A Reliable Peer-to-Peer Protocol for Mobile Ad-Hoc Wireless Networks

M. A. Al-Mouhamed, I. A. Khan, and S. N. Firdous
Department of Computer Engineering
KFUPM
31261 Dhahran, K.S.A
mayez@kfupm.edu.sa, irfana@kfupm.edu.sa, snaeem@kfupm.edu.sa

Abstract— **Reliable, fast, and power aware communication is needed for Ad-Hoc wireless networks. Current techniques based on client-server and Publish/Subscribe communication models are not suitable in multi-robot systems and generally for mobile applications. For this we propose a reliable peer-to-peer protocol based on a UDP Broadcast and Token Passing (UBTP). The protocol is implemented on a WLAN using the Stargate embedded system. For this, a customized UDP protocol with an imperative Poll-based communication is proposed. The protocol is implemented using (1) a communication thread (TC) and (2) a processing thread (TP). A testbed system which allows modules to run TC and TP, in addition to the generation of broadcast request is presented. We used symmetric code in all nodes, Evaluation reports the distribution of auction completion times for peer-to-peer operations. The evaluation reveals: (1) response times are comparable to UBTP operated at head node, (2) improved degree of reliability as at most 2 steps are sufficient for auctioning seven nodes, (3) proved fairness, and (4) comparable power consumption to simple UBTP.**

Keywords- mobile ad-hoc networks, mobile robotics, peer-to-peer protocols, performance evaluation, wireless networks.

I. Introduction

Many mobile applications are in need of an Ad-Hoc Wireless Networking Communication Model that provides fast, reliable, and power aware features. Applications such as mobile robots playing soccer, an expedition of robots moving in a hostile area, or a team of rescue robots exploring a building after a disaster are just a few examples that need the above technology. The client/server model or the publish/subscribe model are not adequate for the above applications. There is need for reliable peer-to-peer model to handle the unpredictable need for communication in mobile systems.

In [1] a centralized peer-to-peer video streaming over hybrid wireless network, infrastructure network and the ad hoc network, is proposed to improve the performance of the video transport over wireless Internet. The video is encoded into multiple layers, and pre-stored in the server. When a mobile station (MS) requests a video from the server, it checks if any MS in the cell caches the video, e.g. super-peer. A super-peer transports the enhancement layers to the requesting MS over multiple paths via the ad hoc mode, while the server transports the base layer to the requesting-peer via the WLAN mode. The base layer of the video is transported from the server via the WLAN mode, which benefits the centralized management of the video distribution, while the enhancement layers are delivered over the multiple paths via the ad hoc mode, which can reduce the congestion in the access point (AP). AP also manages the cached videos and the MSs. The advantages of this proposed scheme are: First, it sustains a centralized management of the content distribution. Second, it reduces the traffic contention in the AP, hence increasing the system throughput and decreasing the packet loss and delay in wireless network. (NS2). The simulation (NS2) results show that the proposed scheme can achieve a better perceptual video quality compared to the WLAN deployment.

In [2] an auction based communication model using (1) TCP, (2) UDP, and (3) UDP Broadcast with Token Passing (UBTP) scheme. In TCP, the head node communicates with each node included in the auction using TCP packets. In UDP the head node sends UDP packets to each node included in the auction. In UBTP, the head node broadcasts a UDP packet to all nodes which includes the node ID of the auction nodes and a sequence of nodes reply. The first node replies to the head node and at the same time initiates a token packet which is forwarded to the second node in the sequence, and so on. If any node fails to forwards the token then the next node automatically reply to the head node after a time-out of T milliseconds. The head node concludes the auction if it receives replies from all nodes otherwise it will broadcast a new auction request including the IDs of the nodes from which no bids were received so far. The process is repeated until all nodes responded or the originator gives up. However, UBTP is not a P2P as all auctions were launched by one node.

