

A Study of Security Attacks on Multicast in Mobile Ad Hoc Networks

Hoang Lan Nguyen
lan@cs.yorku.ca

Department of Computer Science and Engineering
York University

Master's Thesis Defense
Supervisor: Professor Uyen Trang Nguyen



Outline

Introduction

- Overview of Mobile Ad Hoc Networks
- Multicast Communication in Mobile Ad Hoc Networks
- Security Vulnerabilities in Mobile Ad Hoc Networks

Motivation

Our Study

- Our Contributions
- Security Attacks That We Studied
- Experimental Configuration
- Results & Observations

Conclusion



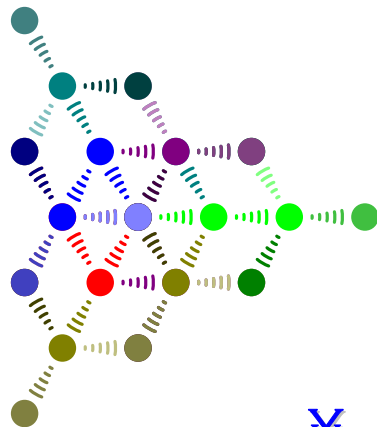
What is Mobile Ad Hoc Network (MANET) ?

Definition

A system of mobile nodes connected with each other via wireless medium **without** infrastructure support.

Applications

- ▶ Military battlefield
- ▶ Emergency rescue
- ▶ Vehicular communication
- ▶ File sharing in a conference or classroom
- ▶ Outdoor Internet access on campus



Why Choose Mobile Ad Hoc Network ?

Pros

- ▶ Easy deployment
 - ▶ Cost effective
 - ▶ Time effective
- ▶ Better reachability
- ▶ Wider accessibility
- ▶ **Network of the future**

Cons

- ▶ Immature
- ▶ Flow and congestion problems
- ▶ Security issues
- ▶ Interoperability issue between wireless vendors



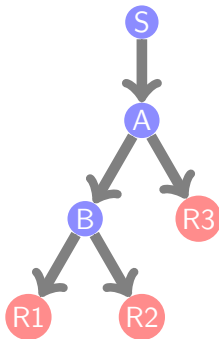
What is Multicasting ?

Definition

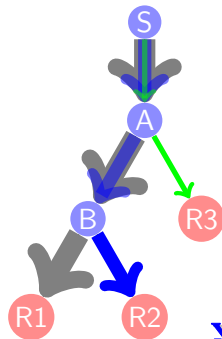
Multicast is a form of **one-to-many** or **many-to-many** communications where data can be delivered to multiple receivers simultaneously.

- ▶ Data are delivered at each link only once
- ▶ Data are duplicated and splitted at branch points

Multicast



Unicast



Classification of Multicast Routing Protocols

Tree-Based Multicast

Single path between a source and a destination

- ▶ Pros: Efficient
- ▶ Cons: Less reliable

Example

MAODV, AMRIS, AMRoute

Mesh-Based Multicast

Multiple paths between a source and a destination

- ▶ Pros: Better connectivity
- ▶ Cons: High overhead

Example

ODMRP, CAMP, MCEDAR

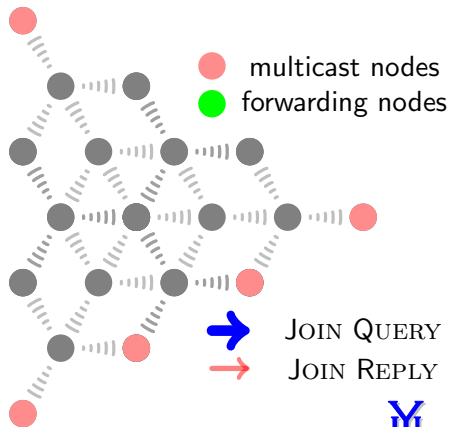


The On-Demand Multicast Routing Protocol (ODMRP)

ODMRP^a is a mesh-based multicast protocol:

- ▶ uses the concept of **forwarding group**.
- ▶ floods JOIN QUERY packets
- ▶ JOIN REPLY packets establish multicast routes & forwarding members
- ▶ only forwarding members forward data packets

^aS.J. Lee et. al., On-Demand Multicast Routing Protocol, Proceedings of IEEE WCNC'99

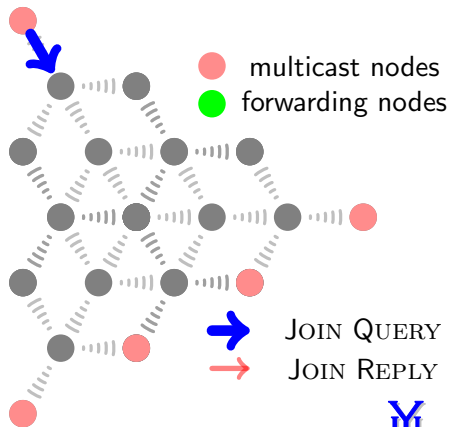


The On-Demand Multicast Routing Protocol (ODMRP)

ODMRP^a is a mesh-based multicast protocol:

- ▶ uses the concept of **forwarding group**.
- ▶ floods JOIN QUERY packets
- ▶ JOIN REPLY packets establish multicast routes & forwarding members
- ▶ only forwarding members forward data packets

^aS.J. Lee et. al., On-Demand Multicast Routing Protocol, Proceedings of IEEE WCNC'99

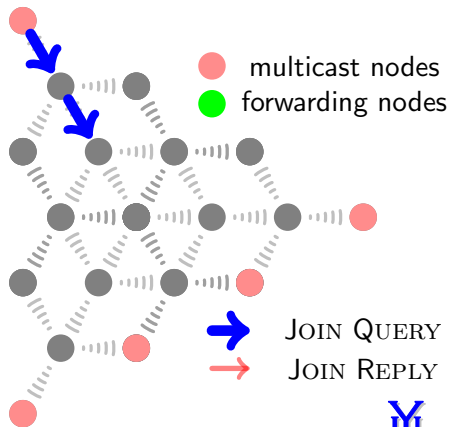


The On-Demand Multicast Routing Protocol (ODMRP)

