Prioritized Overlay Multicast in Mobile Ad Hoc Environments

The authors propose a model to improve the efficiency and robustness of overlay multicast in manets by building multiple role-based prioritized trees, possibly with the help of location information about member nodes.

Li Xiao
Abhishek Patil
Yunhao Liu
Michigan State University

Lionel M. Ni
Hong Kong University of Science and Technology

Abdol-Hossein Esfahanian
Michigan State University

An increasing number of multicast applications are being developed for mobile ad hoc networks. However, available multicast routing protocols for manets are not as efficient and robust as those used in unicast networks.[1] Many network-layer, or IP-based, multicast routing protocols have been proposed for manets[2] to respond to both network and group dynamics. These protocols require member and nonmember nodes of a multicast group to maintain and update route information, which is very complicated and incurs significant overhead when groups have different priorities.

Recently, many researchers have begun focusing on application-layer, or overlay, multicast. In this approach, participating member nodes perform multicast functions, and an overlay network forms a virtual network consisting of only member nodes atop the physical infrastructure. This eliminates the need to change the application-layer tree when the underlying network changes and enables the overlay network to survive in environments where nonmember nodes do not support multicast functionality. An overlay protocol monitors group dynamics, while underlying unicast protocols track network dynamics, resulting in more stable protocol operation and low control overhead even in a highly dynamic environment.

Although not as efficient as IP-based multicast, overlay multicast is flexible and easy to implement. In some applications, participating nodes can be members of more than one overlay tree or they can build a temporary tree to perform certain important tasks. For such applications to be successful, however, nodes belonging to more than one tree must be smart enough to ignore incoming messages from members in low-priority trees while listening to members from a higher-priority tree.

We propose a prototype of prioritized overlay multicast (POM) for manets in which participating nodes can carry out several different functions and thus be associated with more than one overlay tree. At times some member nodes can form a short-term multicast group to perform certain important tasks. Various overlay trees can have different levels of priority depending on the importance of the service they perform.

OVERLAY MULTICAST IN MANETS

Researchers have proposed numerous protocols to improve the efficiency and reduce the latency of overlay multicast in manets. These include the ad hoc multicast routing protocol, progressively adaptive subtree in dynamic mesh, and the location-guided tree construction scheme.

AMRoute

The ad hoc multicast routing protocol[3] builds a robust multicast network out of user-multicast trees and dynamic logical cores. AMRoute first constructs per-group multicast distribution trees using unicast tunnels between group members. It then creates bidirectional tunnels between neighbors in the multicast tree to form a virtual mesh. From this mesh, the protocol uses a subset of links to generate a shared multicast distribution tree. Packets

0018-9162/04/$20.00 © 2004 IEEE

physically pass between neighboring nodes via a unicast tunnel and can go through several intermediate nodes. Unicast tunnel paths can change with the network topology without affecting the user-multicast trees.

AMRoute maintains a logical core in every tree responsible for mesh and tree creation. Noncore members cannot perform these actions and act only as passive responding agents. The core can migrate dynamically depending on group membership or network connectivity; thus, core loss will not disrupt data flow. However, AMRoute is inefficient because it uses a static virtual mesh to build the shared multicast distribution tree.

POM is generally applicable to situations requiring the setup of a communications network without an infrastructure.

PAST-DM

Progressively adaptive subtree in dynamic mesh[4] is an overlay multicast protocol defined for manets that tries to eliminate redundant physical links and thereby reduce the multicast session's overall bandwidth consumption. Unlike AMRoute, PAST-DM's virtual topology constantly adapts to changes in the underlying network topology.

With PAST-DM, each node implements the expanded ring search algorithm[5] to become aware of neighboring member nodes. Nodes periodically exchange the link-state table with their neighbors in a nonflooding manner such that, after several exchanges, a given node's link state reaches distant nodes. Thus, by looking at each node's link state, a node can view the entire network.

PAST-DM uses this information to build source-based trees, which are more efficient for data delivery than shared trees. Because link-state information is more accurate and up-to-date for nodes closer to the source, the virtual link closer to the source node wins any tie between links of the same cost during tree construction.

