SUPPORT REAL-TIME INTERACTIVE SESSION APPLICATIONS OVER A TACTICAL MOBILE AD HOC NETWORK

Li Li and Louise Lamont
Communications Research Centre, Industry Canada
Ottawa, Canada

ABSTRACT

This paper presents the service support mechanism for a high capacity Tactical Mobile Ad hoc Network (TacMAN) that aims to offer reliable realtime session applications. Using the Internet Engineering Task Force's (IETF) standard Session Initiation Protocol (SIP), TacMAN delivers integrated real-time multimedia session and data services over a distributed application platform. In TacMAN, SIP user agents are automatically discovered and configured applying an efficient service discovery mechanism that leverages on the cross-layer design between the application and the proactive routing layer. Such an auto-discovery capability extends to track the presence and loss of user agents in the network, enabling fast re-grouping and re-establishment of session communications in the dynamic ad hoc environment. The paper presents the application architecture of TacMAN and evaluates its performance characteristics, demonstrating the efficiency of the approach, and illustrating network design guidance for supporting SIP applications over the tactical MANET.

INTRODUCTION

Support of robust real-time interactive applications is often required in the tactical Mobile Ad hoc Networks (MANET). The voice and video conference calls, push-to-talk, and peer-to-peer conversation sessions are just a few of the examples. TacMAN is a high capacity Tactical Mobile Ad hoc Network that aims to deliver reliable and integrated multimedia services, including the above mentioned realtime interactive session applications and data transmission services.

The standard protocol published by IETF for setting up multimedia session applications is the Session Initiation Protocol (SIP) [5]. To facilitate service interoperability, SIP is selected as the application protocol for TacMAN. However, similar to many of the popular applications developed in fixed networks adopting the client-server architecture, SIP requires special design considerations when it is applied in MANET to offer realtime interactive sessions. Firstly, because it is ad hoc and infrastructure-less, a MANET may not consist of the centralized servers that are employed in implementing SIP applications in the fixed networks. Secondly, the dynamics of MANET often give rise to issues of network topology change or even network partitioning that debilitate the performance of the applications, which are not encountered in the fixed networks. In TacMAN, a fully distributed SIP architecture is implemented based on a novel service discovery mechanism that automatically discovers and configures the SIP end points, i.e., user agents, for session establishments. Over this application infrastructure, TacMAN delivers conference calls, peer-to-peer conversations and push-to-talk applications without requiring any centralized servers or manual configuration of service locations. Applying cross-layer design between the application and a proactive routing layer, the service discovery capability extends to track the presence of user agents in the network, to identify the arrival and leave of user agents, and to promptly regroup and reestablish interactive session communications under the adversary situation of node loss and user group change.

This paper presents the application architecture solution of TacMAN, focused on its service locating capability for offering reliable realtime interactive session applications. We first review and compare architecture options for supporting SIP session applications in the MANET. Then we present the application architecture devised for TacMAN, which is cross-layered over TacMAN's routing platform to offer a fully distributed SIP service infrastructure with enhanced reliability in the mobile ad hoc environment. Lastly, the performance simulation results are presented, providing design guidance for building the integrated data and multimedia session applications in a tactical MANET.

The rest of the paper is organized as follows: Section 2 gives an overview of SIP application architecture options and the related work; Section 3 presents the SIP application architecture and service support schemes of TacMAN; Section 4 illustrates and analyzes the results of the performance simulations; Section 5 concludes the paper.

I. OVERVIEW AND RELATED WORK

A. SIP MANET Architecture Overview

TacMAN selects SIP as its realtime session application protocol. SIP communications apply the client-server architecture [5]. A SIP endpoint functions alternately as a User Agent Client (UAC), or as a User Agent Server (UAS). During a SIP session, the SIP UA of the caller functions as the UAC, while the callee's UA acts as a UAS to accept the session request from the UAC. SIP application uses SIP URL (Uniform Resource Locator), e.g., sip:bob_fisher@rescue_canada.ca, which is the service Address of Record (AOR) to identify the destination UAS of the requested session [5]. In the fixed-IP network, to map the service AOR of a SIP URL to its current IP address, certain centralized servers are employed, including the proxy server [5], the DNS server [6], the DHCP server [7], the ENUM server [11], etc.