In [3] a Dynamic Token Ring based MAC protocol (DRP) for mobile ad-hoc networks is proposed. DRP solves the intra-flow and inter-flow contention problems using a ring-based protocol. Among all nodes in a cluster, only the node holding token is responsible for sending packets to the successor. This technique is dynamic, because the token ring in a cluster can be a low/high priority token ring. A receiver busy tone (BT) and a token tone (TT) out of band are used. The token tone is used to request updating the token ring in a cluster. Once the node gets the token, it needs to detect the existence of the BT or TT. BT may come from other cluster members, while TT usually comes from inner members who want to join the high priority ring. BT is also used by cluster head to suppress the multiple reporting of the loss of token. Evaluation shows the throughput and average end-to-end delay of the DRP protocol as compared to the IEEE 802.11 protocol.

JXTA is a peer-to-peer technology that enables developers to in the design of distributed computing software. In JXTA is applied to mobile-networked systems. Since mobile devices like PDAs, mobile phones, or laptop computers are much more likely to interoperate with each other in the absence of a coordinating authority such as a server, there is need for a technology above the hardware abstraction level of IrDA and Bluetooth. JXTA enables mobile to share data and to use functions of their respective peers. JXTA components are:

1. The Peer Resolver Protocol (PRP) is the mechanism by which a peer can send a generic query to other peers.
2. The Peer Discovery Protocol (PDP) is used to discover any published resources which are mandatory represented as advertisements.
3. The Pipe Binding Protocol (PBP) is used to establish pipe connections between peers. The Peer Information Protocol (PIP) is used to exchange status information between peers.
4. The Peer Membership Protocol (PMP) is the mechanism by which peers can organize themselves to form groups.

This paper discusses how JXTA works with mobile devices as a peer-to-peer solution. JXTA eases software development and provide platform independence and standardized protocols, and has built-in security features. However, JXTA is not lightweight as it needs hardware resources which might not be available on mobile devices.

In this paper we present a reliable Peer-to-Peer protocol (RPTP) for WLAN based on a customized UDP protocol including an imperative Poll-based communication (like in SNMP). Each peer node run two threads (1) a communication thread (TC), and (2) a processing thread (TP). Both threads are described in details. We also developed a testbed system which allows modules to run as TC and TP, in addition to the generation of broadcast requests within TP. Each module records packet identifiers for each arriving packet as part of the TC thread in addition to the total time needed to complete a customized broadcast with some acknowledgement. We presents statistical data for assessing the degree of reliability of the proposed protocol altogether with overall auctioning times.

This paper is organized as follows. In section 2 we present some well know wireless communication models. In section 3 we present the UBTP protocol which will be extended to a peer-to-peer protocol in Section 4. In section 5 we present the evaluation. We conclude in Section 6.

II. Wireless Network Models

The wireless network can be developed using different networking models. In the following three network models are discussed.

1) Client/Server Model [5][6]: In this network model, one node in the network is assigned as a server and other nodes as clients. The server generally performs the majority of the processing tasks. The clients initiate a connection with the server when they want to transfer instructions or data with the server. This model is not suitable for our Stargate based network because (1) in the network all nodes have equal computational power, therefore heavy computational load on one computer causes delays in the whole network, and (2) in the applications of the auction schemes any node can initiate communication (auction) with any number of the remaining nodes, so each node should have both client and server capabilities at the same time.

2) Publish/Subscribe Model [7][8]: This model consists of publishers and subscribers. The publisher is unaware of the recipients of its messages and rather it publishes messages to a class of subscribers. The subscribers can receive messages from the classes in which they are interested without any knowledge about the publisher. In this way the publisher and subscriber are decoupled in this model. While this model looks suitable for the auction schemes, it has many extra features that are not needed by the auction schemes. First and foremost problem is that the target nodes for each auction are known to the initiator, and once an auction is done the list of nodes participated in the last auction becomes unimportant for other nodes because successive auctions cannot always be interrelated. Secondly, the target nodes for any auction are determined dynamically by the initiator based on the application requirements. Therefore, classes in this model need to be changes dynamically for each auction or too many classes need to be formed to meet requirements of each auction. That is an unnecessary burden for short communications like used in the auction schemes.

