Available online at www.sciencedirect.com
ScienceDirect
Ad Hoc Networks 6 (2008) 32–46
Ad Hoc Networks
www.elsevier.com/locate/adhoc

A study of different types of attacks on multicast in mobile ad hoc networks ☆

Hoang Lan Nguyen *, Uyen Trang Nguyen

Department of Computer Science and Engineering, York University, Toronto, Ont., Canada M3J 1P3

Received 27 July 2006; accepted 31 July 2006
Available online 31 August 2006

Abstract

We present a simulation-based study of the impacts of different types of attacks on mesh-based multicast in mobile ad hoc networks (MANETs). We consider the most common types of attacks, namely rushing attack, blackhole attack, neighbor attack and jellyfish attack. Specifically, we study how the number of attackers and their positions affect the performance metrics of a multicast session such as packet delivery ratio, throughput, end-to-end delay, and delay jitter. We also examine rushing attackers' success rates of invading into the routing mesh when the number of attackers and their positions vary. The results enable us to suggest measures to minimize the impacts of the above types of attacks on multicast in MANETs.

Keywords: Multicast; Attack strategies; Security threats; Vulnerability analysis; Performance analysis

1. Introduction

A mobile ad hoc network is a self-organizing system of mobile nodes that communicate with each other via wireless links with no fixed infrastructure or centralized administration such as base stations or access points. Nodes in a MANET operate both as hosts as well as routers to forward packets for each other in a multi-hop fashion. MANETs are suitable for applications in which no infrastructure exists such as military battlefield, emergency rescue, vehicular communications and mining operations.

In these applications, communication and collaboration among a given group of nodes are necessary. Instead of using multiple unicast transmissions, it is advantageous to use multicast in order to save network bandwidth and resources, since a single message can be delivered to multiple receivers simultaneously. Existing multicast routing protocols in MANETs can be classified into two categories: tree-based and mesh-based. In a multicast routing tree, there is usually only one single path between a sender and a receiver, while in a routing mesh, there may be multiple paths between each sender–

☆ A preliminary version of this paper appeared in the Proceedings of 2006 IEEE International Conference on Networking (ICN 2006). This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) through a Discovery Grant.
* Corresponding author. Tel.: +1 416 739 1586.
E-mail addresses: lan@cs.yorku.ca (H.L. Nguyen), utn@cs.yorku.ca (U.T. Nguyen).

1570-8705/$ - see front matter
doi:10.1016/j.adhoc.2006.07.005

receiver pair. Routing meshes are thus more suitable than routing trees for systems with frequently changing topology such as MANETs due to the availability of multiple paths between a source and a destination. Multicast data may still be delivered to the destination on alternative paths even when the main route breaks. Example tree-based multicast routing protocols are MAODV [8], AMRIS [9], BEMRP [24] and ADMR [23]. Typical mesh-based multicast routing protocols are ODMRP [4], FGMP [22], CAMP [5], DCMP [6], and NSMP [7].

Among all the research issues, security is an essential requirement in MANET environments. Compared to wired networks, MANETs are more vulnerable to security attacks due to the lack of a trusted centralized authority, lack of trust relationships between mobile nodes, easy eavesdropping because of shared wireless medium, dynamic network topology, low bandwidth, and battery and memory constraints of mobile devices. The security issue of MANETs in group communications is even more challenging because of the involvement of multiple senders and multiple receivers. Although several types of security attacks in MANETs have been studied in the literature, the focus of earlier research is on unicast (point-to-point) applications [15–17]. The impacts of security attacks on multicast in MANETs have not yet been explored.

In this paper, we present a simulation-based study of the effects of different types of attacks on mesh-based multicast in MANETs. We consider the most common types of attacks, namely rushing attack, blackhole attack, neighbor attack and jellyfish attack.

- *Rushing attack*. Many demand-driven protocols such as ODMRP [4], MAODV [8], FGMP [22], and ADMR [23], which use some form of duplicate suppression in their operations, are vulnerable to rushing attacks. When source nodes flood the network with route discovery packets in order to find routes to the destinations, each intermediate node processes only the first non-duplicate packet and discards any duplicate packets that arrive at a later time. A rushing attacker exploits this duplicate suppression mechanism by quickly forwarding route discovery packets in order to gain access to the forwarding group. Rushing attacks were first introduced by Hu et al. [1].
- *Blackhole attack*. A blackhole attacker first needs to invade into the multicast forwarding group (e.g., by implementing rushing attack) in order to intercept data packets of the multicast session. It then drops some or all data packets it receives instead of forwarding them to the next node on the routing path. This type of attack often results in very low packet delivery ratio.
- *Neighbor attack*. Upon receiving a packet, an intermediate node records its ID in the packet before forwarding the packet to the next node. An attacker, however, simply forwards the packet without recording its ID in the packet to make two nodes that are not within the communication range of each other believe that they are neighbors (i.e., one-hop away from each other), resulting in a disrupted route.
- *Jellyfish attack*. A jellyfish attacker first needs to intrude into the multicast forwarding group. It then delays data packets unnecessarily for some amount of time before forwarding them. This results in significantly high end-to-end delay and thus degrades the performance of real-time applications. Jellyfish attacks in MANETs were first discussed by Aad et al. [2].

Using simulation, we study how the number of attackers and their positions affect the performance of a multicast session in terms of packet delivery ratio, throughput, end-to-end delay, and delay jitter. Our simulation results show that a large multicast group with a high number of senders and/or a high number of receivers can sustain good performance under these types of attacks due to several alternative paths in the routing mesh. The most damaging attack positions are those close to the senders and around the mesh center. We also examine rushing attackers' success rates of invading into the routing mesh. We find that in order to increase the likelihood of being selected into the routing group the attackers must gather themselves in a group and stay near the receivers or around the mesh center.

Our contributions are as follows. First, we present experimental results that show how a mesh-based multicast session performs under various attack scenarios. Second, we identify several unique behaviors of a multicast network under attack, which have not been seen in unicast environments. Third, the obtained results allow us to suggest some counter-attack measures (e.g., adding more senders and/or receivers to the multicast group to improve

the resiliency to attacks). Finally, our results and observations may also be valuable for researchers who are doing research on intrusion detection and prevention for multicast in MANETs.

The remainder of the paper is organized as follows. Section 2 presents an overview of the multicast routing protocol used in our simulated network, and the implementation of the types of security attacks to be studied. Section 3 describes the setting of our experiments and the performance metrics. In Sections 4–7, we present experimental results obtained from simulating rushing, blackhole, neighbor and jellyfish attacks, respectively. Related work is discussed in Section 8. Section 9 summarizes our findings.

2. Implementation of the multicast routing protocol and security attacks

In our simulations, we used ODMRP (On-Demand Multicast Routing Protocol) as the multicast routing protocol due to its simple implementation and high packet delivery ratio. The advantages of ODMRP make it a very popular multicast routing protocol for MANETs: ODMRP has been cited, evaluated and compared with other multicast routing protocols in over 50 research projects and papers.

In this section, we first give a brief description of ODMRP. We then describe in detail the implementation of rushing, blackhole, jellyfish and neighbor attacks in our simulated network.

2.1. ODMRP overview

ODMRP uses the concept of *forwarding group*, which is a set of nodes responsible for forwarding multicast data on shortest delay paths between a sender and a receiver. An ODMRP source periodically updates routing tables and membership information by flooding the network with route refreshment packets called JOIN QUERY. Upon receiving a non-duplicate JOIN QUERY, an intermediate node stores the ID of the upstream node from which it receives the packet, and then rebroadcasts the packet. (Duplicate JOIN QUERY packets will be discarded.) When the JOIN QUERY packet reaches a multicast receiver, the receiver replies with a JOIN REPLY packet. The JOIN REPLY packet is relayed back towards the multicast source via the reverse path traversed by the JOIN QUERY packet. This process constructs (or updates) routes from the sources to the receivers, and builds a mesh of forwarding nodes.