ODMRP^a is a mesh-based multicast protocol:

- ▶ uses the concept of **forwarding group**.
- ▶ floods JOIN QUERY packets
- ▶ JOIN REPLY packets establish multicast routes & forwarding members
- ▶ only forwarding members forward data packets

^aS.J. Lee et. al., On-Demand Multicast Routing Protocol, Proceedings of IEEE WCNC'99

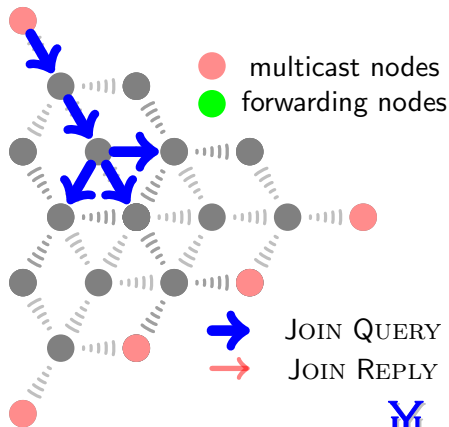


The On-Demand Multicast Routing Protocol (ODMRP)

ODMRP^a is a mesh-based multicast protocol:

- ▶ uses the concept of **forwarding group**.
- ▶ floods JOIN QUERY packets
- ▶ JOIN REPLY packets establish multicast routes & forwarding members
- ▶ only forwarding members forward data packets

^aS.J. Lee et. al., On-Demand Multicast Routing Protocol, Proceedings of IEEE WCNC'99

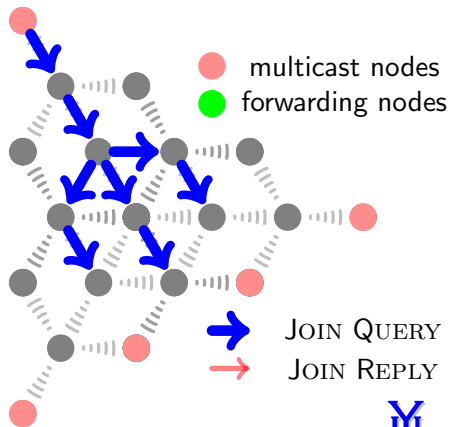


The On-Demand Multicast Routing Protocol (ODMRP)

ODMRP^a is a mesh-based multicast protocol:

- ▶ uses the concept of **forwarding group**.
- ▶ floods JOIN QUERY packets
- ▶ JOIN REPLY packets establish multicast routes & forwarding members
- ▶ only forwarding members forward data packets

^aS.J. Lee et. al., On-Demand Multicast Routing Protocol, Proceedings of IEEE WCNC'99

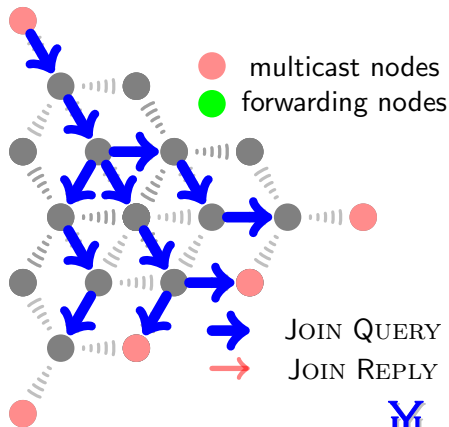


The On-Demand Multicast Routing Protocol (ODMRP)

ODMRP^a is a mesh-based multicast protocol:

- ▶ uses the concept of **forwarding group**.
- ▶ floods JOIN QUERY packets
- ▶ JOIN REPLY packets establish multicast routes & forwarding members
- ▶ only forwarding members forward data packets

^aS.J. Lee et. al., On-Demand Multicast Routing Protocol, Proceedings of IEEE WCNC'99

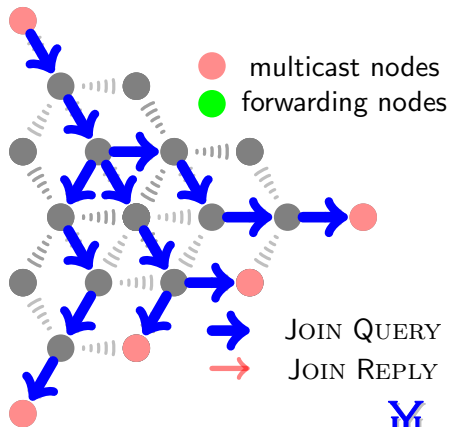


The On-Demand Multicast Routing Protocol (ODMRP)

ODMRP^a is a mesh-based multicast protocol:

- ▶ uses the concept of **forwarding group**.
- ▶ floods JOIN QUERY packets
- ▶ JOIN REPLY packets establish multicast routes & forwarding members
- ▶ only forwarding members forward data packets

^aS.J. Lee et. al., On-Demand Multicast Routing Protocol, Proceedings of IEEE WCNC'99

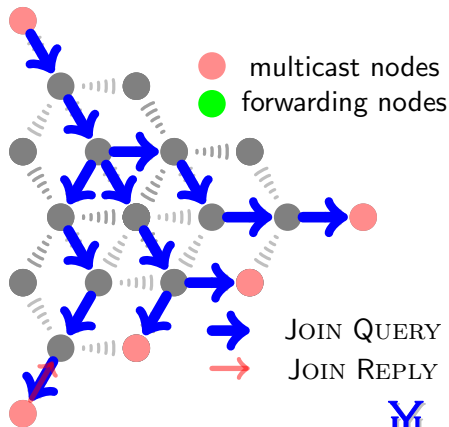


The On-Demand Multicast Routing Protocol (ODMRP)

ODMRP^a is a mesh-based multicast protocol:

- ▶ uses the concept of **forwarding group**.
- ▶ floods JOIN QUERY packets
- ▶ JOIN REPLY packets establish multicast routes & forwarding members
- ▶ only forwarding members forward data packets

^aS.J. Lee et. al., On-Demand Multicast Routing Protocol, Proceedings of IEEE WCNC'99