LGT

The location-guided tree[6] construction scheme includes two position-based multicast protocols for groups of nodes modeled by complete unit graphs, in which the source of multicast messages and all destination nodes are within transmission radius of one another and aware of the geographic position of any other node in the group.

In the location-guided *k*-ary (LGK) algorithm, the sender node selects *k* nearest destinations as children nodes, groups the rest of the *k* children according to close geometric proximity, and forwards a copy of the packet to each of the *k* children with its corresponding subtree as destinations. The process continues recursively with these children as new source nodes.

The location-guided Steiner tree (LGS) algorithm uses an incremental approach to generate a Steiner tree. Initially, the tree contains only the source node. At each iteration, the algorithm finds the nearest—in terms of geographic distance—unconnected destination node to the partially constructed tree and adds it to the tree; this node receives the message from the tree node to which it connects. Only group nodes are used as tree nodes.

Simulations show that the bandwidth cost for LGS is less than that for LGK when the nodes' location information is up-to-date. However, when the information is obsolete, LGK performs better due to its lower computational complexity. The protocols are efficient solutions for the complete graph environment only in wireless networks in which each node's receiver frequency is known to all other nodes.

Two techniques have been proposed to improve the LGT scheme. In the *optimal paths* method, every node receiving a multicast message for a group of nodes forwards it to each neighbor that is closest to one of the group members. In the *aggregate paths* method, each node counts the closest destinations and then applies a covering algorithm to choose a neighbor that covers the maximum number of destinations. These destinations are eliminated from the list, and then another neighbor is chosen that covers the maximum number of remaining destinations, and so on. As in the optimal-paths method, this algorithm changes a multicast group's forwarding list.

PRIORITIZED OVERLAY MULTICAST

In contrast to these approaches, POM builds priority trees with certain nodes carrying important tasks in overlay networks and rearranges low-priority trees whenever some nodes temporarily move to a high-priority network. The model is generally applicable to situations requiring the setup of a communications network without an infrastructure, such as at a large sports venue.

Policing a stadium or arena with thousands of spectators has always been a challenge. The most effective approach is to scatter security personnel among the crowd to monitor any suspicious activity and, if necessary, request nearby assistance. However, security guards typically convey requests for help as well as information such as physical descriptions of suspects via walkie-talkies, which have static noise. Crowd noise can also make it difficult to hear messages.