2.2. Implementations of attacks

We simulate rushing attacks by introducing a simulated processing delay at every honest node. The node delays every JOIN QUERY for a certain amount of time varying from 10 ms to 40 ms before broadcasting it. Meanwhile, the nodes that are designated as rushing attackers have their simulated JOIN QUERY processing delay set to zero. A rushing attacker is considered successful in a route discovery interval if and only if it has forwarded a JOIN QUERY and later receives a JOIN REPLY in the same interval.

Note that rushing attack is not the only method enabling access to a multicast forwarding group. For protocols that do not use a duplicate suppression mechanism such as AMRIS [9], CAMP [5], and BEMRP [24], route invasion can be done by modifying or advertising false routing information. For example, in BEMRP, when a new receiver wants to join a multicast group, it floods a JOIN control packet. An intermediate node may receive more than one JOIN packet. After waiting for some predetermined amount of time, the node chooses one JOIN packet with the smallest *hop count* traversed. An attacker can gain access to the BEMRP forwarding group by falsely updating the *hop count* field in the received JOIN packet to a very small value before forwarding the packet to the next node. Once an attacker has invaded into forwarding routes, it may launch other attacks such as dropping data packets (blackhole attack), delaying them (jellyfish attack), or corrupting or illegally accessing confidential data. In this paper, we consider only route invasion by means of rushing attack since there exist many multicast routing protocols that use some form of duplicate suppression, and are thus vulnerable to rushing attack.

We implement blackhole attacks by assigning a certain number of nodes to be malicious nodes before each simulation begins. These nodes carry out rushing attacks to become multicast forwarding nodes. The malicious nodes have their JOIN QUERY processing delay set to zero. The JOIN QUERY processing delay of the honest nodes is set to 20 ms. Note that in order to have k blackhole attackers for a particular simulation scenario we may need to use more than k malicious nodes for rushing attacks because some of these nodes may fail to gain

access to the forwarding group. Rushing attacks were repeated until exactly k malicious nodes were selected to be forwarding nodes. These nodes then dropped the data packets belonging to the multicast session.

Jellyfish attacks are implemented in the same manner as blackhole attacks, except that an attacker, after gaining access to the multicast forwarding group, will delay every data packet for a random amount of time ranging from 0 to 10 s before forwarding the packet.

To implement neighbor attacks, we modified the JOIN QUERY processing function of ODMRP so that an attacker simply forwards the JOIN QUERY without recording its ID into the packet. This way the attacker makes its upstream and downstream neighbors believe that they are directly connected to each other while they actually are not.

3. Simulation setting

This section describes the parameters and performance metrics used in our simulations.

3.1. Simulation parameters

We conducted our experiments using QualNet version 3.8, a scalable simulation environment for wireless network systems developed by Scalable Network Technologies [14]. Our simulated network consists of 50 mobile nodes placed randomly within a 1000 m x 1000 m area. Each node has a transmission range of 250 m and moves at a speed of 1 m/s. The total sending rate of all the senders of the multicast group, i.e., the *traffic load*, is 1 packet/s. We use a low traffic load value to highlight the effects of the attacks on the packet loss rate, as opposed to packet loss due to congestion and collisions resulting from a high traffic load.

The attackers were positioned around the center of the routing mesh in all experiments, except for those described in Sections 4.3, 5.3 and 7.3. In these experiments, we simulated four scenarios: the attacker group was placed near the senders, near the receivers, and around the mesh center, respectively. In the fourth scenario, the attackers were uniformly distributed over the whole network. The duration of each experiment was 300 s in simulated time. Every experiment was repeated 10 times using 10 different randomly generated seed numbers, and the recorded data was averaged over those runs. Table 1 lists the values of the common parameters used in all the experiments. Other parameters will be given in the description of each specific experiment.

Table 1
Common simulation parameters

Parameter	Value
ODMRP route refreshment interval	20 s
Channel capacity	2 Mbits/s
Packet size (excluding header size)	512 bytes
Traffic model of sources	Constant bit rate
Mobility model	Random way-point [25]
Pathloss model	Two-ray [26]
Queuing policy at routers	First-in-first-out

3.2. Performance metrics

We use the following metrics in our study:

- *Average attack success rate*. The attack success rate of a rushing attacker is defined as the ratio of the number of times the attacker is selected to be a multicast forwarding member over the number of times the route discovery process is initiated. The average attack success rate is the average of the attack success rates taken over all the attackers.
- *Average packet delivery ratio*. The packet delivery ratio (PDR) of a receiver is defined as the ratio of the number of data packets actually received over the number of data packets transmitted by the senders. The average packet delivery ratio is the average of the packet delivery ratios taken over all the receivers.
- *Average end-to-end delay*. The end-to-end delay of a packet is defined as the time a packet takes to travel from the source to the destination. The average end-to-end delay is the average of the end-to-end delays taken over all the received packets.
- *Average delay jitter*. Delay jitter is the variation (difference) of the inter-arrival intervals from one packet received to the next packet received. Each receiver calculates the average per-source delay jitter from the received packets originated from the same source. The receiver then takes the average over all the sources to obtain the average per-receiver delay jitter. The average delay jitter is the average of the per-receiver delay jitters taken over all the receivers.
- *Average throughput*. The throughput of a receiver (per-receiver throughput) is defined as the ratio

of the number of bits received over the time difference between the first and the last received packets. The average throughput is the average of the per-receiver throughputs taken over all the receivers.

Following are our simulation results that demonstrate the effects of various types of attacks on mesh-based multicast in MANETs.

4. Rushing attacks

We simulated rushing attacks by varying the processing delay of honest nodes from 0 ms to 40 ms while setting the processing delay of rushing attackers to 0 ms. An attacker is considered successful only if it receives both JOIN QUERY and JOIN REPLY packets in the same route refreshment interval. We measured the average success rate of rushing attacks when varying the number of receivers and the number of senders of the multicast group, the number of attackers, and their positions.

4.1. Rushing attacks: number of receivers

This set of experiments compares the average attack success rates when there are 10, 20, and 30 multicast receivers, respectively. The number of multicast senders is set to three. We first set the number of attackers to three, and then repeated the experiment with five attackers. The results are shown in Fig. 1.

In these graphs, we observe a similar behavior. As the processing delay of legitimate nodes increases, the average success rate of malicious nodes also rises. The longer legitimate JOIN QUERY packets are delayed at intermediate nodes, the more rushed JOIN QUERY packets arrive at the destinations as the first JOIN QUERY of a route refreshment interval, allowing more attackers to be selected into the forwarding group. We also note that the higher the number of multicast receivers, the higher the attack success rate. In ODMRP the forwarding nodes are determined by the receivers (when they send JOIN REPLY messages). If an attacker was missed by a receiver, it still has another chance of being selected into the forwarding group by another receiver. Therefore, as the number of receivers increases, the attackers' chances of being selected also increase, resulting in higher success rates.

Readers may note that, given the same number of receivers, the average attack success rate in the network having three attackers is slightly higher than the average attack success rate in the network having five attackers. An example for the case of 10 receivers is extracted from Fig. 1(a) and (b), and shown in Fig. 3(a). This is simply due to our definition of the average attack success rate, which is defined as the average taken over the success rates of all attackers. Given the same network size, the same number of multicast senders and the same number of multicast receivers, as more attackers are added to the network, the probability that one or more of them are not selected into the forwarding group increases, making the average success rate of the whole attacker group decrease.