In a MANET, the above-mentioned centralized servers may not exist. Mobility and topology dynamics of a MANET instigate additional complications even if these servers are to be applied. In TacMAN, a fully distributed proxy-less SIP architecture is favored, without requiring the constant availability of any centralized server. The proxy-less model improves application reliability for TacMAN in its mobile ad hoc environment. However, without the assistance of the proxy and other servers for location mapping, the IP address of the called party needs to be discovered before the caller can send a request to the called party/parties for session establishment. One might think to manually configure the IP addresses for the destinations. Though knowing someone's SIP URL or phone number which is one type of the standard AOR [5] is reasonable, knowing and manually configuring IP address of each called party in a dynamic ad hoc environment of TacMAN cannot be assumed.

B. Related Work

Support of realtime interactive session applications in MANET has not been previously well addressed in literatures. Recently, the issue of service discovery under MANET has been examined and a number of solutions have been proposed [1][2][8][9][12][14]. Most of the solutions attempt to integrate or embed the service discovery messages into routing protocols to minimize the protocol overhead in the bandwidth constrained MANET environment [2][8][9][12][14].

Among the service discovery solutions, one approach [1] forms a distributed service directory within MANET to handle service query-and-response. This scheme though offering a generic architecture approach, requires additional algorithms and processing phases which may not be appreciated in the resource constraint MANET. Another approach piggybacks the query-and-response messages onto the MANET reactive routing protocols [2][9][12], with no service directory assumed to improve the protocol efficiency. And the third approach [14] piggybacks the service discovery messages onto a MANET proactive routing protocol.

To reduce the session-setup latency and real-time data transmission delay - factors which are critical to the performance of real-time services, in TacMAN, the proactive routing protocol of OLSR (Optimized Link State Routing Protocol) [3] with enhanced scalability using routing hierarchies [16][17] is adopted instead of a reactive protocol. When topology changes in a MANET are not extremely frequent, a proactive routing protocol may be favored in reducing the real-time data transmission delay [4]. We apply the service discovery solution integrated with the OLSR protocol as that was proposed in [14] but with enhanced schemes for supporting SIP in TacMAN. Using cross-layer design between the application and the routing layers, the topology information acquired by the proactive routing layer as well as its efficient message flooding mechanism [13] are employed to maximize the protocol efficiency. We also further extend the service discovery capability to track the loss of the SIP agent and arrival of the new agent for fast regrouping of sessions. Though SIP protocol has the event and notify method package [15] that can be used to track the presence and status of a node, the methods can only be applied when the IP addresses of the nodes are known. Mapping between the IP address and SIP URL/phone number, the SIP service platform of TacMAN tracks in realtime the presence of service agents without requiring any pre-configurations of the node IP addresses to greatly improve the situation awareness in a fully distributed tactical ad hoc environment.

II. SUPPORT OF REALTIME SESSION SERVICES

Figure 1 illustrates the cross-layered architecture of TacMAN, including the application and routing components. The service discovery component locates and tracks service nodes for applications. The service discovery component utilizes the topology information gathered from the routing layer to facilitate the service tracking at the application layer, and reuses the messaging infrastructure of the routing protocol to reduce the processing overhead.

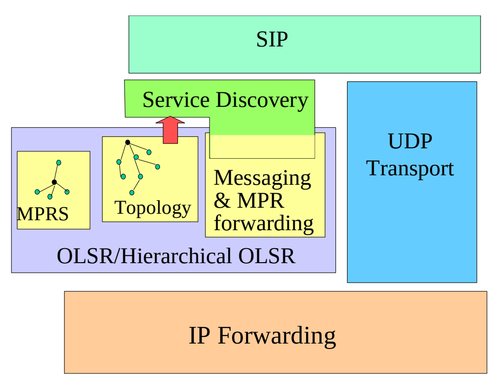

Figure 1. TacMAN Application Cross-Layering Architecture

A. Locating SIP Agents

As discussed in the previous section, the IP addresses of the destination SIP User Agent(s) (UA) need to be known before the caller can initiate a session setup request. In each of the SIP UA (e.g., laptop or PDA in TacMAN) node, the Service Discovery (SD) component automatically discovers the IP address(s) for each of the SIP AOR (a SIP URL or a phone number) that the user wants to contact. To reduce the infrastructure cost and message-processing overhead, the SD component leverages on the routing protocol, particularly applying the MPR mechanism of OLSR [3] for SD messaging.