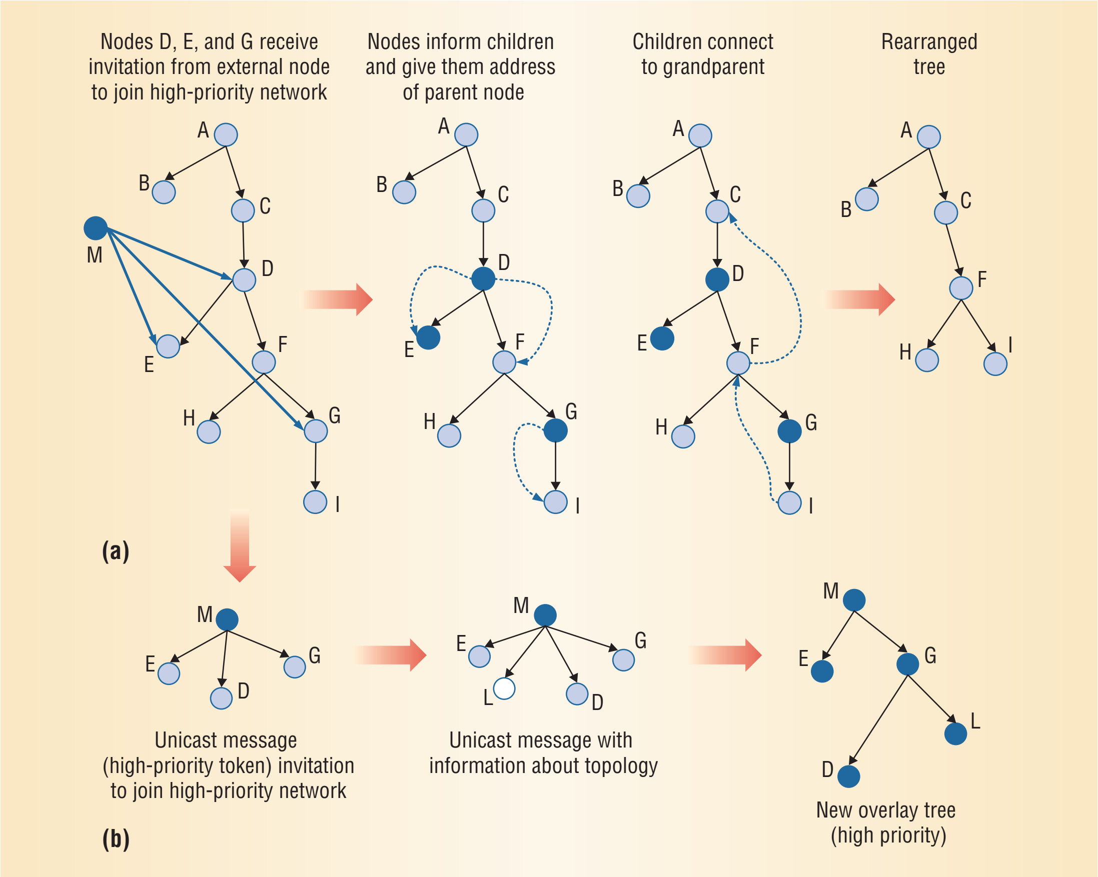

Figure 1. Priority tree formation in POM. (a) Upon receiving an invitation to join a high-priority network, nodes D, E, and G inform their children and provide their parent node's address; the children then connect to their grandparent, resulting in a new tree. (b) Node M sends a unicast high-priority token to each desired node and exchanges information about the formation of the new overlay tree topology.

POM would enable security personnel at such events to exchange multimedia data, such as images or video clips of suspects, via wireless handheld devices using an overlay multicast network. Because this data does not rely on an individual's perception or point of view, it is far more accurate than simple audio descriptions. In addition, a group of security guards could form a temporary network that would give higher priority to messages from group members while ignoring messages from other, lower-priority networks such as those that event operators and managers use.

Priority tree formation

Multiple priority trees can be built in the same environment to offer different services of varying importance. Nodes belonging to different trees switch between networks depending on what functionality they provide. At any given time, a node associates itself only with the highest priority tree in its set, ignoring incoming messages from members in lower-priority trees.

A node that initiates formation of a new priority tree supplies *priority tokens* whose value determines the tree's priority. Thus, a node that is currently a member of priority tree i would not listen to data from member nodes belonging to $i - 1$ or a lower-priority tree. Upon dissolution of i, the member nodes downshift to the next-highest priority tree in the set.

As Figure 1a shows, a node that decides to form a high-priority tree or receives an invitation to join a high-priority network could leave behind several orphans. Because these nodes must connect to another node in the original network to receive data from the source node, a departing node sends a control message to its children informing them that it is leaving the network. The departing node also provides its parent address, enabling the children to contact their grandparent node and receive data from it.

Member nodes use multihop means to communicate with one another; nonmember nodes can act as intermediate nodes. Thus, although nodes F and I are not physically close to each other, they can be neighbors in the logical topology. If the grandparent cannot support the new nodes, it will pass on the connection request to the source node. In location-aware trees, the source node has location information for the entire topology, and it should be able to redirect the orphan's connection to a suitable node.

Figure 1b illustrates the new high-priority tree's formation. External node M contacts nodes D, E, and G of the original tree and another external node, L, to form a high-priority network. M asserts its priority by sending a unicast token message to each of the desired nodes. It then exchanges information about the tree topology's formation, which is based on the nodes' location information with

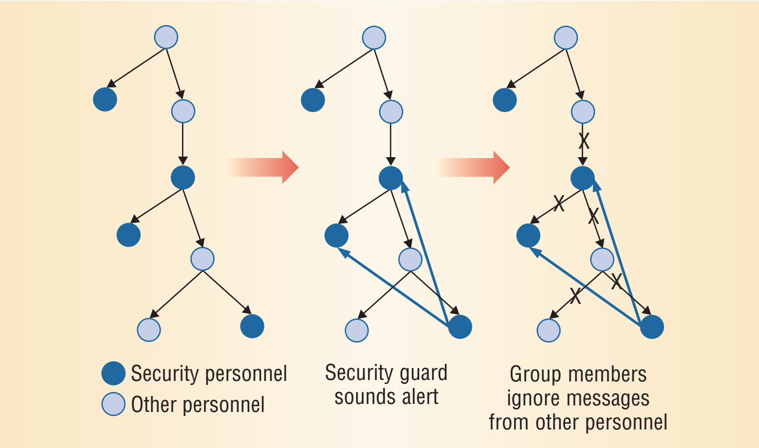

Figure 2. Role-based partitioning of overlay network. Because the network is partitioned at the application layer, nodes in the new higher-priority tree would not recognize members of lower-priority trees.