4.2. Rushing attacks: number of senders

In this set of experiments, we compare the attack success rates of rushing attacks when the number of multicast senders ranges from one to five. The

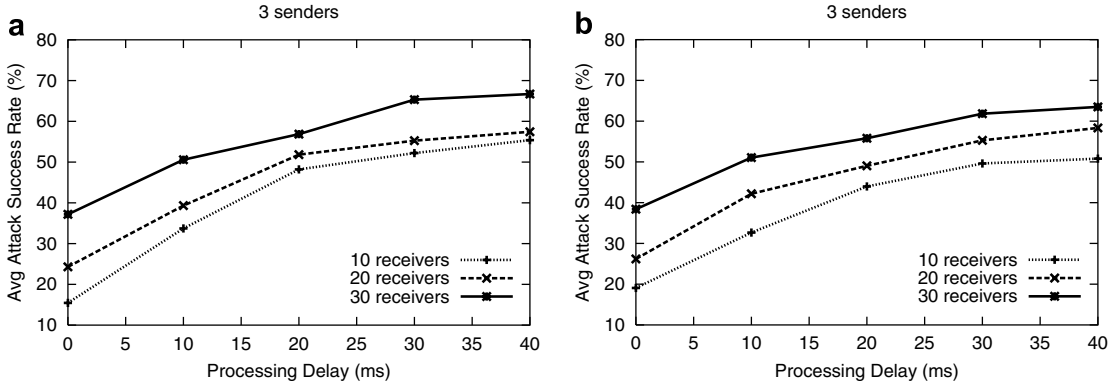

Fig. 1. Success rates of rushing attacks – Different numbers of multicast receivers. (a) Three attackers. (b) Five attackers.

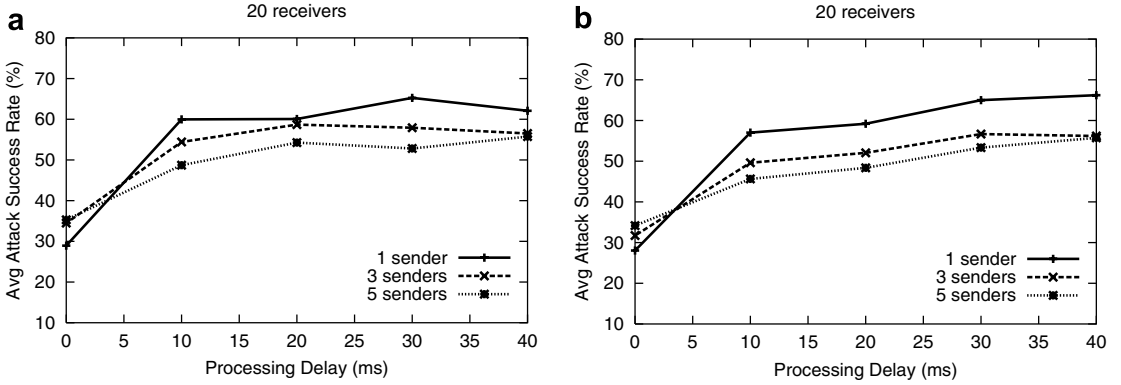

Fig. 2. Success rates of rushing attacks – Different numbers of multicast senders. (a) Three attackers. (b) Five attackers.

number of multicast receivers is fixed at 20 nodes. We ran the experiment with three and five attackers, and the resulting graphs are shown in Fig. 2(a) and (b), respectively.

We observe that, in both graphs, given the same number of attackers and the same processing delay of legitimate nodes, the higher the number of senders, the lower the attack success rate. When the number of senders increases, the routing mesh becomes more dense with more routes. That increases the number of adversary-free paths that are shorter than the attackers' rushing paths, causing more compromised paths to be rejected by the receivers.

Note that given the same number of multicast senders the average attack success rate of the 3-attacker group is slightly higher than that of the 5-attacker group. Fig. 3(b) shows such an example for the case of three senders. The reason has just been explained in Section 4.1.

4.3. Rushing attacks: attack positions

This set of experiments examines the effects of different attack positions on the attack success rate. We considered four cases as mentioned in Section 3.1. The attackers were grouped in three different areas, respectively: near the senders, near the receivers, and around the mesh center. In the fourth case, they were uniformly distributed over the whole network (when the number of attackers is more than one). The number of multicast senders is three, and the number of multicast receivers is 20. We ran experiments with one, three, five and seven attackers, respectively, and the results are shown in Fig. 5.

All the graphs show that the uniform distribution cases give the lowest success rates, indicating that spreading out over the network is the least effective attack strategy. The explanation is as follows. In order to be successfully selected into the forwarding

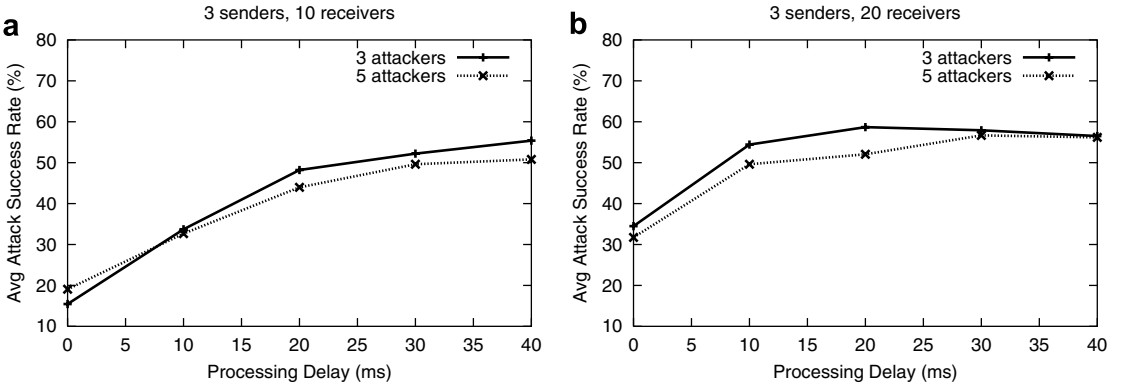

Fig. 3. Success rates of rushing attacks – Comparison between the cases of three and five attackers. (a) Graphs extracted from Fig. 1(a) and (b). (b) Graphs extracted from Fig. 2(a) and (b).

group, a potential attacker should position itself on paths between the senders and the receivers, i.e., either near the senders, or near the receivers, or somewhere inside the routing mesh. In the uniform distribution scenario, the attackers were scattered over the whole network. It may happen that a subset of them were not on any paths from the senders to the receivers, and thus failed to become forwarding members. As a result, the uniform distribution experiments give lower attack success rates than the near-sender, near-receiver, and mesh center experiments. The lesson learned is that, in order to maximize their collective success rate, the attackers should first locate the senders and the receivers, and then gather themselves in a group around them or inside the mesh. This is especially true when the number of attackers is small. The graphs show that as the number of attackers decreases, the gap between the uniform distribution cases and the other cases grows wider.

In all of these graphs, the near-sender area gives the lowest attack success rate, indicating that this is the least vulnerable attack position compared with the near-receiver and mesh center areas.

When there is only one attacker, the attack success rate of the mesh center position is much higher than that of the near-receiver position (Fig. 5(a)). This implies that when the number of attackers is small, the most powerful attack position is at the center of the routing mesh. However, as the number of attackers increases, the attack success rate of the near-receiver position gradually goes up to match that of the mesh center position (Fig. 5(b)–(d)); in other words, the near-receiver area becomes a stronger position to launch attacks.