3) Peer/Peer Model: In this network model, each node can connect to any other one or group of nodes and send/receive the data. The connections are ad-hoc and they lasts until the initiators or any other nodes want to terminate the connection. Features can be added into the basic Peer/Peer model to add functionality needed by the application. Peer/Peer model can use both TCP and UDP protocols. The following features of the auction schemes are best implemented in this network model: (1) The communication among nodes the auction is very short, (2) The behavior of each node is controlled by the node itself so there is no need for continuous data transfer, (3) Many features like broadcast or multicast can be used in Peer/Peer networking, and (4) Any

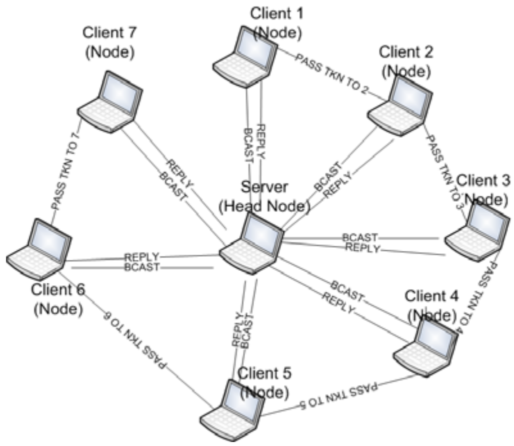

Fig 1: UBTP Broadcast and token passing scheme

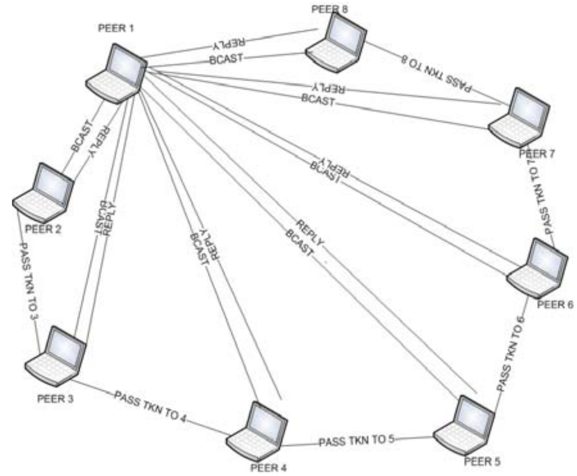

Fig 2: Peer to Peer Auctioning using UBTP

node can initiates communication with any other node or group of nodes using multicasting.

III. UDP Broadcast and Token Passing

In UDP Broadcast and Token Passing (UBTP) [2] Scheme there is one Head node and others are receiving nodes. The Head node broadcast UDP packet and the order in which the nodes must respond to all nodes in the network. Each node replies to the head node and sends a token packet to the next node which is next to it in the order. If the next node does not receive the token it will reply to the Head node after time out of T milliseconds. If the Head node does not receive ACK from any one of the nodes it will do a second broadcast to the selected nodes as shown on Fig 1.

IV. Peer to Peer Auctioning using UBTP

In Peer to Peer Auctioning using UBTP Scheme each node acts like a Peer. Any one of the Peer broadcast UDP packet and the order in which the nodes must respond to all Peer nodes in the network. Each Peer node replies to the Peer node which starts the Auction and pass a token to the next node which is next to it in order. If the next node does not receive the token it will reply to the Peer which has started Auction after time out of T milliseconds. If the Peer node which has started Auction does not receive ACK from any one of the nodes it will do a second broadcast to the selected nodes. We installed the Symmetric java program into each Peer, and this symmetric code is divided into four class files:

- Stargate (starts the Program)
- Receiver(Receives packets, sends token)
- Processing(Controls Auction, Updates the file)
- Sender(Sends packets)