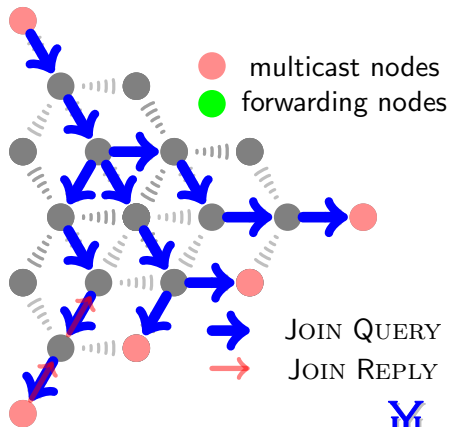


The On-Demand Multicast Routing Protocol (ODMRP)

ODMRP^a is a mesh-based multicast protocol:

- ▶ uses the concept of **forwarding group**.
- ▶ floods JOIN QUERY packets
- ▶ JOIN REPLY packets establish multicast routes & forwarding members
- ▶ only forwarding members forward data packets

^aS.J. Lee et. al., On-Demand Multicast Routing Protocol, Proceedings of IEEE WCNC'99

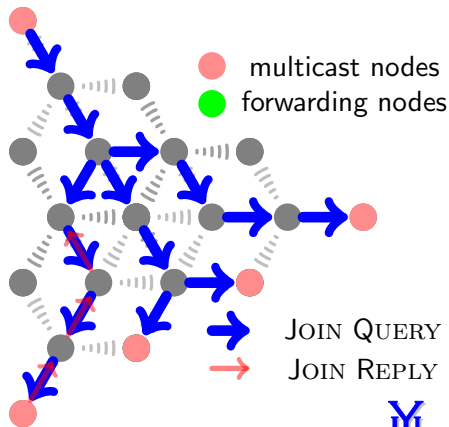


The On-Demand Multicast Routing Protocol (ODMRP)

ODMRP^a is a mesh-based multicast protocol:

- ▶ uses the concept of **forwarding group**.
- ▶ floods JOIN QUERY packets
- ▶ JOIN REPLY packets establish multicast routes & forwarding members
- ▶ only forwarding members forward data packets

^aS.J. Lee et. al., On-Demand Multicast Routing Protocol, Proceedings of IEEE WCNC'99

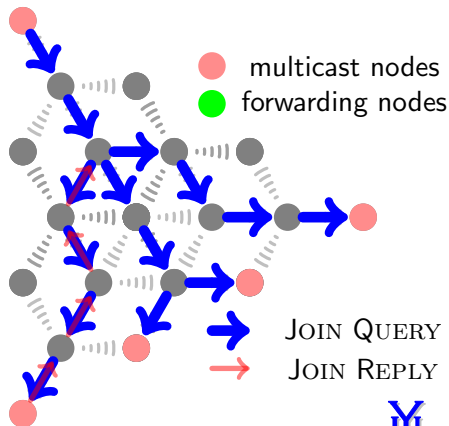


The On-Demand Multicast Routing Protocol (ODMRP)

ODMRP^a is a mesh-based multicast protocol:

- ▶ uses the concept of **forwarding group**.
- ▶ floods JOIN QUERY packets
- ▶ JOIN REPLY packets establish multicast routes & forwarding members
- ▶ only forwarding members forward data packets

^aS.J. Lee et. al., On-Demand Multicast Routing Protocol, Proceedings of IEEE WCNC'99

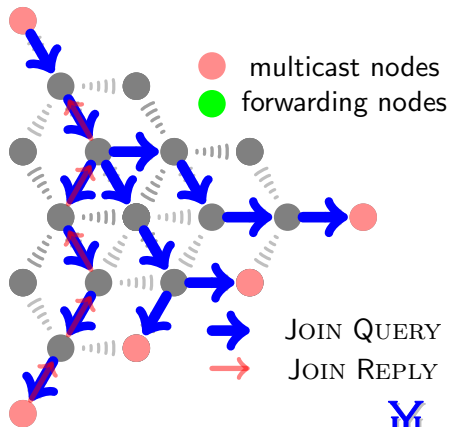


The On-Demand Multicast Routing Protocol (ODMRP)

ODMRP^a is a mesh-based multicast protocol:

- ▶ uses the concept of **forwarding group**.
- ▶ floods JOIN QUERY packets
- ▶ JOIN REPLY packets establish multicast routes & forwarding members
- ▶ only forwarding members forward data packets

^aS.J. Lee et. al., On-Demand Multicast Routing Protocol, Proceedings of IEEE WCNC'99

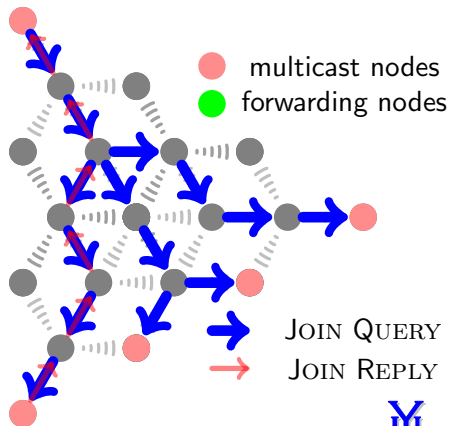


The On-Demand Multicast Routing Protocol (ODMRP)

ODMRP^a is a mesh-based multicast protocol:

- ▶ uses the concept of **forwarding group**.
- ▶ floods JOIN QUERY packets
- ▶ JOIN REPLY packets establish multicast routes & forwarding members
- ▶ only forwarding members forward data packets

^aS.J. Lee et. al., On-Demand Multicast Routing Protocol, Proceedings of IEEE WCNC'99

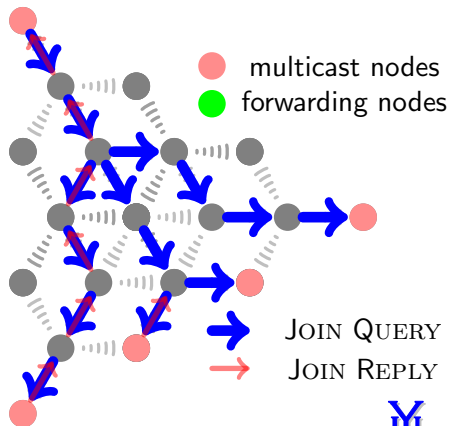


The On-Demand Multicast Routing Protocol (ODMRP)