The reason is as follows. In general, attackers that are close to the multicast receivers have the best chance of being selected as forwarding nodes. An intuitive explanation is illustrated by the examples given in Fig. 4. In Fig. 4(a), attacker A is one hop away from receiver R. Assume that a JOIN QUERY forwarded by node C arrives at A and legitimate node B at the same time. Since A *rushly* forwards the JOIN QUERY, the packet will arrive at R before the one forwarded by legitimate node B. When A moves away from R, for example, to a position that is two hops away from R (Fig. 4(b)), its success rate is likely to decrease. The reason is that the JOIN QUERY rushed by A now depends on legitimate node E to reach receiver R. Assume that a JOIN QUERY forwarded by node F arrives at A and legitimate nodes B and D at the same time. If E experiences congestion or high contention for wireless medium, the JOIN QUERY from either B or D may arrive at R before the rushed JOIN QUERY from A. R will then discard the JOIN QUERY from A as a duplicate, and the path passing through A (S–G–F–A–E–R) will not be chosen as a routing path. As A moves farther away from receiver R, its rushed JOIN QUERY packets depend on more intermediate legitimate nodes to reach the receiver, making its success rate smaller and smaller. This explains why positions near the senders give the lowest attack success rates compared with those close to the receivers and around the mesh center. In short, the closer to a receiver an attacker is, the higher its chance of invading into the multicast forwarding group.

However, when the number of attackers is small compared to the number of receivers, it is not feasible for them to stay close to all the receivers and still be able to intercept the data flows to every receiver. In this case, the small attacker group should stay at the center of the multicast mesh in order to cover as many paths that are shared by multiple receivers as possible.

4.4. Rushing attacks: summary

Rushing attacks are more likely to succeed in a multicast session where the number of multicast senders is small and/or the number of multicast receivers is large. With respect to attack positions,

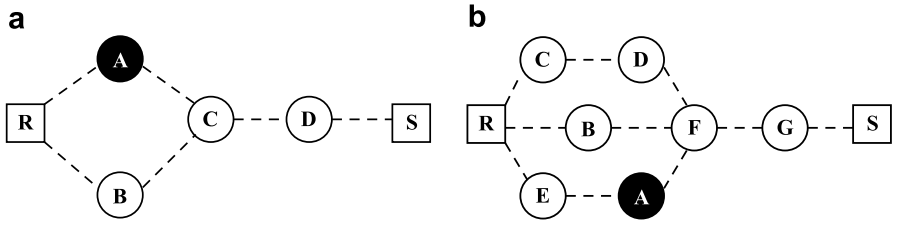

Fig. 4. Demonstration of rushing attacks with different positions. (a) Attacker A one hop away from receiver R. (b) Attacker A two hops away from receiver R.

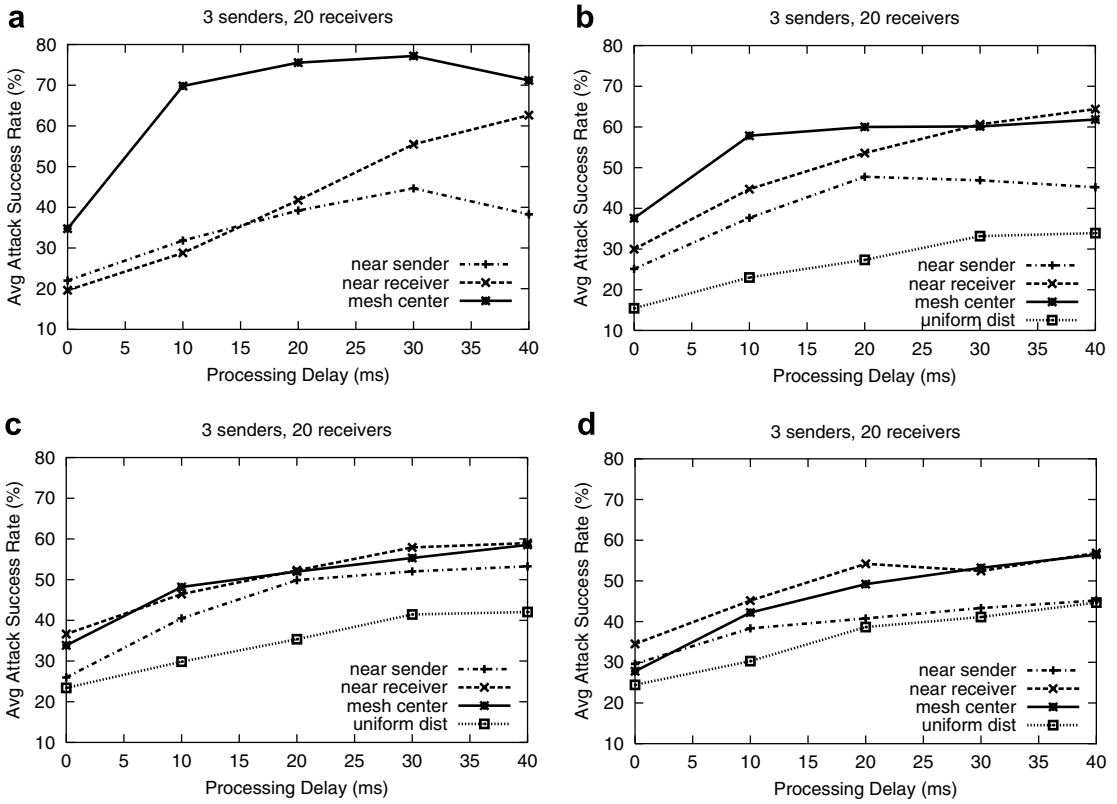

Fig. 5. Success rates of rushing attacks – Different attack positions. (a) One attacker. (b) Three attackers. (c) Five attackers. (d) Seven attackers.

the best position to launch rushing attacks is at the center of the multicast mesh if the number of attackers is small compared to the number of multicast receivers. Otherwise, attackers staying close to the receivers have the highest success rates. Attackers should identify the senders and the receivers, and stay close to them or between the senders and the receivers (as opposed to spreading out over the network) to increase their success rates.

5. Blackhole attacks

In this section, we examine the packet delivery ratio of multicast sessions under blackhole attacks. A blackhole attacker first implements rushing attacks as described in the previous section to gain access to the routing mesh, and then later drops all data packets it receives.

The processing delay of legitimate nodes is set at 20 ms. We investigated various scenarios by varying the number of senders, the number of receivers, the number of attackers, and their positions. In each experiment, we measured the packet delivery ratio (PDR) as a function of the number of attackers. The results are given in Fig. 6. We can see that in almost all cases, as the number of attackers increases, the PDR decreases as we would expect.

5.1. Blackhole attacks: number of receivers

We compare the PDRs of multicast groups having 10, 20 and 30 receivers, respectively. The number of multicast senders is fixed at three. The graph in Fig. 6(a) shows that, given the same number of attackers, the higher the number of multicast receivers, the higher the packet delivery ratio. As the number of receivers increases, the routing mesh becomes more dense. If a packet is dropped on one path, a duplicate copy of the packet may be delivered to the receivers via other paths in the mesh.

