACNET is generic which can facilitate collaborative computing for a large number of applications.

A context-aware framework for the development of mobile collaborative computing has also been proposed [1]. Their approach is inspiring as MACNET can utilize context-aware applications for collaborative computing. MACNET is powerful in that it can be utilized for urgent data collection in case of disaster as proposed in [15] [16].

B. Mobile Cloud Computing

Mobile Cloud Computing refers to the use and study of computing, in which smart phones utilize cloud computing for enhanced performance and efficient utilization of resources [4][8][9]. The Cuckoo [7] framework utilizes computation offloading to save resources. Dynamic allocation is followed in that the amount of computation is decided at the run time.

In addition to computational enhancements, mobile cloud computing can also conserve resources. Kumar and Lu [12] discussed offloading techniques for smart phones. The authors described energy savings as the most desired feature and proposed offloading computation on cloud for saving energy. Kovachevet al. describes need and requirements for mobile cloud computing [11]. Huijun et al. proposed an offloading scheme [27] which models the application execution time and in unstable network conditions.

In comparison to all the related research mentioned above, MACNET is focused on forming a collaborative network of smart phones to enhance resources. This research is inspired by the idea that using a distant cloud is not always beneficial. In many scenarios, collaboration among peers can be used for enhanced performance. The MACNET framework can be

utilized either as a peer-to-peer system or as a client-server architecture.

C. Mobile Ad-Hoc Networks

Computing in Mobile Ad hoc networks (MANETS) has become more feasible with increased processing power, memory, and faster connectivity mediums. Existing work in this field was really helpful in developing MACNET [26].

Haggle [21] is an architecture for mobile devices that enables seamless network connectivity and application functionality in dynamic mobile environments. Scott et al. have identified that current application exploiting distributed architecture must contain significant network binding and protocol logic, which makes them inflexible to the dynamic networking environments facing mobile devices. Haggle supports Bluetooth and 802.11 wireless mediums for communication between nodes. This work is inspired by Haggle’s architecture that focused on application level code independence from network level requirements. However, considering scalability and reliability, MACNET is based on WiFi network only.

MANETs have also been used for data management and data sharing [24] [25]. MACNET is inspired by their usage. In comparison, it is extensive as it promotes collaboration for a variety of applications. Anna-Kaisa Pietiläinen et al. [17] proposed MobiClique that forms and exploits ad hoc social networks to disseminate content using a store-carry-forward technique. The authors proposed a middleware to form an ad-hoc social network which can dissipate content without the need of a central server. The proposed work helped us identify critical parameters and challenges in MANETS.

Vimoware [6] is a toolkit that connects devices in ad hoc network. It follows a SOA (Service Oriented Architecture) and exposing service by URI. In comparison, the process of service discovery is automated such that there is no need to remember or share URIs between devices.

VI. CONCLUSIONS AND FUTURE DIRECTIONS

Smart phones possess extensive capabilities for massive deployment and usage. However, they exhibit limited resources which effect their capability for wide spread usage. This paper proposes collaboration-based ad-hoc networks to empower smart phones with effective utilization of resources, enriched user experience, and improved performance. The paper provides sample applications which can utilize collaborative computing for enhanced performance. MACNET can provide services either through the peer to peer mode or through the centralized mode. The former mode is resilient to user mobility as a number of peers are available for operation. However, in the latter mode, the operation of MACNET could be affected if a user, providing services to other peers becomes unavailable. The current version of MACNET ignores user churn under such a scenario.

MACNET is based on trust-based computing. The current implementation of MACNET provides limited security such that the communication can only be interpreted by MACNET-enabled phones. Other means of security such as encryption may be utilized with added cost. Similarly, trust-based analysis may also be done to assess strengths of application-specific scenarios. MACNET can facilitate collaborative computing for a wide variety of applications [26].

