Wireless Network Security
Spring 2016

Patrick Tague

Class #11 - Identity Mgmt.;
Routing Security

Class #11

- Identity threats and countermeasures
- Basics of routing in ad hoc networks
- Control-plane attacks and defenses

Addressing

- In traditional networking, each device (radio) has two identities, in the form of addresses
  - MAC address: hardware address of the radio needed for link layer communication (e.g., 802.3, 802.11)
    - Hard-coded into the NIC
    - In theory, unique and static
  - IP address: network layer address used for routing and some other higher layer services
    - Virtual software address

MAC Addresses

- MAC addresses in the Internet
  - Ethernet and WiFi use MAC addresses for link layer communication
  - Independent of any higher-layer functionality
  - Link layer frames carry source and destination MAC addresses (6B each)
- MAC addresses in other systems
  - Not typically used in sensor networks due to overhead
  - Not needed if other addressing is available

IP Addresses

- IP addresses in the Internet
  - Network layer and above use IP addresses for some identity purposes
  - Independent of whatever is below the network layer
  - IP addresses must be unique

- IP addresses in other systems
  - To support common applications, most designers are aiming to support IP addressing (to some extent)

IP Address Resolution

- In most Internet domains, IP addresses are assigned centrally using DHCP and bound to MAC addresses using ARP
  - DHCP = Dynamic Host Configuration Protocol: host asks server for IP address, which it keeps until expiry
  - ARP = Address Resolution Protocol: host asks other hosts for MAC address corresponding to an IP address

ARP

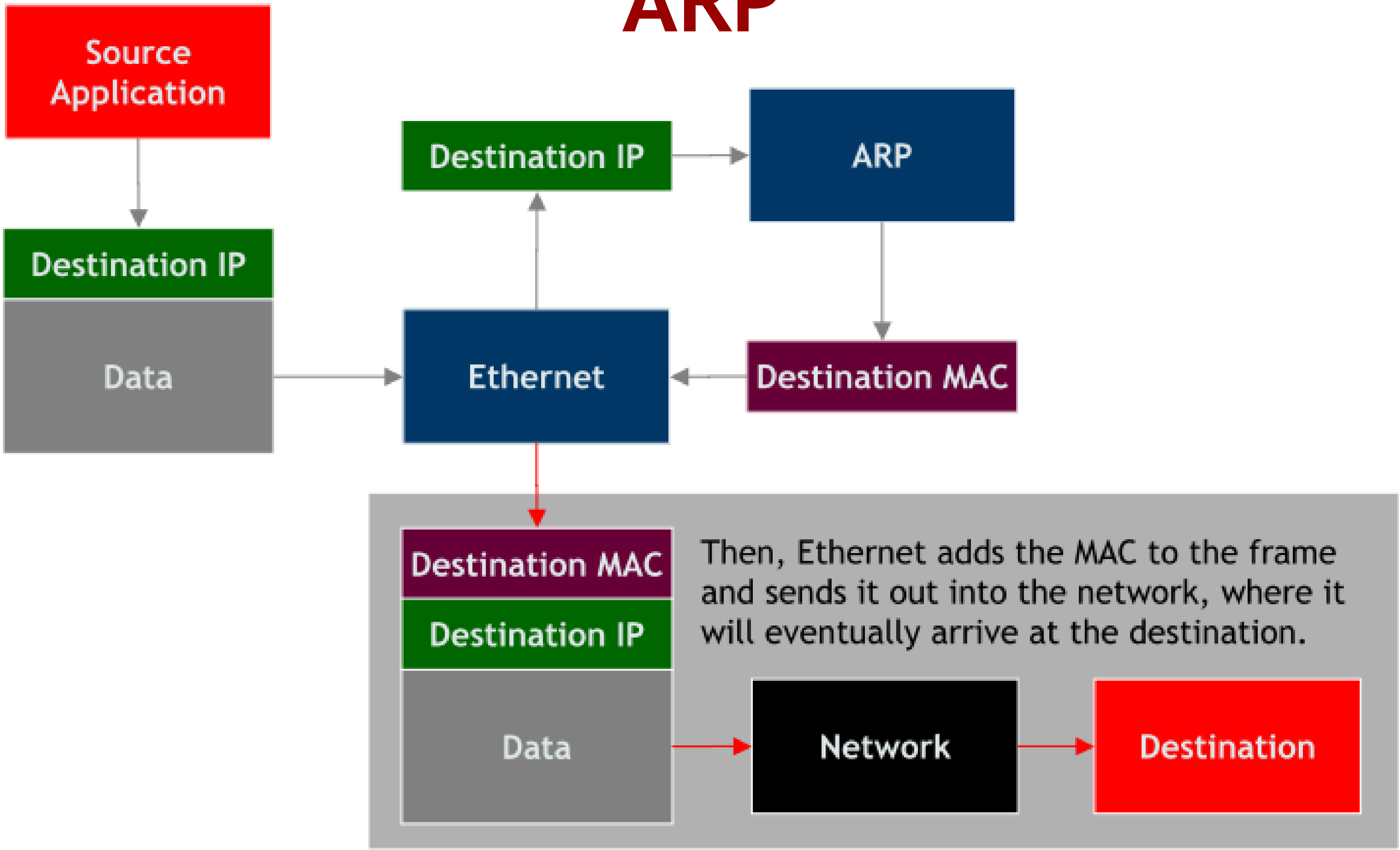


image from [Whalen et al., 2001]