ODMRP^a is a mesh-based multicast protocol:

- ▶ uses the concept of **forwarding group**.
- ▶ floods JOIN QUERY packets
- ▶ JOIN REPLY packets establish multicast routes & forwarding members
- ▶ only forwarding members forward data packets

^aS.J. Lee et. al., On-Demand Multicast Routing Protocol, Proceedings of IEEE WCNC'99

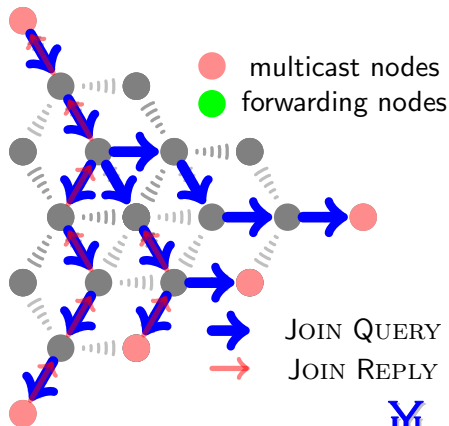


The On-Demand Multicast Routing Protocol (ODMRP)

ODMRP^a is a mesh-based multicast protocol:

- ▶ uses the concept of **forwarding group**.
- ▶ floods JOIN QUERY packets
- ▶ JOIN REPLY packets establish multicast routes & forwarding members
- ▶ only forwarding members forward data packets

^aS.J. Lee et. al., On-Demand Multicast Routing Protocol, Proceedings of IEEE WCNC'99

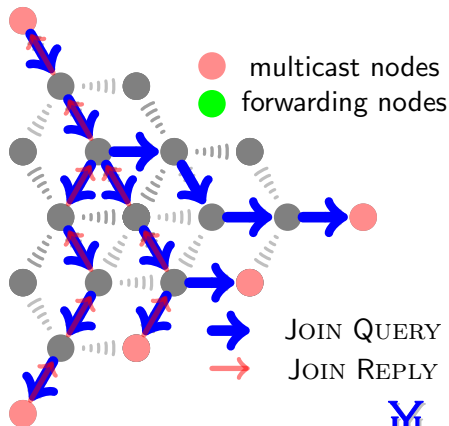


The On-Demand Multicast Routing Protocol (ODMRP)

ODMRP^a is a mesh-based multicast protocol:

- ▶ uses the concept of **forwarding group**.
- ▶ floods JOIN QUERY packets
- ▶ JOIN REPLY packets establish multicast routes & forwarding members
- ▶ only forwarding members forward data packets

^aS.J. Lee et. al., On-Demand Multicast Routing Protocol, Proceedings of IEEE WCNC'99

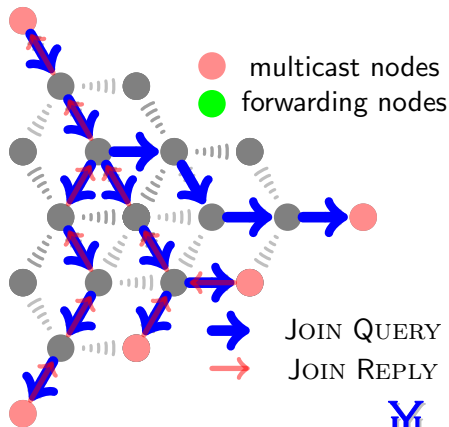


The On-Demand Multicast Routing Protocol (ODMRP)

ODMRP^a is a mesh-based multicast protocol:

- ▶ uses the concept of **forwarding group**.
- ▶ floods JOIN QUERY packets
- ▶ JOIN REPLY packets establish multicast routes & forwarding members
- ▶ only forwarding members forward data packets

^aS.J. Lee et. al., On-Demand Multicast Routing Protocol, Proceedings of IEEE WCNC'99

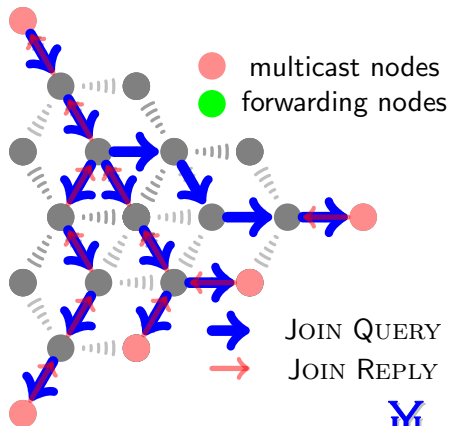


The On-Demand Multicast Routing Protocol (ODMRP)

ODMRP^a is a mesh-based multicast protocol:

- ▶ uses the concept of **forwarding group**.
- ▶ floods JOIN QUERY packets
- ▶ JOIN REPLY packets establish multicast routes & forwarding members
- ▶ only forwarding members forward data packets

^aS.J. Lee et. al., On-Demand Multicast Routing Protocol, Proceedings of IEEE WCNC'99

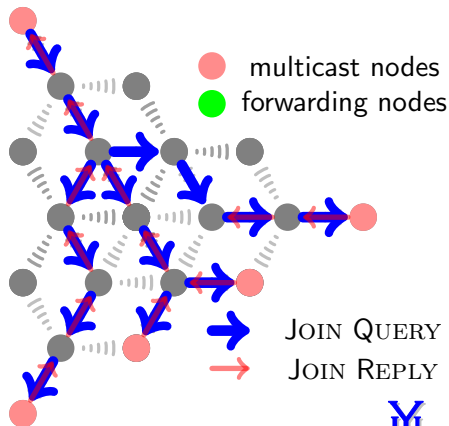


The On-Demand Multicast Routing Protocol (ODMRP)

ODMRP^a is a mesh-based multicast protocol:

- ▶ uses the concept of **forwarding group**.
- ▶ floods JOIN QUERY packets
- ▶ JOIN REPLY packets establish multicast routes & forwarding members
- ▶ only forwarding members forward data packets

^aS.J. Lee et. al., On-Demand Multicast Routing Protocol, Proceedings of IEEE WCNC'99