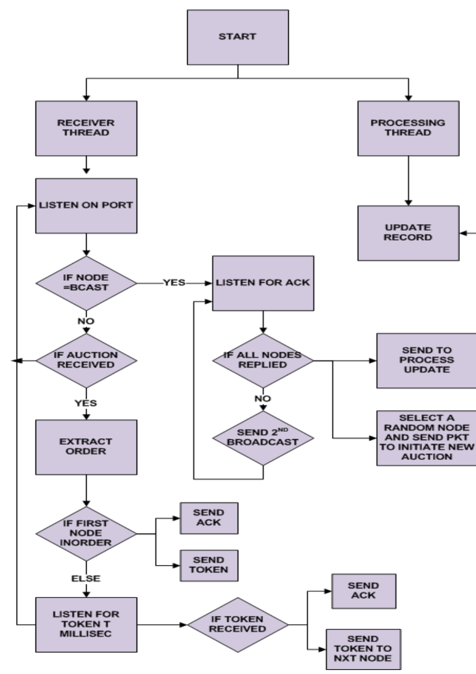

Fig 3: Flow chart of the node program

The main parts of the program are: (1) Main Component, (2) Receiver, (3) Processing, and (4) Sender. The main Program initializes the program and creates two threads Receiver thread and processing thread. The function of the receiver and processing threads is shown in the above flow chart (Fig 2).

The processing thread is used to send the auction and the receiver thread starts listening on the ports for incoming packets. Once the receiver thread receives the packet it checks for message field and decides whether it is an auction packet or acknowledgement packet. If the node receives auctions then Send Acknowledgment to auctioning node Send token to next node. If the node receives auctions then Send Acknowledgment to auctioning node Send token to next node. It also calculates the response time of each node which sent the acknowledgement.

The Receiver Thread performs the following steps:

- Listen on the Port: When the Main program initializes the Receiver Thread, it starts listening on the port for the UDP packet; once the packet is received it goes to the next step.
- Extract the IP address: Once the packet is received the Receiver Thread extracts the IP address of the Sender node from the received UDP packet, then it will do the following tasks:

    - If the Sender node is itself i.e., auctioning node then go to the listening mode and listen for Acknowledgments from all the other nodes. If all nodes replied with the Ack then update the record i.e. write the time taken to complete an auction into the file. If any one of the node doesn't reply within the specified time then broadcast the same packet second time to the specified nodes that has not replied with Ack's.

    - If the node receives an auction i.e., not an auctioning node then extract the order from the packet in which the node must send an Ack to the Sender node , if it is the first in order then immediately Send an Ack to the auctioning node and Send a token to next node which is next to it in order . If it is not the first node in order then listen for the token for Tms. If the token received within Tms then Send Ack to the auctioning node and Send a token to the next node which is next to it in order. If token is not received within Tms i.e. after Tms Send Ack to the auctioning node and Send a token to the next node which is next to it in order.

- Next Auction: After all nodes replied with Ack then send an initiation packet to a randomly selected node through the Sender Module.

The Process Thread performs the following steps:

- Generate Auction: The Process Thread generates a sequence or the order in which all nodes should reply and added it to the UDP packet and Sends an Auction through Sender Module.
- Update Record Method: This method maintains a file related to the completion times of the Auctions. Whenever all the nodes reply with Ack's, the time taken to complete an auction is recorded in the file i.e. the time from the start of the Auction to the time till all nodes replied with Ack's.
- Process the Auction Request: Whenever the Receiver Thread receives an initiation packet by selecting a random node; the Process Thread is restarted again.

The Sender Module performs the following steps:

- Sends the Auction or the initiation packet: The main purpose of this method is to send an auction started by the Process Thread and it is also used to send an initiation packet to the randomly selected node.
- Sends Ack: This method is also invoked whenever a node wants to send an Ack to the auctioning node.
- Mostly Inactive: Most of the time this method is inactive as this method is invoked only when there is something to be sent.

V. EVALUATION

The hardware consists of Stargate boards with wireless networking cards. The Stargate board is a powerful single board computer that consists of Intel 32-bit, 400 MHz XScale processor and 96 MB of memory in terms of SDRAM and Flash. The Stargate also have a daughter board that contains socket for the wireless card and Ethernet interface. The software of the Stargate comprises of Linux OS with drivers for all peripherals and Java Runtime Environment (JRE). In our experiments each Stargate have an Ambicom IEEE 802.11b wireless card. The wireless card has an additional 64 MB of memory for storing drivers and program files.