respect to one another. The number of steps in the tree's final formation varies depending on the implementation or algorithm used. In our approach, the source node implements location discovery and informs the other member nodes.

Location discovery

POM also uses location information to build overlay trees. In the security scenario, security personnel and event organizers would carry mobile devices that are tuned to communicate over the same channel using the same service set identifier for the wireless network. Under normal conditions, the nodes would all belong to the same tree. However, in an emergency situation, a security node can initiate formation of a new high-priority network consisting of only security nodes.

The security guards would carry devices that implement some form of neighbor discovery protocol to identify mobile devices that other nearby guards carry. This protocol can proactively conduct periodic checks for neighboring devices, or the user can initiate the search. When the participating security nodes receive an alert message, they will ignore messages from nonsecurity nodes, which are now considered as belonging to a lower-priority network.

The application layer at the mobile receivers can filter messages according to source or message/group priority. For example, security personnel would carry devices that are smart enough to ignore messages from an event organizer while they are receiving a video clip from another security guard or when they are assigned the task of frequently reporting on a suspect's activities. As Figure 2 shows, the security nodes would be part of a new higher-priority tree; thus, they would not recognize lower-priority tree members.

IMPLEMENTING POM

To assess POM's feasibility, we have identified a suitable unicast ad hoc routing protocol, explored the use of location information to build more efficient overlay trees, and studied how wireless node density and packet size impact system performance.

Simulation methodology

We used the CMU extension for ad hoc networks to simulate POM with an *ns* 2.1b7a discrete event simulator on a Dell Precision Workstation 330 running RedHat Linux 7.3. Currently, *ns* does not have an extension for simulating overlay multicast, although it does have an extension to simulate two network-layer multicast routing protocols: on-demand multicast routing and adaptive demand-driven multicast routing.

With the help of *bash* scripting, we modified the traffic pattern that CMU's *cbrgen* utility generates. We used the extension's *setdest* utility to generate different node positions and movement patterns. The parameters included the number of nodes, pause time, speed, and time and area of simulation. In our simulation, the total number of nodes is 25, of which 12 are member nodes, and the simulation area is 800 × 800 square meters.

The nodes move according to the random waypoint model.[8] When the simulation starts, each node is stationary at a particular location in the specified area for a time equal to the pause time. After the pause time expires, the nodes select a random destination within the given area and start moving with a maximum speed specified during creation of the scenario file. After reaching the destination, the nodes remain stationary for a period equal to the pause time, then they select another destination and proceed toward it in the same manner. Given the high sensitivity of protocol performance to the movement pattern, we carried out simulations for 10 different patterns for every parameter combination.

Unicast protocol identification

Because an overlay network forms a logical network of multicast member nodes, the underlying network regards the data exchange between such nodes as a unicast communication. This communication can use dynamic source routing (DSR),[8] ad hoc on-demand distance vector routing (AODVR),[5] destination-sequenced distance-vector routing (DSDVR),[9] or the temporally ordered routing algorithm (TORA).[10]

To identify an efficient ad hoc routing protocol with low latency, a low drop rate, and minimal overhead for POM, we analyzed simulation results for a general overlay tree. Because it performed poorly compared with the other three protocols, TORA was eliminated after the initial simulation rounds.