5.2. Blackhole attacks: number of senders

In this set of experiments, the multicast group has 20 receivers. The number of senders is set to

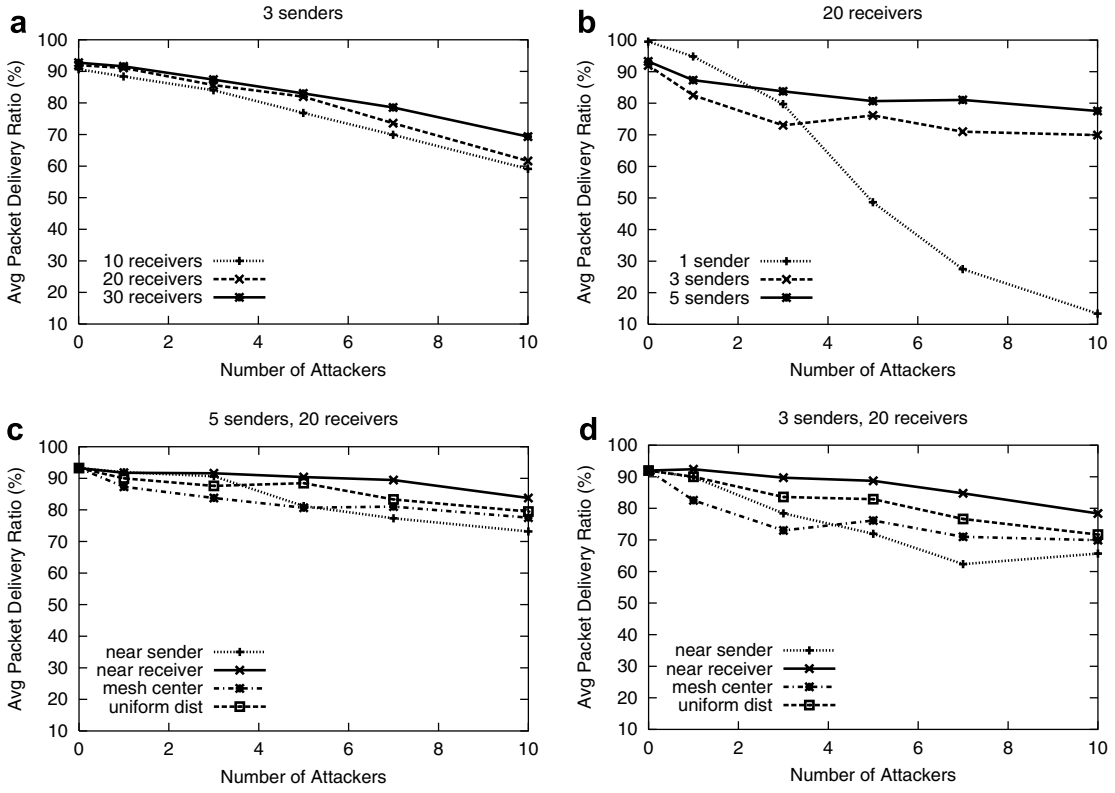

Fig. 6. Effects of blackhole attacks on average packet delivery ratios. (a) Different numbers of receivers. (b) Different numbers of senders. (c) Different attack positions. (d) Different attack positions.

one, three and five, respectively. Fig. 6(b) shows the PDRs for one, three and five senders as a function of the number of attackers.

As we would expect, the higher the number of attackers, the lower the PDR. However, the PDR in the case of one sender declines dramatically to 14% as the number of attackers increases to 10 nodes. On the other hand, the PDRs of the 3-sender and 5-sender groups decrease to only 70% and 78%, respectively. This indicates that multicast sessions with a higher number of senders performs better under blackhole attacks, thanks to a higher number of alternative routing paths in the mesh. If a data packet is dropped by an attacker, a duplicate copy of this packet may still be delivered to the destination(s) successfully via other adversary-free paths.

5.3. Blackhole attacks: attack positions

The multicast group used in this set of experiments has five senders and 20 receivers. The attackers were positioned according to the four scenarios described in Section 3.1: near the senders, near the receivers, around the mesh center, and uniformly distributed over the whole network. Fig. 6(c) shows the packet delivery ratios as a function of the number of attackers.

The results show that the near-receiver area is the least harmful attack position in all cases: the PDRs resulting from this position are always higher than those from the other three cases. The most damaging attack position when the number of attackers is high is the near-sender area: when the number of attackers is greater than five, the near-sender position incurs the lowest PDRs. This observation can be explained as follows. In a multicast mesh, although a data packet is dropped by an attacker, a duplicate version of the packet may still be able to reach its destination via another adversary-free path. An attacker that discard packets in the vicinity of a sender effectively prevents the packets transmitted by this sender from being duplicated any further, resulting in a high packet loss rate.

Theoretically, the near-sender area is the most damaging attack position as explained above. However, when the number of attackers is less than the

number of senders, the attackers cannot cover the vicinity of *all* the senders, especially when the senders spread out over the network. In this case, it is best for the attackers to be at the center of the multicast mesh in order to intercept packets from as many senders as possible. The graph in Fig. 6(c) shows that when the number of attackers is smaller than five, which is the number of senders in this experiment, the mesh center position gives the lowest PDRs. In addition to the case of five senders, we also repeated the experiment with three senders, and the results were consistent with the above explanation (see Fig. 6(d)).

In the uniform distribution experiments, the attackers are distributed all over the routing mesh. Some are thus close to the senders; some, to the receivers; and the others, somewhere between the senders and the receivers. The average PDRs of all these attackers can be considered as the average of the other three cases (i.e., near-sender, near-receiver, and around the mesh center). Therefore, the uniform distribution experiments offer higher PDRs than the near-sender and mesh center experiments in almost all cases, but lower PDRs than the near-receiver experiments.

5.4. Blackhole attacks: summary

A multicast group with a high number of senders and/or a high number of receivers is more resilient to blackhole attacks due to better path redundancy. In contrast, a multicast group with only one sender suffers severely from blackhole attacks, having the PDR dropping from 99% to as low as 14% in our experiment. With regard to attack positions, the multicast mesh center is the strongest attack position if the number of attackers is less than the number of multicast senders. Otherwise, areas near the senders are the most damaging attack positions.

6. Neighbor attacks

The goal of neighbor attackers is to disrupt multicast routes by making two nodes that are in fact out of each other's communication range believe that they can communicate directly with each other. If these two nodes are part of the routing mesh, the data packets that they exchange will be lost because there is no actual connection between them. A neighbor attack is similar to a blackhole attack in the sense that they both prevent data packets from being delivered to the destination(s). The difference is that a blackhole attacker complies with the routing protocol but later drops the data packets it is supposed to forward. A neighbor attacker, on the other hand, violates the routing protocol and does not need to involve itself later in the packet dropping process, since the packets will be lost eventually due to the fake links.

We ran experiments with neighbor attacks implemented, and used the same simulation setting as that used for blackhole attacks (Section 5). The results are shown in Fig. 7, and very similar to those obtained from the blackhole attack experiments. The reason is that breaking a route at a forwarding node in a neighbor attack can be considered as dropping data packets at that node in a blackhole attack.

7. Jellyfish attacks

In this set of experiments, a jellyfish attacker first implements rushing attacks to gain access to the routing mesh. If successful, it then delays all data packets it receives for a random period of time ranging from zero to 10 s before forwarding them.

We measured the PDR, packet end-to-end delay, delay jitter and throughput of multicast sessions under jellyfish attacks when the number of multicast senders, the number of multicast receivers, the number of attackers, and the attack positions varied. We observed that, given the same number of multicast senders and the same number of multicast receivers, the number of attackers and their positions did not affect the average PDR or the average throughput. In other words, jellyfish attacks have no effect on the PDR or the throughput of a multicast group. The reason is that jellyfish attackers do not drop packets, and thus have no impact on the PDR. Jellyfish attackers delay almost all the packets, and thus do not effectively affect the time to deliver the whole file (i.e., the time interval between delivering the first and the last packets of the file). Therefore this kind of attack does not impact the throughput of a multicast group either. We do not show the results for recorded PDRs and throughputs because they are straightforward as just stated.