REFERENCES

- [1] R. Alarcon, L. Guerrero, S. Ochoa, J. Pino. "Analysis and design of mobile collaborative applications using contextual elements," *Computing and Informatics*. vol 22 no. 2, pp. 469-496 June 2006.
- [2] K. P. Birman, R. Friedman, M. Hayden, I. R. Imran. "Middleware support for distributed multimedia and collaborative computing," *Software: Practice and Experience* vol. 29, no. 14, pp. 1285-1312, December 1999.
- [3] M. Conti, M. Kumar. "Opportunities in Opportunistic computing," *IEEE Computer* vol 43 no. 1. pp 42-50, Jan 2010.
- [4] D. Huang, "Mobile cloud computing," *IEEE COMSOC Multimedia Communications Technical Committee (MMTC) E-Letter*, 2011.
- [5] Z. Jiang, S. Kim. "Trust Model for Mobile Devices in Ubiquitous Environment," In *Proceedings of the 1st international conference on Network-based information systems*, 2007.
- [6] L. Juszczyk, S. Bashir, A. Manzoor, S. Dustdar, H. Truong. "Vimoware - a Toolkit for Mobile Web Services and Collaborative Computing," In *Proceedings of Software Engineering and Advanced Applications*, SEAA 2008.
- [7] R. Kemp, N. Palmer, T. Kielmann, H. Bal. Cuckoo. "A Computation Offloading Framework for Smartphones," In *Proc. of MobiCASE Santa Clara, CA, USA*, 25-28 October 2010.
- [8] R. Kemp, N. Palmer, T. Kielmann, H. Bal. "Energy Efficient Information Monitoring. Applications on Smartphones through Communication Offloading," In *Proc. of MobiCASE Santa Monica, CA, USA*, October 24-27 2011.
- [9] R. Kemp, N. Palmer, T. Kielmann and H. Bal. "The Smartphone and the Cloud: Power to the User, (invited paper)" In *Proc. of MobiCloud, Santa Clara, CA, USA*, 28 October 2010.
- [10] R. Kemp, N. Palmer, T. Kielmann and H. Bal. "Opportunistic Communication for Multiplayer Mobile Gaming: Lessons Learned from PhotoShoot (demo paper)," In *Proc. of MobiOpp Pisa, Italy*, 22-23 February 2010.
- [11] D. Kovachev, Y. Cao and R. Klamma. "Mobile Cloud Computing: A Comparison of Application Models," *CoRR abs/1107.4940*: (2011).
- [12] K. Kumar, Y. Lu. "Cloud Computing for Mobile Users: can offloading computation save energy?," *IEEE Computer* vol 43 no 4, pp 51-56. April 2010.
- [13] D. G. Murray, E. Yoneki, J. Crowcroft, S. Hand. "The Case for Crowd Computing," in *Proc of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds*, 2010
- [14] M.T Nkosi, F. Mekuria. "Cloud Computing for Enhanced Mobile Health Applications," In *Proc. of the 2nd IEEE International Conference on Cloud Computing Technology and Science*, 2011.
- [15] N. Palmer, R. Kemp, T. Kielmann, H. Bal. "RAVEN: Using Smartphones. For Collaborative Disaster Data Collection," in *Proc of the 9th International ISCRAM Conference*. Vancouver, Canada, April 2012.
- [16] N. Palmer, R. Kemp, T. Kielmann, H. Bal. "The Case for Smartphones. As AN Urgent Computing Client Platform," in *Proc. Of the International Conference on Computational Science, ICCS 2012*.
- [17] A. Pietiläinen, E. Oliver, J. LeBrun, G. Varghese, C. Diot. "MobiClique: middleware for mobile social networking," in *Proc of the 2nd ACM workshop on Online social networks*, pp 49-54, 2009.
- [18] Z. Sanaei, S. Abolfazli, A. Gani, and R. Khokhar. "Tripod of requirements in horizontal heterogeneous mobile cloud computing," in *Proc. 1st Int'l Conf. Computing, Information Systems, and Communications*, 2012.
- [19] Z. Sanaei, S. Abolfazli, A. Gani, M. Shiraz. "SAMI: Service-Based Arbitrated Multi-Tier Infrastructure for Mobile Cloud Computing," in *Proc of MobiCC 2012: IEEE Workshop on Mobile Cloud Computing*, Beijing, China.
- [20] G. Scholten, N. Palmer, R. Kemp, T. Kielmann and H. Bal. "A Decentralized Decision Support System for Mobile Devices," in *Proc of 2nd International ICST Conference on Mobile Computing, Applications, and Services*, October 2010.
- [21] J. Scott et al. "Seamless Networking for Mobile Applications," In *Proc of the Ninth International Conference on Ubiquitous Computing 2007*.
- [22] C. Watkins, P. Dayan. "Q-learning," *Machine Learning* vol 8, issue 3, pp 279-292, 1992.
- [23] N. S. Fatima, P. S. A., Khader. "Efficient Data Accessibility using TTL Based Caching Strategy in Manet," *European Journal of Scientific Research*, vol. 69, no 1, pp 21-32, 2012.
- [24] Ghosekar, P., G. Katkar, and P. Ghorpade, P Ghosekar, G Katkar, DP Ghorpade. "Mobile ad hoc networking: imperatives and challenges," *IJCA Special Issue on "Mobile Adhoc Networks" MANETs no 3*, pp. 153-158, 2010
- [25] M. Denko, E. Shakshuki, H. Malik. "Cross-layer based Data Management in mobile ad hoc networks," in *proc of IEEE Global Telecommunications Conference 2009*. Honolulu, HI pp. 1-6.
- [26] Gabler, Jan, Ronny Klauck, Mario Pink, and Hartmut Konig. "uBeeMe—A platform to enable mobile collaborative applications." *Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom)*, 2013 9th International Conference on. IEEE, 2013.
- [27] Wu, Huijun, Dijiang Huang, and Samia Bouzefrane. "Making offloading decisions resistant to network unavailability for mobile cloud collaboration." *Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom)*, 2013 9th International Conference on. IEEE, 2013.