The design principle of the OLSR protocol facilitates extension of the protocol functionality through the addition of new message types [3]. With its default message processing and forwarding mechanism, OLSR provides the backward compatibility enabling nodes to properly handle all messages, regardless of whether a particular message type is recognized. A new message type, Service Location Extension (SLE), is introduced into OLSR to implement the service location discovery. The SLE message is used to query about servers, to respond to queries and to advertise service location information. To reduce the message overhead, the time interval for advertising SLE can be increased to suppress the advertisement when the query capability is applied.

The query and advertise SLE messages are disseminated to all the nodes using MPR forwarding, same as other OLSR messages. Response SLE messages can be sent using unicast or using MPR forwarding. In the latter case, all the nodes can hear and can cache the mapping information for later use. In some instances this response strategy can be excessive, but in other cases it reduces redundant queries for the same information, which would otherwise be generated by some other nodes. In fact, if K nodes are seeking a same server using MPR forwarding, one query message and one response message are forwarded throughout the network. If a unicast response is sent instead, K query messages might be forwarded by the MPR nodes throughout the network plus the response messages. When $K>1$, sending unicast response messages may then entail more overhead than sending them through the MPR forwarding.

B. Push and Pull SLE schemes

In TacMAN, we support both the push scheme of service discovery using advertising and the pull scheme using query-response. The former is simple and the latter improves the response time for accessing the service. The push solution may be suitable for certain scenarios where a small number of server(s) are needed by most of the nodes and where the service information may change frequently. The pull scheme is often more efficient when individual node requires different servers.

Assume the same cost for sending query, response and advertise SLE messages. That is, they are all MPR forwarded. Applying a simple push scheme with an advertising interval of τ, the SLE message is generated at an average rate of N/τ, where N is the number of SIP nodes in the network. When N is small, e.g., all the SIP agents would call into one conference URL, the push scheme may be a simple option. However, the overhead of the push scheme increases when N scales up.

The pull scheme can be applied in various ways. We describe here two variations, the "pull-in-advance" and the "push-and-pull" approaches. In the "pull-in-advance" scheme, upon arriving/restarting at TacMAN, a new SIP UA is to send out a query to locate all the UAs in its contact list. Assume a Poisson distribution with rate λ of generating such a query by the arrival of new nodes in the network. A query may also be originated in the network by a node n if a UA contacted by node n using its current known location information does not respond; or if the node n wants to contact a UA that was not located before. Assume that this type of query is generated in the network with a Poisson rate δ. Then we denote that the "pull-in-advance" scheme originates in average a message rate of about $(x+1)\lambda + 2\delta$, assuming that a query message of the first type elicits x response messages and a query of the second type one response. Thus, when the number of SIP UA nodes increases to $N > \tau((x+1)\lambda + 2\delta)$, the "pull-in-advance" scheme generates less message overhead than the push option. If the node increase in TacMAN is modeled using a Poisson rate μ, then at $t > (\tau((x+1)\lambda + 2\delta) - N_0)/\mu$ where N_0 is the initial number of nodes in the network, the average message rate of the push scheme may exceed the pull-in-advance scheme.