The graphs in Figs. 8–10 show the average packet end-to-end delay and the average delay jitter as functions of the number of attackers, when the number of multicast receivers, the number of multicast senders, and the attack positions vary, respectively. We can see that in almost all cases, as the number of attackers increases, the average packet

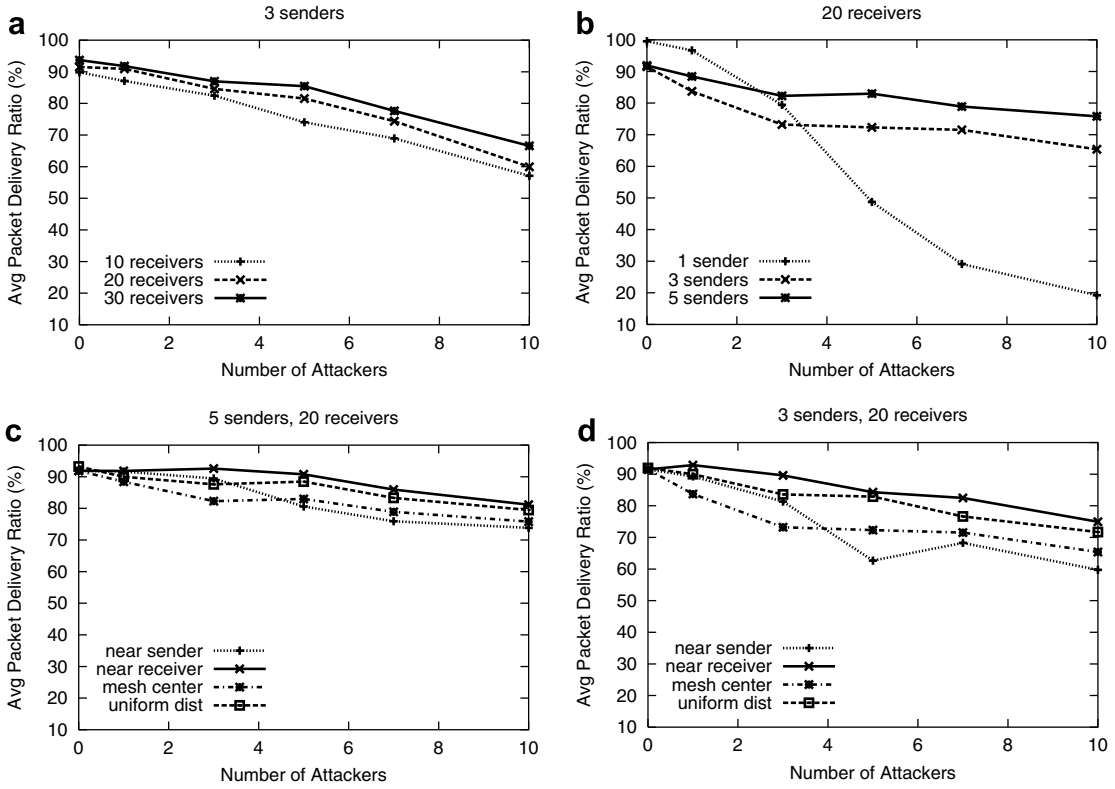

Fig. 7. Effects of neighbor attacks on average packet delivery ratios. (a) Different numbers of multicast receivers. (b) Different numbers of multicast senders. (c) Different attack positions. (d) Different attack positions.

end-to-end delay also increases. The more attackers there are, the more delay they can add to packet traveling time, as expected. The average delay jitter also increases as the number of attackers goes up. Since jellyfish attackers delay data packets for *random* amounts of time, the more attackers there are, the more randomness they add to the arrival times of data packets at the destinations, resulting in higher jitter values.

7.1. Jellyfish attacks: number of receivers

The multicast group used in this experiment has three senders, and the number of receivers is set to 10, 20 and 30 nodes. Fig. 8 shows the average end-to-end delay, and delay jitter as a function of the number of attackers. As the number of attackers increases from zero to 10, the average end-to-end delay increases from 0.1 s to nearly 0.8 s

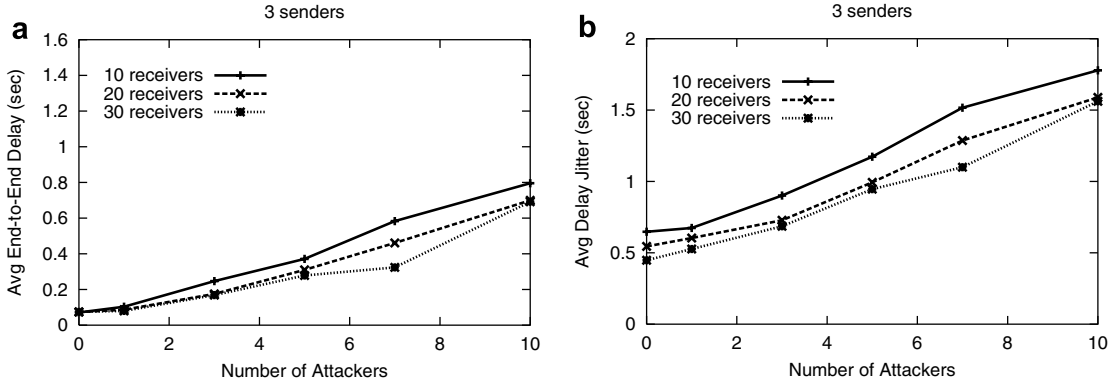

Fig. 8. Effects of jellyfish attacks with different numbers of multicast receivers. (a) Average end-to-end delay. (b) Average delay jitter.

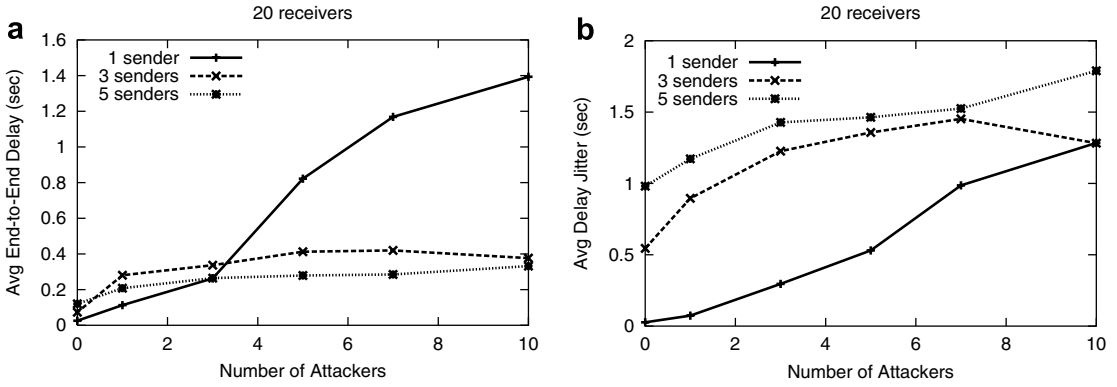

Fig. 9. Effects of jellyfish attacks with different numbers of multicast senders. (a) Average end-to-end delay. (b) Average delay jitter.

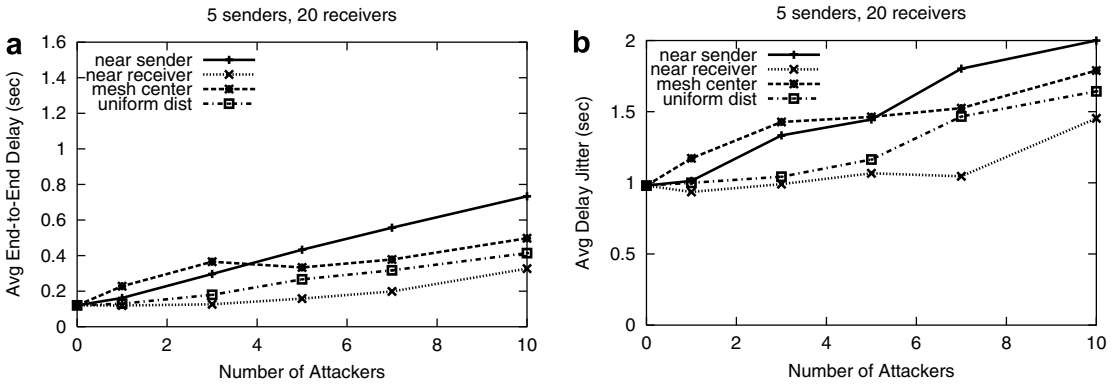

Fig. 10. Effects of jellyfish attacks with different attack positions. (a) Average end-to-end delay. (b) Average delay jitter.