The speeds considered in the simulation were 1 meter per second (human walking) and 5 meters per second (human running). We measured the average time to complete the transfer of a 100-Kbyte file

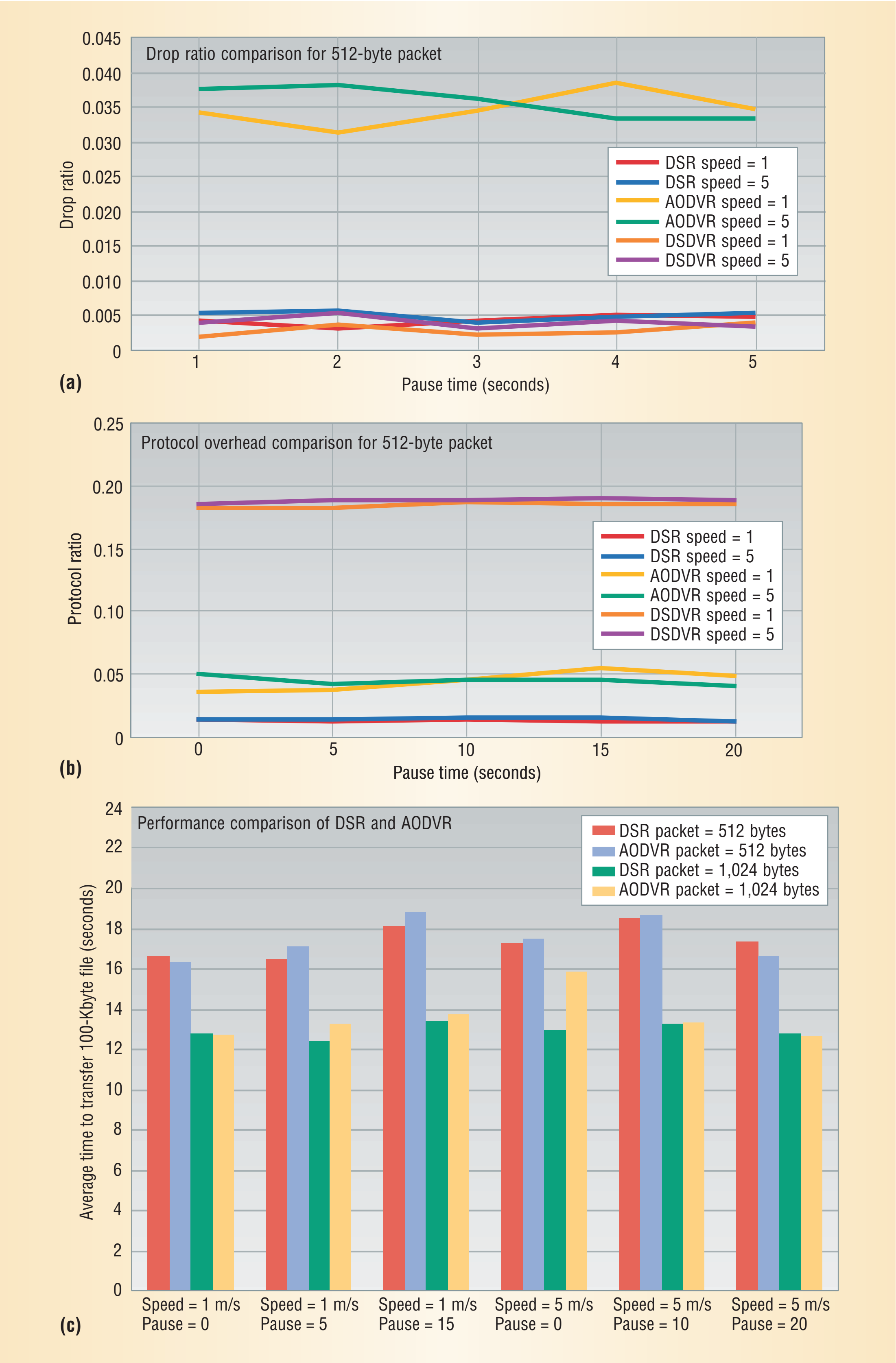

Figure 3. Performance comparison of unicast protocols in POM in terms of (a) drop ratio, (b) protocol overhead, and (c) average completion time. DSR = dynamic source routing; AODVR = ad hoc on-demand distance vector routing; DSDVR = destination-sequenced distance-vector routing.

to all member nodes, the average drop ratio (the ratio of total number of packets dropped to the total number of packets sent), and the average protocol ratio (the ratio of total number of protocol message packets to total number of packets sent).