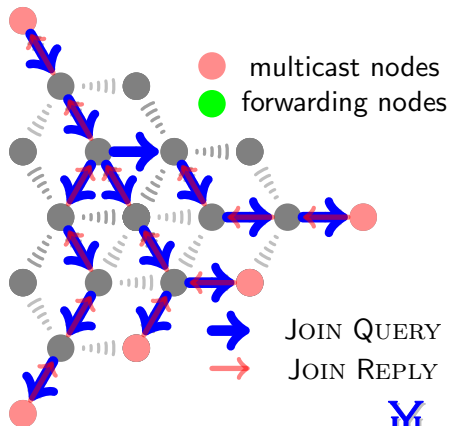


The On-Demand Multicast Routing Protocol (ODMRP)

ODMRP^a is a mesh-based multicast protocol:

- ▶ uses the concept of **forwarding group**.
- ▶ floods JOIN QUERY packets
- ▶ JOIN REPLY packets establish multicast routes & forwarding members
- ▶ only forwarding members forward data packets

^aS.J. Lee et. al., On-Demand Multicast Routing Protocol, Proceedings of IEEE WCNC'99

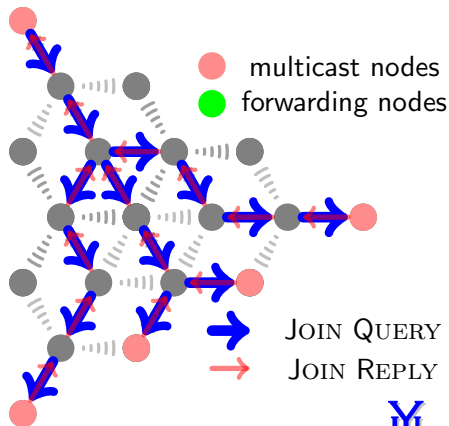


The On-Demand Multicast Routing Protocol (ODMRP)

ODMRP^a is a mesh-based multicast protocol:

- ▶ uses the concept of **forwarding group**.
- ▶ floods JOIN QUERY packets
- ▶ JOIN REPLY packets establish multicast routes & forwarding members
- ▶ only forwarding members forward data packets

^aS.J. Lee et. al., On-Demand Multicast Routing Protocol, Proceedings of IEEE WCNC'99

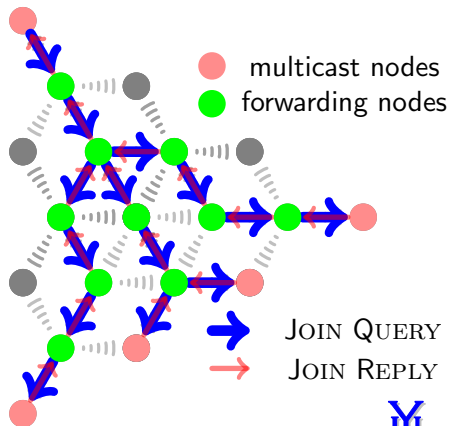


The On-Demand Multicast Routing Protocol (ODMRP)

ODMRP^a is a mesh-based multicast protocol:

- ▶ uses the concept of **forwarding group**.
- ▶ floods JOIN QUERY packets
- ▶ JOIN REPLY packets establish multicast routes & forwarding members
- ▶ only forwarding members forward data packets

^aS.J. Lee et. al., On-Demand Multicast Routing Protocol, Proceedings of IEEE WCNC'99



Security Vulnerabilities in Mobile Ad Hoc Networks

- ▶ Lack of central administration
- ▶ Open system due to wireless nature
- ▶ Lack of trust relationship
- ▶ Limited energy power
- ▶ Physical vulnerability



Previous Security Studies on Mobile Ad Hoc Networks

- ▶ Ning et. al. (2003) and Awerbuch et. al. (2004) studied the vulnerabilities of AODV unicast protocol.
- ▶ Gupta et. al. (2002) studied the denial of service attack at the link layer (one-to-one) in wireless ad hoc networks.
- ▶ Y.C. Hu et. al. (2002) proposed Ariadne to secure DSR unicast protocol.
- ▶ K. Sanzgiri et. al. (2002) and M. Zapata et. al. (2002) proposed ARAN and SAODV respectively, to secure AODV unicast protocol
- ▶ And more ...



Our Motivation

- ▶ Previous security research focused only on **unicast** (one sender to one receiver) communications
- ▶ Noone has studied the vulnerabilities of **multicast** communications yet.
- ▶ Multicast is more complicated due to the involvement of multiple senders and multiple receivers
- ▶ There has been **NO** secure protocols for multicast
- ▶ So comes this thesis “A Study of the Effects of Security Attacks on Multicast in Mobile Ad Hoc Networks”



Our Contributions

- ▶ Show how multicast behaves in various attack scenarios
- ▶ Suggest measures to minimize the effects of some common security attacks on multicast
- ▶ Report attack strategies that would maximize the damage to a multicast session
- ▶ The results presented will be useful for researchers who are working on attack/intrusion detection for multicast protocols



Security Attacks That We Studied

Rushing Attack

Rushing attackers forward routing packets **as quick as possible** to gain access to multicast forwarding group

Neighbor Attack

Make two or more hop away nodes think that they are one-hop away (neighbors)

Blackhole Attack

After successfully invading into the forwarding group, blackhole attackers will **drop** some or all data packets that arrive at them

Jellyfish Attack

After successfully invading into the forwarding group, jellyfish attackers **delay unnecessarily** forwarding data packets for a random amount of time

Implementation of Rushing Attack

Implementation

Rushing attackers forward JOIN QUERY with zero processing delay

```
// if the node is a rushing attacker
if (node->attack_flag == TRUE) {
    // forward JOIN QUERY with zero delay
    NetworkIpSendPacketToMacLayerWithDelay
    (node, msg, DEFAULT_INTERFACE, ANY_DEST, 0);
}
// else forward with a pre-defined non-zero delay
else {
    NetworkIpSendPacketToMacLayerWithDelay
    (node, msg, DEFAULT_INTERFACE, ANY_DEST,
    node->process_delay);
}
```