(Fig. 8(a)), and the average delay jitter increases from about 0.5 s to more than 1.5 s (Fig. 8(b)).

Fig. 8(a) also shows that given the same number of attackers, the higher the number of receivers, the lower the average end-to-end delay. As the number of receivers goes up, more routes are created in the routing mesh. If a packet is delayed on one path, another copy of the packet may take another path and arrive at a receiver earlier than the delayed copy (which will be discarded by the receiver as a duplicate). Thus more receivers help to lower the average end-to-end delay. This is also the case for the average delay jitter (Fig. 8(b)).

7.2. Jellyfish attacks: number of senders

In this set of experiments, the number of multicast senders is varied between one and five. The number of multicast receivers is fixed at 20 nodes. The average end-to-end delay and average delay jitter increase rapidly with increasing number of attackers as shown in Fig. 9(a) and (b), respectively.

The results in Fig. 9(a) also show that in most cases a multicast session that has a higher number of senders experiences lower average end-to-end delays. The more senders there are, the more routes are created in the routing mesh. This allows more packets to travel on alternative adversary-free paths to arrive at the destinations earlier than those captured by the attackers, as explained in the previous section.

In the case of delay jitter, the higher the number of senders, the higher the average delay jitter, as illustrated by the graph in Fig. 9(b). The reason is due to the variation of packet sending times at multiple senders. Consider an example in which the total sending rate of all the senders, i.e., the traffic load, is 10 packets/s. If there is only one sender in the multicast group, there will be a smooth stream of packets transmitted at the rate of one packet per 1/10 s. If there are two senders in the multicast group, each will transmit at a rate of 5 packets/s so that the total sending rate is 10 packets/s. However the gap between packets sent by the two

sources is no longer 1/10 s because the sources do not transmit packets in synchronization with each other. This creates more variation in the inter-arrival times of packets at the intermediate nodes (the routers) and thus at the destinations, resulting in higher delay jitter. In general, the more senders there are, the higher the delay jitter is (Fig. 9(b)).

7.3. Jellyfish attacks: attack positions

This experiment studies the effects of jellyfish attacks on the packet end-to-end delay and the delay jitter in the four cases as mentioned in Section 3.1: near the senders, near the receivers, around the mesh center, and uniformly distributed over the entire network area. The multicast group used in this experiment consists of five senders and 20 receivers. Fig. 10 displays the simulation results. We observe that the average end-to-end delay and the average delay jitter of the multicast session increase considerably as the number of jellyfish attackers increases from zero to 10 (Fig. 10(a) and (b)), similarly to the results given in the previous sections.

As mentioned before, while a data packet is being delayed by an attacker, it is possible that a duplicate copy of the packet takes another route, and eventually arrives at the destination before the delayed copy. If a data packet is delayed midway towards the destination, then all copies duplicated from this packet will also experience the delay. On the other hand, if a data packet is delayed in the vicinity of the destination, (e.g., one hop away from the destination), then only this copy of the packet suffers from the delay. Therefore the near-receiver attack position results in the lowest end-to-end delay and the lowest delay jitter in all cases.

According to the above explanation, the near-sender area is the most damaging attack position, because an attacker immediately delays all original packets and thus their duplicate copies. This is especially true when the number of attackers is high (five or higher in the graphs in Fig. 10(a) and (b)). However, when the number of attackers is lower than the number of senders (one and three attackers in the graphs in Fig. 10(a) and (b)), the most damaging attack position is the mesh center, causing the highest end-to-end delay and the highest delay jitter. This position allows a small group of attackers to intercept packets from as many senders as possible, as explained in detail in Section 5.3.

In the uniform distribution experiments, the attackers are scattered over the entire routing mesh. Some of them stay close to the senders; some, to the receivers; and the others, somewhere between the senders and the receivers. The average end-to-end delays obtained from these experiments can be regarded as the average taken over the other three cases (i.e., near-sender, near-receiver, and around the mesh center). The graphs indeed show that the uniform distribution experiments result in higher end-to-end delays than the near-receiver experiments, but lower delays than the near-sender and mesh center experiments. The same observation and explanation apply to the delay jitter metric.

7.4. Jellyfish attacks: summary

Jellyfish attacks affect the packet end-to-end delay and the delay jitter, but not the packet delivery ratio or the throughput. As the number of attackers increases, the end-to-end delay and the delay jitter also increase. A multicast group with a large number of senders and/or a large number of receivers has low end-to-end delay because of a high number of alternative routes. However, a multicast group with a large number of senders may experience higher delay jitter due to asynchronous transmissions between multiple senders. Similar to the case of blackhole attacks, the mesh center is the strongest attacking position if the number of attackers is small compared to the number of senders. Otherwise, the most damaging attack positions are those close to the senders.

8. Related work

Attacks against unicast communications in MANETs have been studied to some extent [15–17]. Gupta et al. studied the weaknesses of the 802.11 MAC protocol by measuring the throughput of attacked nodes under flooding attacks, a type of resource consumption attack [15]. The study showed that the throughputs of all the victim nodes that were one hop away from the attackers degraded to almost zero. The damage was less severe if the attackers were two or more hops away. Other scenarios such as collusions between attackers with different attack rates were also considered.

Ning and Sun [16] examined the vulnerability of the unicast routing protocol AODV [10] under the following types of attacks. route disruption (e.g., neighbor attack), route invasion (e.g., rushing

attack), and resource consumption. Their study considered a network of size 1000 m × 1000 m having five to 20 mobile nodes. The experiments simulated one sender, one receiver, and one attacker. The results showed that under route disruption attacks, the packet delivery ratio dropped to 0%, as opposed to over 75% in the no-attack case. Under route invasion attacks, the percentage of data packets that were intercepted by the attacker increased from 0% to over 50%.

Awerbuch et al. [17] compared the performance of AODV to that of ODSBR [11], a secure unicast routing protocol, under rushing, wormhole, and blackhole attacks. The results indicate that a routing protocol that does not use a duplicate suppression mechanism is not vulnerable to rushing attacks. The study also showed that the center area of a network is the most effective attack position. This finding is consistent with our observation that the center of a multicast routing mesh is one of the most damaging attack positions.

In addition to the above studies, many security protocols have been proposed to provide security features to unicast routing protocols. For example, SEAD [18] was proposed to secure the unicast protocol DSDV [12]. Ariadne [3] was designed to support secure routing in DSR [13]. ARAN [19] and SAODV [20] are two secure versions of AODV [10]. SRP [21] is a new protocol that provides both routing and security mechanisms.

Although security issues for unicast have been addressed by many researchers, research on multicast security in MANETs is still at a very early stage due to several challenges specific to multicast operations such as group key management, member access control, and secure routing. Given that a mature secure multicast routing protocol has not been developed yet, an intrusion detection system may be deployed as an alternative solution to provide protection for multicast in MANETs. Our study in this paper can be used to build an intrusion detection model for that purpose.

9. Conclusion

The performance of a multicast session in a MANET under attack depends heavily on many factors such as the number of multicast senders, the number of multicast receivers, the number of attackers as well as their positions. Our simulation results confirm an intuitive claim: the more attackers there are in the network, the more damage they inflict on a multicast session in terms of packet delivery ratio (blackhole attack and neighbor attack), or delay and delay jitter (jellyfish attack).