In the "push-and-pull" case, a new node sends out an advertise SLE announcing itself upon arriving/restarting at the network, assumed having the Poisson rate λ. A node only sends out a query message when it wants to contact a SIP UA, whose location information has not been heard of. We denote the rate for sending such a query message as ϕ, which is often much smaller than the inter-arrival rate of incoming calls. This is because that often a query is only sent to ask for a SIP node when the caller contacts the node for the first time. The SIP UA nodes also cache the location information heard in the advertising SLE messages as well as the response SLE messages to queries sent by other nodes. Then the total SLE message-generating rate is denoted as $\lambda + 2\phi$, assuming a query message eliciting one response message in this case. When $\lambda > 2(\phi - \delta)/x$, the push-and-pull scheme may generate less message than the pull-in-advance scheme. The efficiencies of the above two pull schemes depend on the events rated by λ, ϕ and δ. If the contact list is small and is fairly fixed at each agent node (i.e., each caller agent knows in advance whom he will be communicating with when starting in the network), x can be limited and δ reduced (tends to be much smaller than ϕ) for an efficient pull-in-advance scheme. On the other hand, with large values of λ and x, the pull-in-advance strategy may result in more messaging overhead.

III. TRACKING SIP AGENTS IN GROUP COMMUNICATIONS

In TacMAN, a group of SIP UAs often have a push-to-talk group session using their SIP devices to keep in contact. The commander of the group needs to be aware of the presence of the nodes in the current network and decide then if they should be invited into the group push-to-talk session. The arrival of the new node and loss of an existing node should be alerted to the commander in a timely fashion.

TacMAN achieves node tracking through the cross-layer information shared between the routing layer and the application layer. The proactive routing layer can identify the removal of a node, and the arrival of a new node from its topology view of the network. When a SIP agent *n* drifts away from the current network, routing layers of other nodes would find out that node *n* is removed from the topology table, and would inform the application layer the loss of node *n*. As TacMAN applies the OLSR routing in a scalable hierarchical approach, the topology table at a node only holds the topology information of the local cluster of the network, not that of the entire network [17]. Upon being informed that node *n* is removed from the topology table, the service discovery component of the commander can launch a query SLE asking for node *n*'s SIP AOR. A response to the query would confirm that *n* is still reachable in the network, as well as its IP address. Otherwise, the node is not present in the network, either the local cluster or the entire network anymore.

When a node moves into the network or restarts in the network, the cluster/network change indication from the routing layer prompts the service discovery function to send an advertising SLE reporting itself, its SIP AOR and IP address. The commander in the current network can then contact the new node immediately if wanted. Alternatively, when a new node appears in the topology table, the service discovery component of a commander node may send out a unicast query SLE for an inverse request on the SIP AOR given the IP address. This would let the commander notice a new node entering the network without using bandwidth to set up an actual communication session with the node.

The issue of security and protection of location information is another key area in TacMAN and beyond the scope of this paper, and is to be published in a separate paper.

IV. PERFORMANCE STUDIES

The TacMAN testbed has been implemented on the Linux operating system for field test of the supported features. To evaluate the performance behavior of the realtime session services, an OPNET simulator of TacMAN is also implemented.

In the simulations, each mobile node modifies its location within the subnet based on the “random waypoint” model [10] at an average speed of 3km/s with pause intervals. The voice data makes use of the G.723 codec, generating a data rate of 5.3Kbps. The OLSR timer intervals for the Hello and TC messages are assigned default values in accordance with RFC 3626 [3]. The 802.11 WLAN (Wireless LAN) interface is set to a speed of 1Mbps. The WLAN transmission range is approximately 250 meters.

A. Service Discovery Scheme Studies

In this set of scenarios, a dynamic network where nodes are constantly entering the network is modeled to evaluate the impact of the service discovery scheme on the service performance. Most often, new nodes entering the network would need to query and discover about services. Therefore, we simulate dynamic network scenarios with nodes entering the network that generate the service discovery traffic.

The simulated network is comprised of 51 nodes with a size of $1.2 \times 1.2\ \mathrm{km}^2$. Of those, 40 are SIP UA nodes. During the simulations, 31 of the 51 nodes are assumed to reside within the network, with another 20 SIP nodes coming into the network at an average of every 360 seconds. Only these 20 SIP nodes initiate the SIP session setup requests. The “push” and “pull” in the labels used in the table and diagrams below represent if the “push” scheme or the “pull” scheme is used in the service discovery process. We applied an “aggressive” pull scheme in the simulations, which is a combination of the “pull-in-advance” and “push-and-pull” schemes. This means that a node would advertise itself upon arriving/starting in the network or upon incurring changes on its location information. A node also sends out a query asking for all the nodes on its contact list upon arriving/starting in the network. The contact list remains the same during the simulation. Thus unless a node is partitioned from the network, no query is sent again later to locate it. The nodes are entering the network in groups of two or four nodes, represented using the “_2” and “_4” in the labels respectively. For example, in the "pull_2" case, two nodes enter the network after approximately 360 seconds; after approximately 720 seconds, another two nodes arrive, etc.