Figure 3 shows that AODVR has a high drop ratio compared with the other two protocols, while DSDVR has a high protocol overhead. The figure

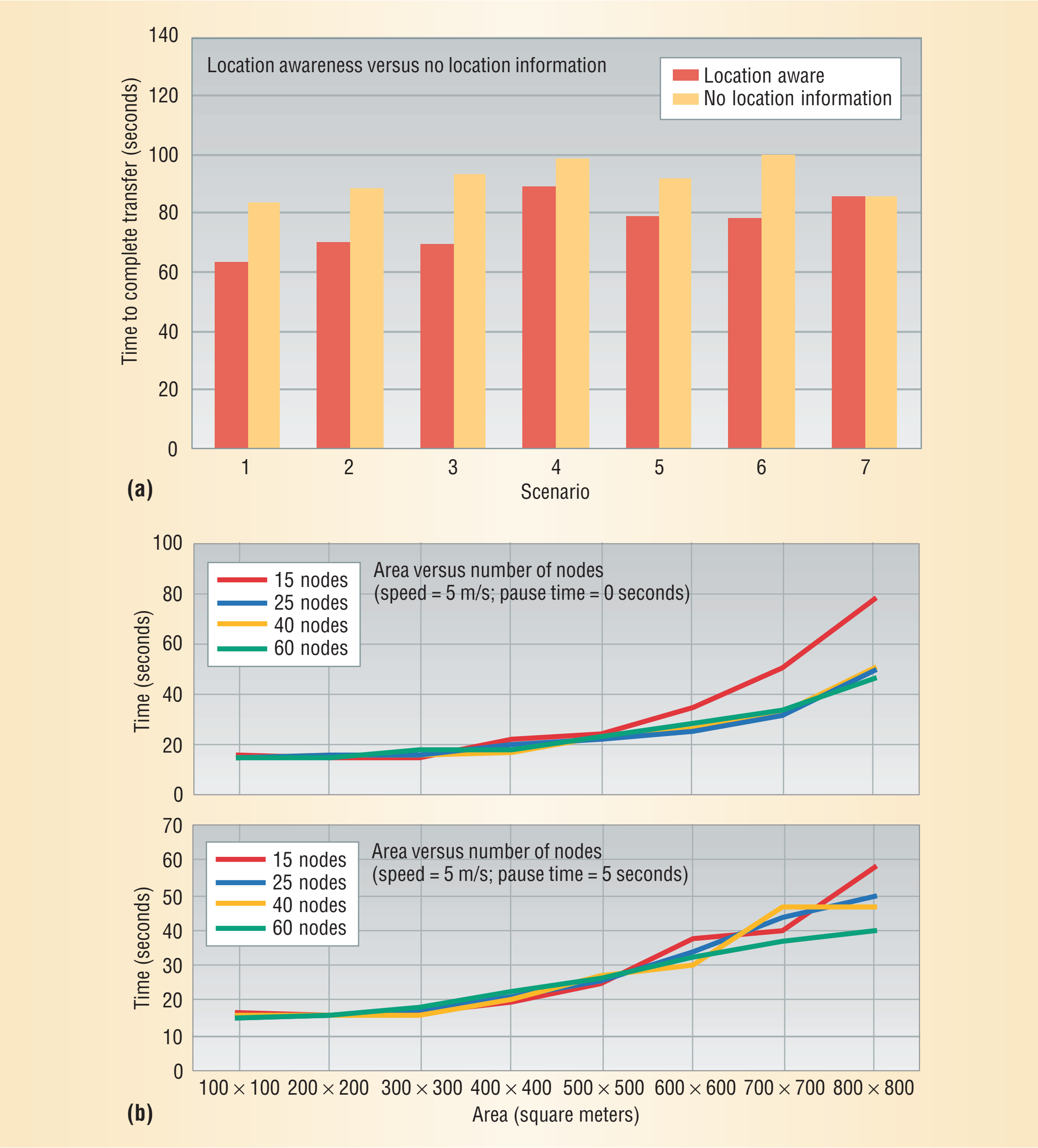

Figure 4. Impact of location information and node density on performance. (a) Location-aware overlay trees generally have lower latency than trees built without such information. (b) Increasing node density facilitates multihop forwarding, resulting in lower latency.

does not show DSDVR's average completion time because it is much higher than that of AODVR and DSR. Completion times for 1,024-byte packets are less than those for 512-byte packets, indicating that larger packet sizes can result in better performance. DSR and AODVR are comparable in terms of transfer time. However, because AODVR's drop ratio increases with packet size, it is high for 1,024-byte packets. Increasing packet size also reduces transfer time.