Implementation of Blackhole Attack

Implementation

Blackhole attackers, who belong to the forwarding group, drop all data packets they receive

```
// if the node is a forwarding member
if (OdmrpLookupFgFlag(mcastAddr, &odmrp->fgFlag))
{
    // if the node is a blackhole attacker
    if (node->attack_flag) {
        // then drop the data packet
        MESSAGE_Free(node, msg);
        return;
    }
}
```



Implementation of Neighbor Attack

Implementation

Neighbor attackers don't update their IP addresses in the last-hop field of the JOIN QUERY packet

```
// if the node is a neighbor attacker
if (node->attack_flag) {
    // then do not update anything
}
// else, for an honest node
else {
    // update the last hop field
    option.lastAddr = NetworkIpGetInterfaceAddress(
        node, DEFAULT_INTERFACE);
}
```



Implementation of Jellyfish Attack

Implementation

Jellyfish attackers delay data packets for a random period of time between 0 and 10 seconds

```
// if the node is a jellyfish attacker
if (node->attack_flag) {
    // generate a random time from 0 to 10
    clocktype jelly_delay = (clocktype)
        (pc_eraud(node->seed) * 10 * SECOND);
    // forward data packet with this additional delay
    NetworkIpSendPacketToMacLayerWithDelay(
        node, msg, DEFAULT_INTERFACE, ANY_DEST,
        node->process_delay + jelly_delay);
}
```



Experimental Configuration

Common parameters used in all experiments

- ▶ Network size: 1000 m x 1000 m
- ▶ Total number of nodes: 50
- ▶ Channel bandwidth: 2 Mbps
- ▶ Mobility speed: 1 m/s
- ▶ Traffic load: 1 pkt/s per source

Performance metrics used for comparison and analysis

- ▶ Average attack success rate
- ▶ Average packet delivery ratio
- ▶ Average throughput
- ▶ Average end-to-end delay
- ▶ Average delay jitter



Average Attack Success Rate of Rushing Attackers

Definition

- ▶ The **attack success rate** of one attacker is the ratio of the number of times the attacker has successfully invaded into the routing mesh over the total number of times the route discovery process is initiated.
- ▶ The **average** rushing attack success rate is the average of the attack success rates taken over all attackers.



Average Packet Delivery Ratio

Definition

- ▶ The **packet delivery ratio** of a receiver is defined as the number of data packets sent by all multicast senders over the number of data packets received by the receivers
- ▶ The **average** packet delivery ratio is the average of the packet delivery ratios taken over all receivers.



Average Throughput

Definition

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



Average End-to-End Delay

Definition

- ▶ The **end-to-end delay** of a packet is defined as the time taken for the packet to travel from the sender to the receiver.
- ▶ The **average** end-to-end delay is the average of the end-to-end delays taken over all packets.



Average Delay Jitter

Definition

- ▶ The **per-source delay jitter** of a receiver is the time difference of the arrival times of two consecutive packets received by the receiver from the same source.
- ▶ The **delay jitter** of a receiver is the average of the per-source delay jitters taken over all sources.
- ▶ The **average** delay jitter is the average of the delay jitters taken over all receivers.



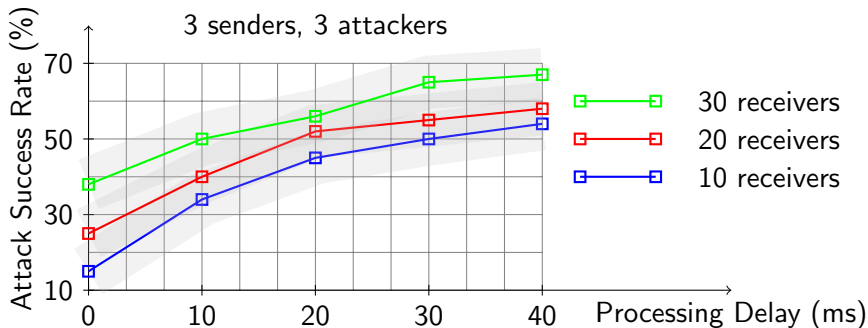
Results & Observations

For each attack, we consider the following factors:

- ▶ Different numbers of multicast receivers: 10, 20, and 30 nodes
- ▶ Different numbers of multicast senders: 1, 3, and 5 nodes
- ▶ Different attack positions:
 - ▶ near the multicast senders
 - ▶ near the multicast receivers
 - ▶ at the network mesh center
 - ▶ uniform distributed over the network
- ▶ Number of attackers: varies from 0 to 10 nodes
- ▶ Processing delay at a non-adversary node: varies from 0 to 40 ms



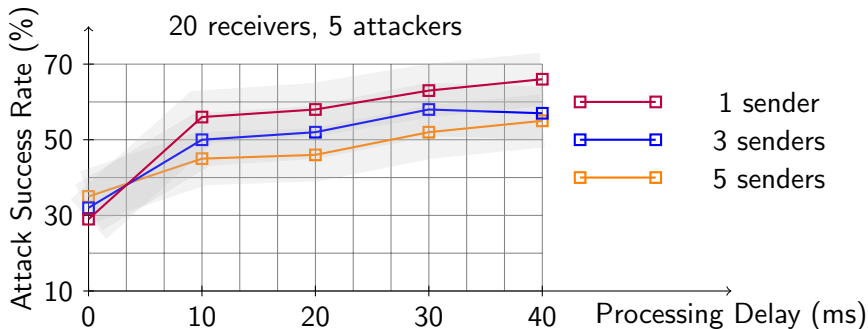
Rushing Attack Success Rate vs. Number of Receivers



1. The longer the processing delay at non-adversary nodes, the higher the attack success rate
2. The higher the number of multicast receivers, the higher the attack success rate