We used 7 Stargate Boards to implement the peer to peer protocol. The Stargate boards are configured to form a WLAN network in which each node (Stargate Board) acts like a Peer. All the seven nodes form a Peer to Peer wireless ad-hoc network. We installed the Symmetric java program into each Peer using HyperTerminal.

The Stargate system is assumed to be one single collision domain. Auctions are being generated by separate nodes and transmitted to all the other nodes.

Here we present the performance plots of the peer-to-peer UBTP protocol which derive from the measurement data out of the use of seven Stargate nodes. Fig 4 presents the histogram distribution of auction overall completion times. Each node had the opportunity to generate 50 auctions, e.g. a total of 350 auctions. The plot shows the percentage of auctions that falls into the reported time (ms). For each auction time (t) the plot displays a set of columns (1 to 7), each corresponds to the percentage of cases for which the auctioning time was measured as t. It is clear that in the large

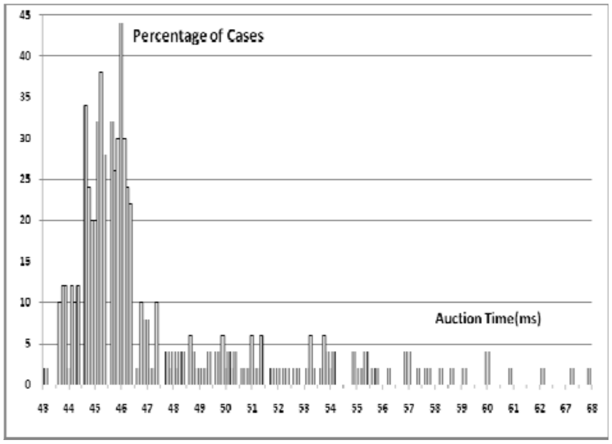

Fig 4: Percentage distribution of auction completion times for seven nodes for the Peer to Peer Auctioning using UBTP.

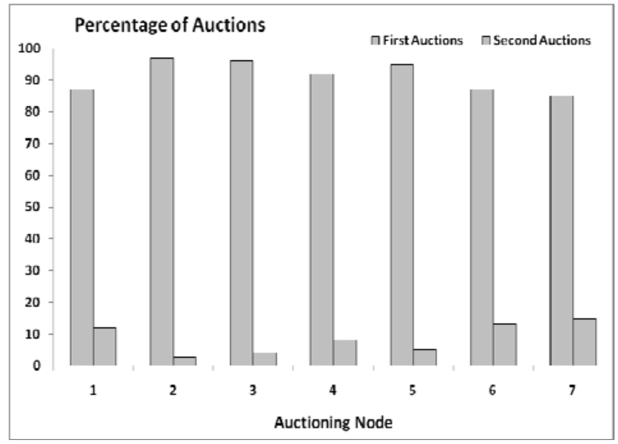

Fig 5: All auctions were completed in no more than two auction broadcasts: percentage of auction completed in first and second steps for seven nodes.

majority of experimented cases the average auction times fall below 50 ms. The distribution also shows some scattering within the range [43, 68] ms with some concentration over the range [44, 47] ms. The average auctioning time is 51 ms that is slightly above that achieved by UBTP [2] in which the auctions were started from one singe node.

The statistical data and plots (Fig 4 and 5) for assessing the degree of reliability of the proposed protocol altogether with overall auctioning times show that peer-to-peer UBTP appears to be reliable as all experienced auctions among 7 stargate nodes have been completed using only two auctioning steps. For all nodes, the first auctioning step successfully completed the auction in more than 80% of the tested cases and in less than 20% of the cases a second auctioning step was needed. In no experienced case, the proposed protocol had to carry out 3 or more auctioning steps. The above analysis and plots presents some useful insight on the reliability aspects of proposed peer-to-peer UBTP.

We note the symmetry across the nodes when each is running the peer-to-peer UBTP protocol. This is evidenced by: (1) the distribution of auctioning times for each node (Fig 4), and (2) the percentage of auction completed in the first and second steps of the protocols (Fig 5). The nodes exhibited a quite symmetric performance pattern with respect to distribution of auction times and success rates through the two steps which is an indicator of fairness for the proposed protocol.