DSR benefits from source routing, in which packets carry route information to the destination. Consequently, aside from initial route discovery, DSR does not exhibit a high protocol overhead. With AODVR, each node participating between the source and destination must maintain information about the route. Also, to maintain routes, AODVR normally requires periodical transmission of a hello message with a default rate of once per second.

With these results in mind, we settled on a 1,024-byte packet size and DSR as the underlying unicast routing protocol for location information and node density simulations.

Location-based multicast trees

Improvements in location sensing techniques now make it possible to inexpensively locate an object's position within 1 meter in an indoor environment.[11] At the same time, differential Global Positioning System technologies have greatly increased outdoor location-positioning accuracy;[12] CompactFlash cards are available that can be plugged into handheld devices to enable GPS capability. Given the dynamically changing topology of manets, location-sensing technologies could be helpful in identifying nearby member nodes during tree formation—for example, using geometric distance as a heuristic.[6]

Figure 4a compares the performance of overlay trees built with and without location information in seven different movement scenarios. As shown, location-aware overlay trees have lower latency than trees built without such information. However, in extreme cases such as scenario 7, in which one or more member nodes is isolated or does not have overlapping coverage with other nodes, even location information brings little improvement.

Mobile node density

Because each mobile node has limited coverage, node density greatly influences network performance. When there is a high density of nodes in a given area, more nodes are available to perform multihop forwarding. We tested 15, 25, 40, and 60 nodes in areas ranging from 100 × 100 to 800 × 800 square meters. As Figure 4b shows, in smaller areas, the number of nodes has little impact on the network, but as the area increases, nodes scatter and there is little overlap in coverage. Thus, as the number of hops from source to destination increases, latency increases.

POM can be applied to a wide spectrum of mobile communications applications in which setting up an infrastructure-based system is difficult and the organizers desire a role-based partition in their network. For example, at a hospital with an overlay network for medical personnel certain doctors, nurses, and attendants could form their own temporary network to respond to a particular emergency. We continue to explore alternative ways to build a location-aware tree in POM that efficiently balances the tree's effectiveness with the overhead involved in building it. ■

References

1. C. Perkins, "Mobile Ad Hoc Networking Terminology," IETF Mobile Ad Hoc Networks Working Group, Internet Draft, work in progress, 30 Oct. 1997.
2. C.M. Cordeiro, H. Gossain, and D.P. Agrawal, "Multicast over Wireless Mobile Ad Hoc Networks: Present and Future Directions," *IEEE Network,* vol. 17, no. 1, 2003, pp. 52-59.
3. E. Bommaiah et al., "AMRoute: Adhoc Multicast Routing Protocol," IETF Internet Draft, work in progress, 6 Aug. 1998.
4. C. Gui and P. Mohapatra, "Efficient Overlay Multicast for Mobile Ad Hoc Networks," *Proc. 2003 IEEE Wireless Comm. and Networking Conf.*, vol. 2, IEEE Press, 2003, pp. 1118-1123.
5. C.E. Perkins, E.M. Belding-Royer, and I. D. Chakeres, "Ad hoc On-Demand Distance Vector (AODV) Routing," IETF Mobile Ad Hoc Networks Working Group, Internet Draft, work in progress, 19 Oct. 2003.
6. K. Chen and K. Nahrstedt, "Effective Location-Guided Tree Construction Algorithms for Small Group Multicast in MANET," *Proc. 21st Ann. Joint Conf. IEEE Computer and Comm. Societies,* vol. 3, IEEE Press, 2002, pp. 1180-1189.
7. M. Mauve et al., *Position-Based Multicast Routing for Mobile Ad-Hoc Networks,* tech. report TR-03-004, Computer Science Dept., Univ. of Mannheim, 2003.
8. D.B. Johnson and D.A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing,* T. Imielinski and H. Korth, eds., Kluwer Academic, 1996, pp. 153-181.
9. C.E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *ACM SIGCOMM Computer Comm. Rev.,* vol. 24, no. 4, 1994, pp. 234-244.
10. V. Park and M. Corson, "Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification," IETF Mobile Ad Hoc Networks Working Group, Internet Draft, work in progress, 20 July 2001.
11. L.M. Ni et al., "LANDMARC: Indoor Location Sensing Using Active RFID," *Proc. 1st IEEE Int'l Conf. Pervasive Computing and Communication,* IEEE CS Press, 2003, pp. 407-415.
12. G. Dommety and R. Jain, *Potential Networking Applications of Global Positioning Systems (GPS),* tech. report TR-24, Computer Science Dept., Ohio State Univ., 1996.