The message overhead of the service discovery scheme was very insignificant as illustrated in [14]. Here we are focused on the studies of the SIP application performance. The following traffic profile is set in the simulated network. At each SIP user agent, the SIP session requests generated follow an exponentially distributed inter-arrival time with a mean value of 360 s. The military session conversation is concise in style and its duration is modeled as exponentially distributed with an average of 60 s.

The session throughput measurements as illustrated in Figure 2 indicate that the pull scheme may outperform the push scheme in rendering higher session throughputs under the same session request generation profile. The push scheme causes in average more waiting time for the destination UAS's location information before a UAC can launch a session request.

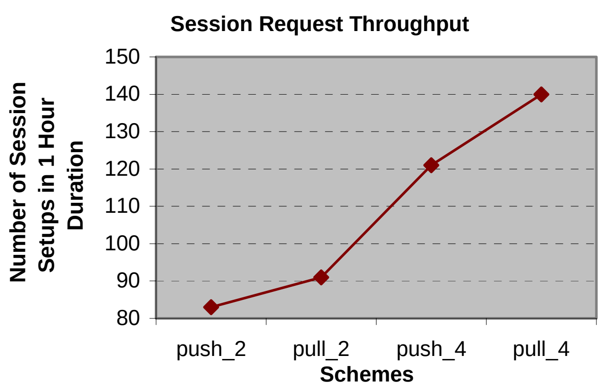

Figure 2 Session Throughput Measurements

The session performance metrics, including the average session setup delay, the percentile of sessions that have seen their setup delay less than 5 s, the percentile of voice packets that have experienced an end-to-end delay of less than 200 ms, and the average voice packet delivery ratio are presented in Table 1 below.

	Push_2	Pull_2	Push_4	Pull_4
Session setup delay	0.323 s	0.436 s	1.245 s	0.814 s
Percentile of setup delay < 5 s	94.19%	87.91%	82.44%	87.9%
Percentile of pk end-to-end delay < 200 ms	100%	100%	98.00%	99.03%
Voice pk delivery ratio	94.26%	94.1%	92.36%	93.24%

Table 1. Voice Performance Metrics Comparison

When more sessions have been established in the network, the setup delay is expected to increase. However, if more nodes are entering the network faster, e.g., in the case of push_4 and pull_4, the scenarios running the pull scheme may experience lower setup latency. Waiting for the destination UAS's location information may synchronize the session request attempts from different UACs when they hear about the UAS at almost the same time. Concentrated traffic load can result in increased session setup latency and higher packet delivery loss. In general, increased network traffic load degrades the voice packet delivery performance, e.g., the packet delivery ratio. However, when groups of UACs are making session requests at almost the same time as that in the push_4 case, concentrated traffic may affect the packet end-to-end delay adversely.

Enabling the tracking capability provided by the service discovery component, a new node arriving in a network can be identified by the commander node of the network and thus can get connected immediately into the push-to-talk group session if needed. Figure 3 depicts the CDF (Cumulative Distribution Function) of the time delay for a newly arrived node to be connected into the group push-to-talk session in the network when the tracking capability is enabled. The plot in Figure.3 does not capture the time needed for the update of the topology database upon the arrival of the new node, which may amount to a few seconds for the OLSR to establish the neighborhood information with the neighboring nodes. When a new node can be discovered and connected immediately upon its arrival into a network, the commander's situation awareness of the member nodes in the network is greatly improved. Note that Figure 3 includes the session setup time. The commander actually discovers the new arrival node, its SIP URL/phone number (e.g., displayed on the screen) and other information with even a shorter time delay, before the session is set up, or without the need to set up any session.