We arrived at the following conclusions regarding rushing attack. Rushing attackers have a higher chance of gaining access to the forwarding group when the number of multicast senders is small and/or the number of multicast receivers is large. To maximize their collective success rate, the attackers should gather themselves in a group and stay near the receivers or around the mesh center. Attackers located close to the receivers have the highest success rates. However when the number of attackers is small compared to the number of multicast receivers, they should stay at the center of the multicast mesh to increase their chances of being selected into the forwarding group.

We also note that although the operations of blackhole attacks and neighbor attacks are different, they both cause the same degree of damage to the performance of a multicast group in terms of packet loss rate and throughput. Jellyfish attacks do not affect the packet delivery ratio or the throughput of a multicast group, but they severely increase the packet end-to-end delay and delay jitter.

Although their attacking mechanisms are different, blackhole and neighbor attacks affect the packet delivery ratio similarly to the way jellyfish attacks affect the end-to-end delay. In particular, the performance of a small group will degrade seriously under these types of attacks. A large group with a high number of senders and/or a high number of receivers can sustain good performance under these types of attacks due to more alternative paths in the routing mesh. With respect to attack positions, areas near the senders are the most damaging positions since the original packets are intercepted early, before being duplicated at branch points. However, when the number of attackers is smaller than the number of multicast senders, the mesh center is the strongest attack position, causing the most packet losses (blackhole and neighbor attacks) or the longest delay (jellyfish attack).

To the best of our knowledge, our work presented in this paper is the first that studies the vulnerability and the performance of multicast in MANETs under various kinds of security threats.

References

[1] Y.C. Hu, A. Perrig, D.B. Johnson, Rushing attacks and defense in wireless ad hoc network routing protocols, in:

Proceedings of ACM WiSe 2003, San Diego, CA, September 2003.
[2] I. Aad, J.P. Hubaux, E.W. Knightly, Denial of service resilience in ad hoc networks, in: Proceedings of ACM MobiCom 2004, Philadelphia, PA, September 2004.
[3] Y.C. Hu, A. Perrig, D.B. Johnson, Ariadne: A secure on-demand routing protocol for ad hoc networks, in: Proceedings of ACM MobiCom 2002, Atlanta, Georgia, September 2002.
[4] S.J. Lee, W. Su, M. Gerla, On-Demand Multicast Routing Protocol in Multihop Wireless Mobile Networks, ACM/Kluwer Mobile Networks and Applications 7 (6) (2002) 441–453.
[5] J.J. Garcia-Luna-Aceves, E.L. Madruga, The Core-Assisted Mesh Protocol, IEEE Journal on Selected Areas in Communications 17 (8) (1999) 1380–1994.
[6] S.K. Das, B.S. Manoj, C. Siva Ram Murthy, A dynamic core-based multicast routing protocol for ad hoc wireless networks, in: Proceedings of ACM MOBIHOC 2002, June 2002, pp. 24–35.
[7] S. Lee, C. Kim, Neighbor supporting ad hoc multicast routing protocol, in: Proceedings of ACM MOBIHOC 2000, August 2000, pp. 37–50.
[8] E.M. Royer, C.E. Perkins, Multicast operation of the ad hoc on-demand distance vector routing protocol, in: Proceedings of MobiCom'99, Seattle, WA, August 1999.
[9] C.W. Wu, Y.C. Tay, C.K. Toh, Ad hoc multicast routing protocol utilizing increasing id-numbers (amris) functional specification, Internet draft, work in progress, draft-ietf-manet-amris-spec-00.txt, November 1998.
[10] C.E. Perkins, E.M. Royer, S.R. Das, Ad hoc on demand distance vector (AODV) routing, in: Proceedings of IEEE WMCSA'99, New Orleans, LA, February 1999.
[11] B. Awerbuch, D. Holmer, C. Nita-Rotaru, H. Rubens, An on-demand secure routing protocol resilient to byzantine failures, in: Proceedings of ACM WiSe 2002, Atlanta, Georgia, September 2002.
[12] C.E. Perkins, P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, in: Proceedings of ACM SIGCOMM'94, August 1994.
[13] D.B. Johnson, D.A. Maltz, Y.C. Hu, J.G. Jetcheva, Dynamic Source Routing in Ad Hoc Wireless Networks, Mobile Computing, vol. 5, Kluwer Academic Publishers., 1996, pp. 153–181.
[14] QualNet Simulator, Available from: <http://www.qualnet.com>.
[15] V. Gupta, S. Krishnamurthy, M. Faloutsos, Denial of service attacks at the MAC layer in wireless ad hoc networks, in: Proceedings of IEEE MILCOM'02, 2002.
[16] P. Ning, K. Sun, How to misuse AODV: a case study of insider attacks against mobile ad-hoc routing protocols, in: Proceedings of the 4th Annual IEEE Information Assurance Workshop, West Point, June 2003.
[17] B.Awerbuch, D. Holmer, C. Nita-Rotaru, H. Rubens, Mitigating byzantine attacks in ad hoc wireless networks, Technical Report, March 2004.
[18] Y.C. Hu, D.B. Johnson, A. Perrig, SEAD: secure efficient distance vector routing for mobile wireless ad hoc networks, in: Proceedings of IEEE WMCSA'02, June 2002.
[19] K. Sanzgiri, B. Dahill, B.N. Levine, C. Shields, E.M. Belding-Royer, A secure routing protocol for ad hoc networks, in: Proceedings of IEEE ICNP'02, Paris, France, November 2002.
[20] M. Zapata, N. Asokan, Securing ad hoc routing protocols, in: Proceedings of ACM WiSe 2002, Atlanta, GA, September 2002.
[21] P. Papadimitratos, Z.J. Hass, Secure routing for mobile ad hoc networks, in: Proceedings of CNDS'02, San Antonio, TX, January 2002.
[22] C.-C. Chiang, M. Gerla, L. Zhang, Forwarding Group Multicast Protocol (FGMP) for Multihop, Mobile Wireless Networks, AJ. Cluster Comp, Special Issue on Mobile Computing 1 (2) (1998) 187–196.
[23] J.G. Jetcheva, D.B. Johnson, Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks, in: Proceedings of ACM MobiHoc'01, Long Beach, CA, October 2001.
[24] T. Ozaki, J.B. Kim, T. Suda, Bandwidth efficient multicast routing protocol for ad hoc networks, in: Proceedings of IEEE ICCCN'99, October 1999, pp. 10–17.
[25] D. Johnson, D. Maltz, Dynamic Source Routing in Ad Hoc Wireless Networks, Mobile Computing, Kluwer Academic Publishers., Norwell, MA, 1996, pp. 153–181.
[26] T.S. Rappaport, L.B. Milstein, Effects of Radio Propagation Path Loss on DS-CDMA Cellular Frequency Reuse Efficiency for the Reverse Channel, IEEE Transactions on Vehicular Technology 41 (3) (1992).

Hoang Lan Nguyen is currently a M.Sc. student in the Department of Computer Science and Engineering at York University (Toronto, Canada). He received his B.E degree with the highest honors in Telecommunications in 2003 from University of Wollongong (NSW, Australia). Before joining York University, he worked for Australia Nortel Networks for half a year. His current research interests include wireless communications, multicast and network security.

Uyen Trang Nguyen received her Bachelor of Computer Science and Master of Computer Science degrees in 1993 and 1997 from Concordia University (Montreal, Canada). She completed her Ph.D. degree at University of Toronto (Toronto, Canada) in 2003. From 1995 to 1997 she was a software engineer at Nortel Networks (Montreal, Canada). She joined the Department of Computer Science and Engineering at York University (Toronto, Canada) in 2002 and is currently an assistant professor. Her research interests are in the areas of wireless ad-hoc networks, multipoint communications, and multimedia applications.