***Li Xiao** is an assistant professor in the Department of Computer Science and Engineering at Michigan State University. Her research interests include parallel/distributed and Internet systems, system resource management, and the design and implementation of experimental algorithms. Xiao received a PhD in computer science from the College of William and Mary. She is a member of the IEEE Computer Society and the ACM. Contact her at lxiao@cse.msu.edu.*

***Abhishek Patil** is a PhD candidate in the Department of Computer Science and Engineering at Michigan State University. His research interests include wireless networking, location-aware technologies, and Web technologies. Patil received an ME in electrical and computer engineering from Michigan State University. He is a student member of the IEEE Computer Society. Contact him at patilabh@cse.msu.edu.*

***Yunhao Liu** is a PhD candidate in the Department of Computer Science and Engineering at Michigan State University. His research interests include peer-to-peer systems, overlay networking, pervasive computing, distributed systems, network security, and Internet and e-commerce technologies. Liu received an MS in computer science from Michigan State University. He is a student member of the IEEE Computer Society. Contact him at liuyunha@cse.msu.edu.*

***Lionel M. Ni** is a professor and head of the Department of Computer Science at Hong Kong University of Science and Technology. His research interests include location-aware computing, high-speed networking, wireless networking, high-performance computer architectures, Internet and Web technologies, parallel and distributed systems, and network security. Ni received a PhD in electrical and computer engineering from Purdue University. He is an IEEE Fellow. Contact him at ni@cs.ust.hk.*

***Abdol-Hossein Esfahanian** is an associate professor and graduate program director in the Department of Computer Science and Engineering at Michigan State University. His research interests include graph theory applications in computer science, fault tolerance in computer networks, and the design and analysis of algorithms and database applications. Esfahanian received a PhD in computer science from Northwestern University. He is a senior member of the IEEE and a member of the ACM and the Society for Industrial and Applied Mathematics. Contact him at esfahanian@cse.msu.edu.*

ISSRE 2004 **November 2-5, 2004**

The Fifteenth International Symposium on Software Reliability Engineering

General Chair
Yves Le Traon, France
yletraon@irisa.fr

Program Chairs
Lionel Briand, Canada
briand@sce.carleton.ca
Jeffrey Voas, USA
jmvoas@cigital.com

Tutorials Chair
Sherif Yacoub, USA

Workshop Chair
Benoit Baudry, France

Industry Practice Chairs
Peter Santhanam, USA
Brendan Murphy, USA
Yves-Marie Quemener, France

Supported by:
Cigital, USA
Irisa labs, France

Saint-Malo, Bretagne, France **http://issre.org/**

ISSRE focuses on the theory and practice of Software Reliability Engineering. The conference scope includes techniques and practices to (1) verify and validate software, (2) estimate and predict its dependability, and (3) make it more tolerant/robust to faults. The theme of this year is achieving software dependability through Model-Driven Engineering.

Important dates

• Deadline for Abstracts	April 2, 2004
• Deadline for Submissions	April 18, 2004
• Notification to Authors	July 15, 2004

Submissions

Regular papers, written in English, should be submitted electronically. Submissions must be unpublished and must not be submitted elsewhere. Proposals for advanced tutorials and workshops are requested. Details of the submission process are available on the website.

Conference Location

The conference Workshop and Tutorials Programme will be held Tuesday, November 2, at the Irisa Laboratory in Rennes. The Main Conference Programme will be held Wednesday through Friday, November 3-5 at the Palais du Grand Large in Saint-Malo.