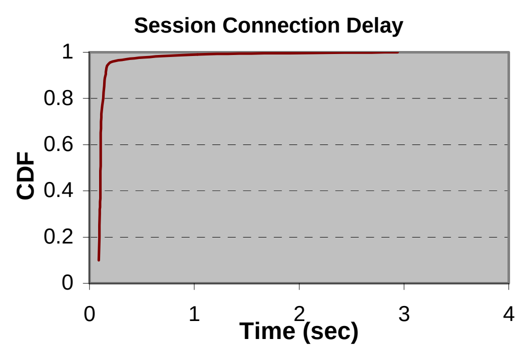

Figure 3. Session Connection Delay for New Arrival Nodes

Without the tracking capability, the newly arrived node in the network is connected only when the node starts to communicate with someone in the network. It thus often takes more than several minutes before the new node is known/heard in the network.

B. Application Scalability Studies

A MANET with a size of $4 \times 4\,\text{km}^2$ is compared with the one that has a size of $1.2 \times 1.2\,\text{km}^2$. The networks that consist of 31, 51 and 71 nodes are simulated. In the network comprised of 31 nodes, all of them are SIP UA nodes. In the network of 51 and 71 nodes, 10 of them are non-SIP nodes participating only in the media transmissions. The pull_2 scheme is applied in all cases. All the other simulation parameters remain the same as in the previous cases except that the pausing time of the nodes is varied to test the mobility impact on the performance. Figure 4 presents the successful session setup ratios comparing different scenarios. The labels used in Figure 4, such as "71_n_Bnet" and "71_n_Snet" represent the networks of $4 \times 4\,\text{km}^2$ and $1.2 \times 1.2\,\text{km}^2$ that are comprised of 71 nodes, respectively.

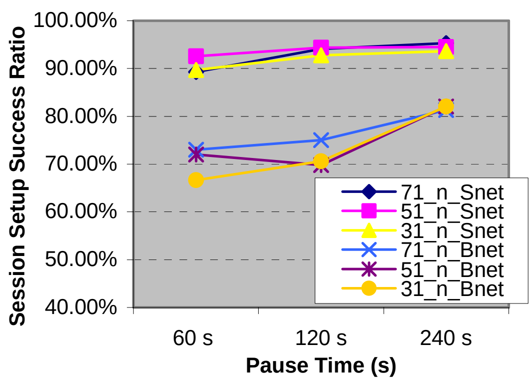

Figure 4 Comparison between the networks

The less dense network often has poorer session throughput ratio. A less dense network also suffers more in session setup failures when the node mobility increases, due to the loss of connectivity among the nodes.

To study the application performance under higher network traffic load, the networks that are comprised of 30, 50 and 70 SIP nodes with a size of $3 \times 3\,\text{km}^2$ are simulated. In each scenario, half of the nodes would initiate sessions according to an exponentially distributed inter-arrival time with a mean value of 180 s. The session duration is still modeled as exponentially distributed with an average of 60 s. All nodes are in the network from the beginning of the simulation. In addition to the voice session communications, 20 percent of the nodes send continuous UDP traffic at a rate of 5 kbps/s. Only the pull scheme is applied in these cases for the SIP UACs to discover the UAS nodes.

The session setup success ratios comparing different scenarios are illustrated in Figure 5. The session setup ratio degrades when the network scales up from 30 nodes to 70 nodes. Network traffic load, including the application and the routing protocol traffic load is the dominating factor impacting the session throughput performance.