Rushing Attack Success Rate vs. Number of Senders

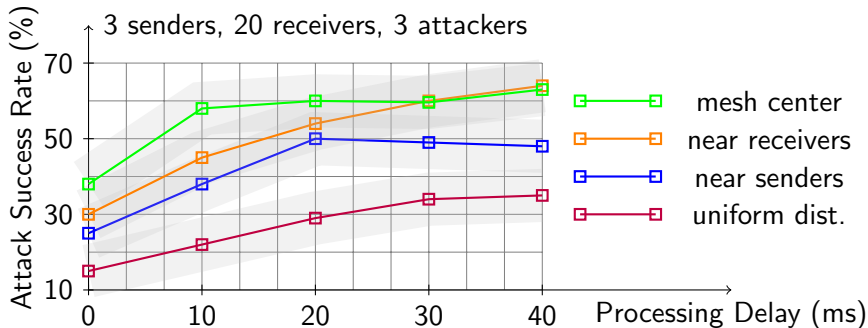


The higher the number of multicast senders, the **smaller** the attack success rate

- ▶ Higher number of multicast senders creates better path redundancy/alternative, thus more non-adversary paths available that are shorter than rushing attack paths



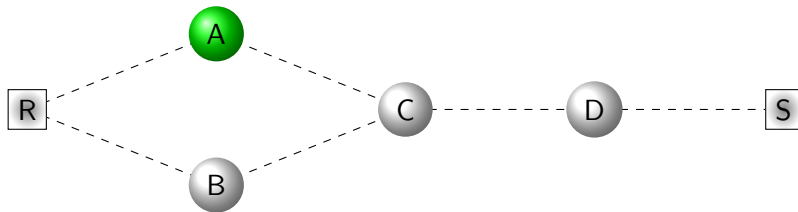
Rushing Attack Success Rate vs. Attack Positions



1. The uniform dist. position is the least efficient attack position
2. The mesh center position has the highest attack success rate
3. However, the near-receiver position is **theoretically** the most successful attack position. **Why ?**



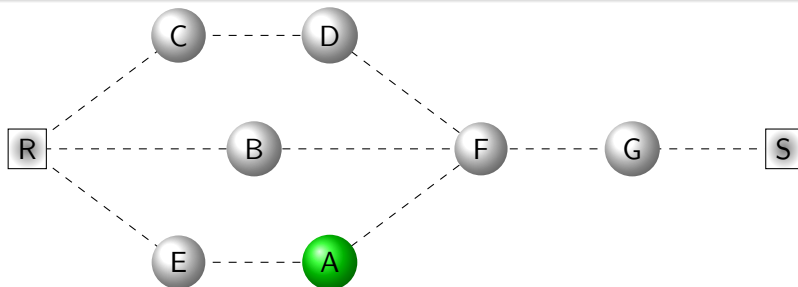
Rushing Attack Position: One-hop from the Receiver



1. Consider two possible paths from S to R: S-D-C-A-R and S-D-C-B-R
2. Both A and B forward JOIN QUERY to R
3. If A is a rushing attacker, it is most likely that A's JOIN QUERY will arrive at R first
4. A will successfully be chosen by R as a forwarding member



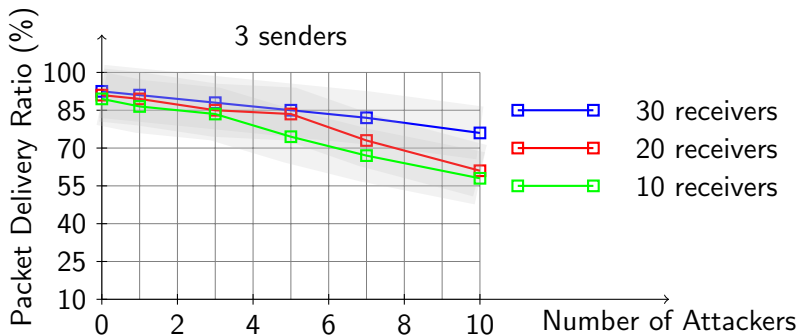
Rushing Attack Position: Two-hop from the Receiver



1. Consider three alternative paths from S to R: S-G-F-A-E-R, S-G-F-D-C-R and S-G-F-B-R
2. JOIN QUERY forwarded by attacker A may still arrive **later** than the other two paths
 - ▶ Due to congestion delay at node E
 - ▶ Path S-G-F-B-R has lesser number of hops



Blackhole Attack: PDR vs. Number of Receivers

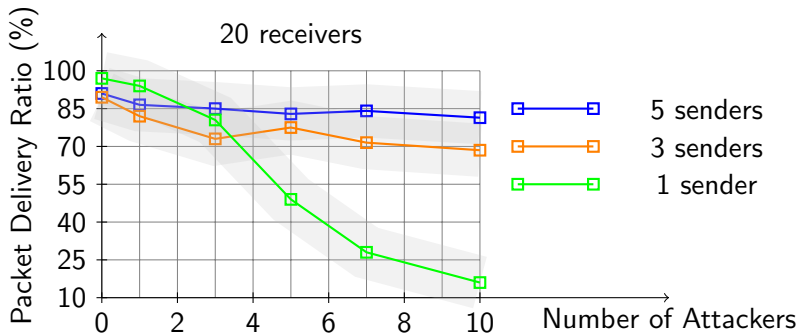


1. The higher the number of attackers, the lower the packet delivery ratio
2. The higher the number of multicast receivers, the higher the packet delivery ratio

► Higher number of receivers also create more alternative paths



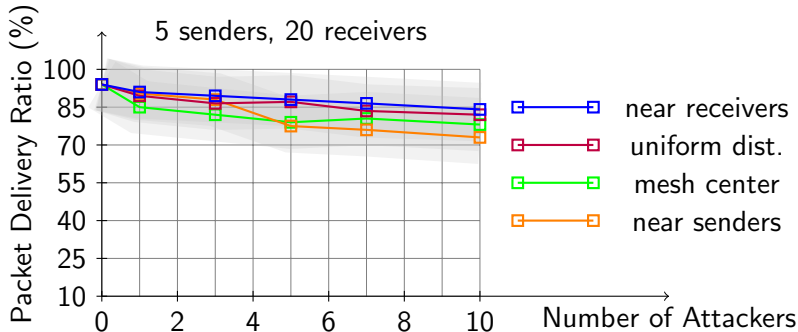
Blackhole Attack: PDR vs. Number of Senders



1. The higher the number of multicast senders, the higher the packet delivery ratio
2. The explanation is similar to the receiver scenario



Blackhole Attack: PDR vs. Attack Positions



1. When the number of attackers is high, the near-sender position is the most damaging position
2. When the number of attackers is small, the mesh-center position is the most effective attack position