Direct measurements of the current during the communication phase, the computation phase, and idles phase of Stagate module gives 0.37 mA, 0.46 mA, and 0.52 mA, respectively. Table I summarizes the average completion times, the bound for 95% of distribution of auction times, and the power consumption for auction and peer-to-peer communications.

Power taken by each scheme	**Average time taken by one auction in milliseconds**	**power (using current x current x time) mW**
UDP Broadcast tokens based scheme	37.13	7.86
UDP point to point scheme	44.67	9.45
TCP point to point scheme	180.95	38.29
UDP P2P Broadcast with Token passing	57.03	12.07
UDP P2P Distributed Broadcast	51.96	11

Table I: Summary of average auction completion times and corresponding power consumption.

VI. CONCLUSION

In this paper we presented the design of a reliable Peer-to-Peer UBTP protocol for WLAN. It uses an imperative Poll-based communication like in SNMP. Each peer node run two threads (1) a communication thread (TC), and (2) a processing thread (TP). We described how the protocol operates within each thread. We presented statistical data for assessing the degree of reliability of the proposed protocol altogether with overall auctioning times. Peer-to-Peer UBTP appears to be reliable as all experienced auctions among 7 stargate nodes have been completed using only two auctioning steps. The first auctioning step was successfully covered all the node in more than 80% of the test cases and in less than 20% a second auctioning step was needed. The general observations from the evaluation are: (1) symmetric code in all nodes, (2) response times are comparable to UBTP auction scheme operated at head node, (3) improved degree of reliability (at most 2 steps for seven nodes), and (4) comparable power consumption to simple UBTP. Note that the response time and reliability can also be measured for indoor and outdoor environments.

ACKNOWLEDGMENT

The authors would like to thank King Fahd University of Petroleum and Minerals for the computing support.

REFERENCES

[1] Al-Mouhamed, M.A.; Siddiq, U.F., Performance evaluation of auctions WLAN for RoboCup multi-robot cooperation," IEEE/ACS International Conference on Computer Systems and Applications, 2009. AICCSA 2009., pp.610-615.

[2] Yifeng He; Ivan Lee; Xijia Gu; Ling Guan, Centralized Peer-to-Peer Video Streaming Over Hybrid Wireless Network," IEEE International Conference on Multimedia, ICME 2005, pp.550-553.

[3] Lu, X., Fan, G., and Hao, R, A dynamic token passing MAC protocol for mobile ad hoc networks". Proceedings of the 2006 international Conference on Wireless Communications and Mobile Computing, IWCMC 2006. ACM, pp. 743-748.

[4] N. Maibaum and T. Mundt, JXTA: a technology facilitating mobile peer-to-peer networks, International Workshop in Mobility and Wireless Access, MobiWac 2002, pp. 7-13.

[5] LaMaire, R.O.; Krishna, A.; Ahmadi, H.;Analysis of a wireless MAC protocol with client-server traffic and capture, IEEE Journal on Selected Areas in Communications, Vol. 12 , Issue: 8, 1994, pp: 1299 – 1313.

[6] Chen, I.-R.; Baoshan Gu; George, S.E.; Sheng-Tzong Cheng; On failure recoverability of client-server applications in mobile wireless environments, IEEE Transactions on Reliability, Vol. 54(1), 2005 , pp: 115 – 122.

[7] Jeppe Bronsted; Klaus Marius Hansen; Rolf Thorup; A Uniform Publish-Subscribe Infrastructure for Communication in Wireless Mobile Environments, 6th Inter. Conf. on ITS Telecommunications Proceedings, 2006, pp: 1145 – 1150.

[8] Wen-Jun Yang; Bing-Wen Wang; Wei Li; Xiao-Ya Hu; Decomposition of Subscription Based Publish-Subscribe Middleware for Wireless Sensor Networks, Inter. Workshop on Intelligent Systems and Applications, ISA 2009, pp: 1 – 4.