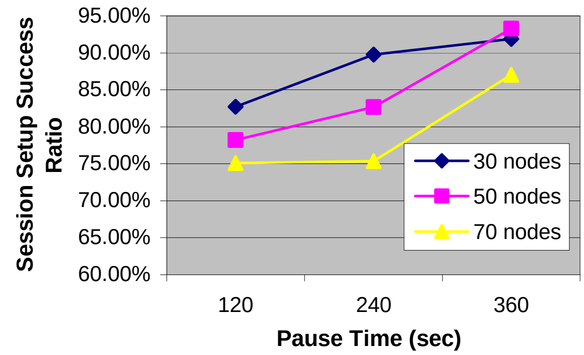

Figure 5 Session Setup Ratio in 3x3km^2 scenarios

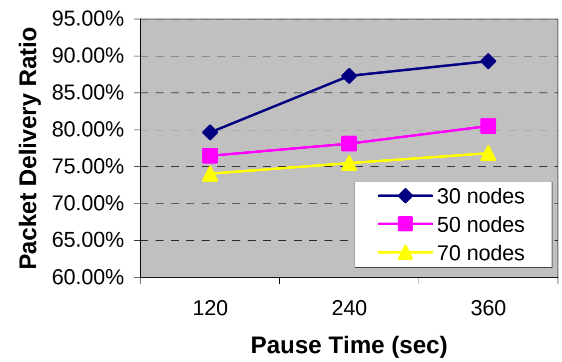

Figure 6 Packet Delivery Ratios in 3x3km^2 scenarios

Figure 6 presents the average packet delivery ratios of the above scenarios, including both the data and the SIP session voice packets. The overall packet delivery ratio worsens when the number of nodes increases in the network. The less dense network of 30 nodes also sees a faster drop in the metric when the network mobility increases.

Pause Time	Voice Packets Delivery Ratio		
	30N	50N	70N
360 s	88.09%	80.81%	81.64%
240 s	90.88%	81.11%	74.66%
120 s	90.65%	82.14%	73.21%
	Data Packets Delivery Ratio		
	30N	50N	70N
360 s	90.89%	80.18%	72.47%
240 s	82.97%	75.01%	79.78%
120 s	69.01%	72.01%	73.93%

Table 2: Packet Delivery Ratios of Voice and Data Packets

In Table 2, the voice and the data packet delivery ratios are depicted separately. In many cases, except in the scenarios with large number of nodes, e.g., 70 nodes, the session based voice packet delivery ratio holds its measurements better than that of the UDP data packets when the network mobility increases. In the scenarios of 30 and 50 nodes, when the data packets experience higher drop rate due to the reduced node pause time, the voice packets see even slight increase in their packet delivery ratios, benefiting from a less loaded network that has fewer voice sessions and lower UDP data throughput. The reason for this can be that the media transmission of a session only starts after the session is set up. When the session can be set up successfully, the nodes involved in the session are already "tested" as reachable. Then there is a relatively higher possibility that they may remain reachable for the session, especially in a session with short duration.

The data delivery ratio, on the other hand, shows that denser network may outperform the sparse network when the node mobility increases (e.g., when pause time is at 120 s, networks of 50 and 70 nodes show higher data delivery ratios than the 30 nodes scenarios amid all the voice traffic in the network.) The network of 70 nodes may perform poorly due to heavy traffic load, including both the higher routing protocol overhead and the application traffic, though its data delivery may benefit insignificantly from the degradation in its session setup ratios. In TacMAN, when network is scaled over 50 nodes, the Hierarchical OLSR (HOLSR) is applied to improve the performance for large scaled MANET, as demonstrated in [16][17].

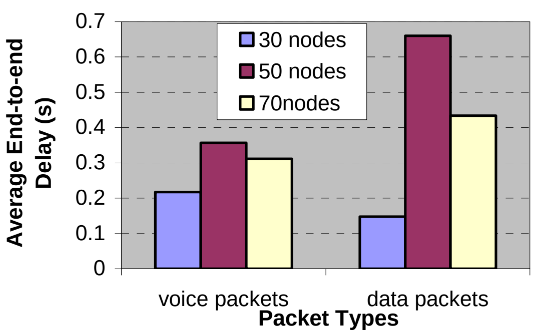

Figure 7 Packets End-to-end Delay

Figure 7 depicts the average end-to-end delay of the voice and data packets in the scenarios of 30, 50 and 70 nodes with a pause time of 360 s. Different levels of traffic load affects the delay metrics. In the scenarios with 50 nodes, the delays seem to be longer even than those of the networks with 70 nodes. This can be caused by the lower voice session setup ratio in the networks of 70 nodes, which may result in relatively more bandwidth for the traffic delivery, as more nodes add more bandwidth and higher level of path connectivity in the network.