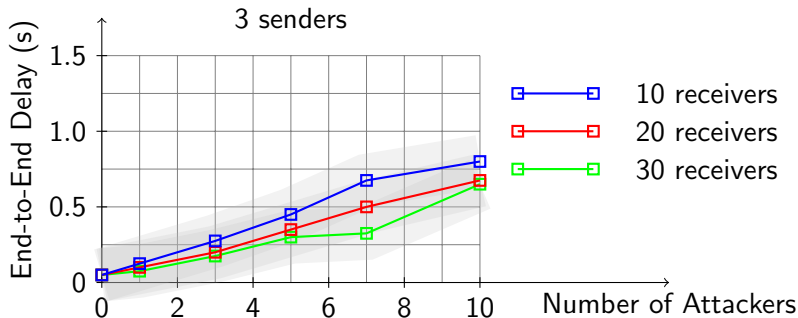


Effects of Neighbor Attack on Multicast Sessions

- ▶ Behaviour is similar to those in blackhole attacks.
- ▶ Because breaking a link can be considered the same as dropping data packets at that link
- ▶ Plots are not shown here



Jellyfish Attack: E2E Delay vs. Number of Receivers

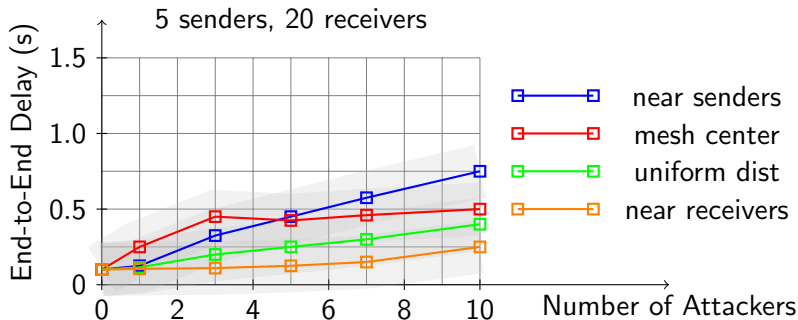


1. The higher the number of attackers, the longer the E2E delay
2. The higher the number of attackers, the smaller the E2E delay
3. The delay jitter graph showed the same behavior
4. The packet delivery ratio and throughput were not affected.

Not shown here



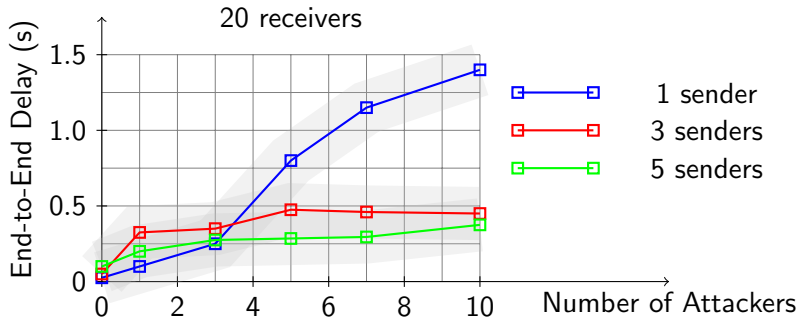
Jellyfish Attack: E2E vs. Attack Positions



1. Similar observations as in the blackhole and neighbor attacks
2. Similar behaviour for the delay jitter. Not shown here.



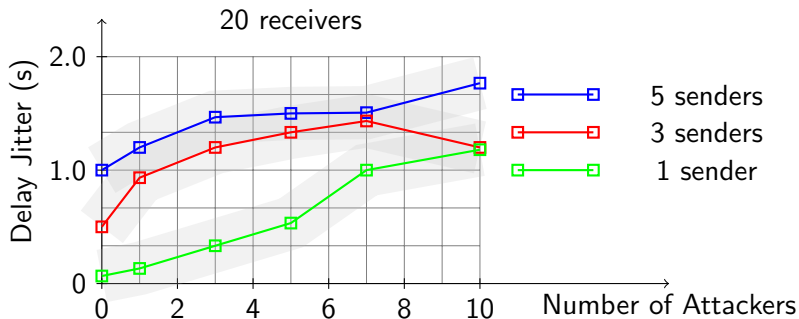
Jellyfish Attack: E2E Delay vs. Number of Senders



1. Same observations as in blackhole and neighbor attacks
2. The higher the number of multicast senders, the smaller the end-to-end delay
3. **However**, it is not the case for the delay jitter



Jellyfish Attack: Delay Jitter vs. Number of Senders



1. The higher the number of multicast senders, the higher the delay jitter
2. Because senders' transmissions are not **synchronized**
 - ▶ The times that data packets from different senders arrive at a receiver vary significantly



Final Remarks 1

- ▶ The performance of a multicast session depends on: the number of multicast senders, the number of multicast receivers, the number of attackers as well as their positions
- ▶ In terms of rushing success rate, a multicast with low number of senders and/or high number of receivers is most vulnerable to rushing attacks
- ▶ The position, which is close to the receivers is the most efficiently rushing position if the number of attackers is high *enough*, otherwise staying at the mesh center results in more rushing success.



Final Remarks 2

- ▶ In terms of PDR, E2E delay and delay jitter, a multicast with high number of senders and/or high number of receivers is most resilient to blackhole, neighbor and jellyfish attacks
- ▶ In terms of attack positions, attackers who are near the senders cause the most damage if the number of attackers is high *enough*, otherwise the mesh center is the strongest attack position.
- ▶ In spite of their different operations, blackhole and neighbor attacks cause the same degree of damage.



Publications

-  [Journal] Hoang Lan Nguyen, Uyen Trang Nguyen.
A Study of Different Types of Attacks on Multicast in Mobile Ad Hoc Networks,
To appear in the Elsevier Journal of Ad Hoc Networks, 2006
-  [Conference] Hoang Lan Nguyen, Uyen Trang Nguyen.
A Study of Different Types of Attacks on Multicast in Mobile Ad Hoc Networks,
5th IEEE International Conference on Networking (ICN'06), Mauritius, Apr. 2006.