V. CONCLUSION

In this paper, we presented the service architecture of TacMAN that supports realtime session applications using the standard protocol of SIP. Applying cross-layer design, TacMAN leverages on the functionality and information provided by the proactive routing layer to efficiently deliver the integrated realtime multimedia session services. We have verified using simulation that the pull scheme in service discovery for setting up individual sessions may be more advantageous and flexible than the push scheme as the former mitigates the session setup delay caused by waiting for the agent information. The tracking capability offered by the service discovery function greatly improves the situation awareness in the network. The performance measurements also reveal that a less dense network with fewer nodes is more vulnerable to the packet delivery performance impairment when network mobility increases. The packet delivery performance metrics of the session based service traffic are also found to be more sustainable under the increased network mobility, compared to those of the UDP data traffic.

ACKNOWLEDGEMENT

This work was supported by Defense Research and Development Canada.

REFERENCES

[1] U Kozat and L Tassiulas, "Service Discovery in Mobile Ad Hoc Networks: An Overall Perspective on Architectural Choices and Network Layer Support Issues", *Ad Hoc Networks*, Vol 2, 2004, pp.23-44

[2] Paal E. Engelstad et al, "Name Resolution in on-demand MANETS and over external IP Networks" *IETF Internet Draft*, draft-engelstad-manet-name-resolution-01.txt, Jan 2004

[3] T. Clausen, P. Jacquet et al., "Optimized Link State Routing Protocol", *IETF RFC 3626*, October 2003

[4] T. Clausen, "Comparative Study of Routing Protocols for Mobile Ad hoc Networks", INRIA Research Report, N.5135, Mar 2004

[5] J. Rosenberg et al., "SIP: Session Initiation Protocol", IETF RFC 3261, June 2002.

[6] J. Rosenberg and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", IETF RFC 3263, June 2002.

[7] H. Schulzrinne, "Dynamic Host Configuration Protocol (DHCP-for-IPv4) Option for Session Initiation Protocol (SIP) Servers", IETF RFC 3361, August 2002 .

[8] J. Jeong et al., "DNS Service for Mobile Ad Hoc Networks", IETF Internet draft, draft-jeong-manet-dns-service-00.txt, February 2004

[9] Rajeev Koodli and Charles Perkins, "Service Discovery in On-Demand Ad Hoc Networks", draft-koodli-manet-service-discovery-00.txt, IETF Internet Draft, 2003.

[10] David B. Johnson and David A. Maltz, *Dynamic Source Routing in ad hoc Wireless Networks, in Mobile Computing*, edited by Tomas Imielinski and Hank Korth, Chapter 5, pages 153-181, Kluwer Academic Publishers, ISBN: 0792396979, 1996.

[11] P. Faltstrom and M. Mealling, "The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)", IETF RFC 3761, April 2004

[12] Liang Cheng, "Service Advertisement and Discovery in Mobile Ad Hoc Networks", Workshop on Ad hoc Communications and Collaboration in Ubiquitous Computing Environments, in conjunction with the ACM 2002 Conference on Computer Supported Cooperative Work, New Orleans, Louisiana, USA, November 16-20, 2002.

[13] A. Qayyum et L. Viennot A. Laouiti. "Multipoint relaying: An efficient technique for flooding in mobile wireless networks", Technical report, INRIA, 2002.

[14] L. Li, L. Lamont: "A Lightweight Service Discovery Mechanism for Mobile Ad hoc Pervasive Environment using Cross-Layer Design", In Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications Workshops, Kauai Island, Hawaii, March 2005.

[15] J. Rosenberg, "A Presence Event Package for the Session Initiation Protocol (SIP)", RFC 3856, Aug 2004.

[16] L. Villasenor, Y. Ge, and L. Lamont, "HOLSR – A Hierarchical Proactive Routing Mechanism for Mobile Ad hoc Networks", to appear in *IEEE Communications Magazine*, July 2005.

[17] Y. Ge, L. Lamont and L. Villasenor, "Hierarchical OLSR – A Scalable Proactive Routing Protocol for Heterogeneous Ad hoc Networks", Proceedings of IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, August 22 - 24, 2005, Montreal.