Limitations

- MAC addresses are no longer hardware-bound
  - Most Linux-like systems allow software to change MAC address used, despite hard-coded MAC address
  - Many devices don't have (unique) MAC addresses
- DHCP is impractical for distributed systems
  - Requires centralization
  - High overhead in dynamic systems
- ARP has high overhead in distributed systems
  - Requires request flooding

Distributed Addressing

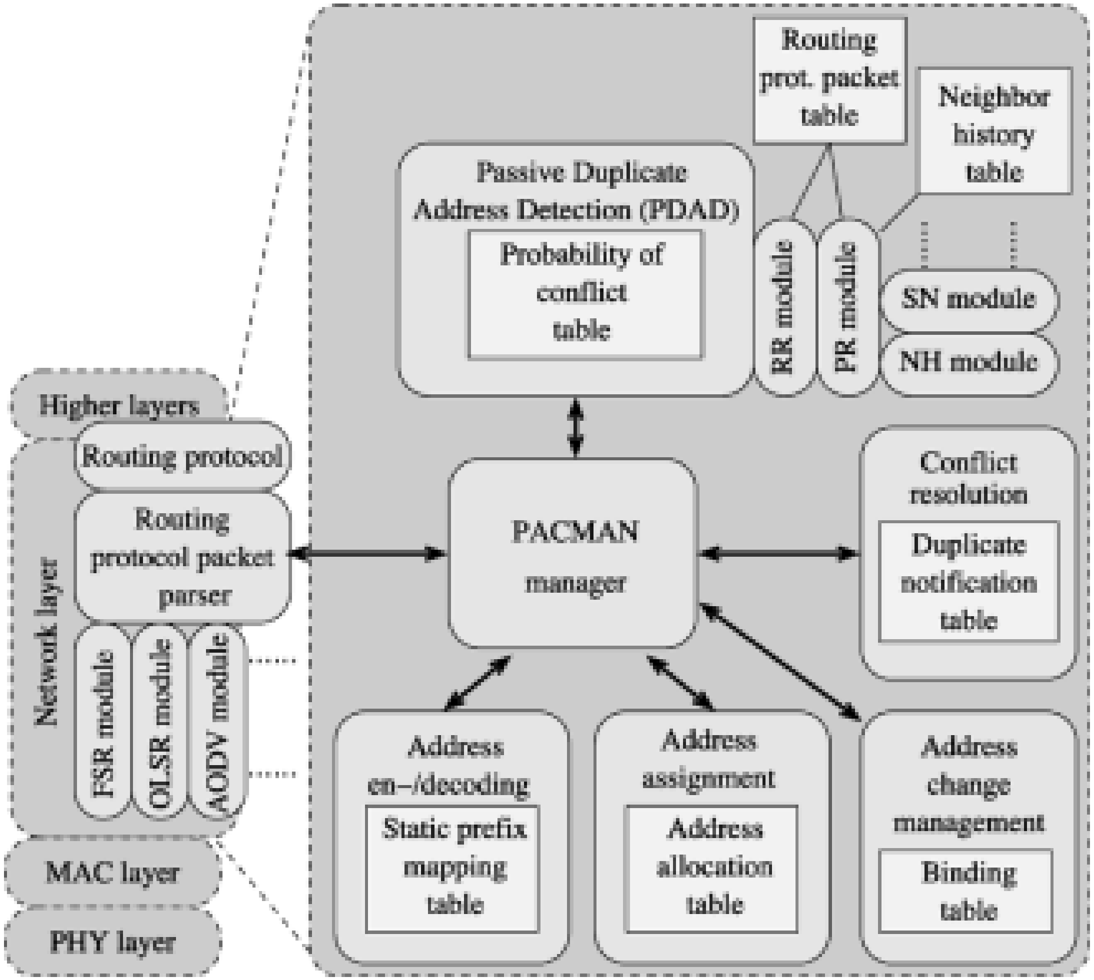
- **Problem**: How should IP addresses (or other suitable identities) be determined in a distributed system such that:
  - Addresses are compact(-able) for low-overhead communication in sensors or embedded devices
  - Network overhead is (relatively) low
  - Addresses are (sufficiently) unique
  - Systems can split and join
    - Duplicate addresses can be detected and fixed
    - Address space is large enough and dynamic

A Few Approaches

- Random selection with duplicate address detection (DAD)
  - Send a query to the selected address; if no response, the address probably isn't in conflict
  - Requires flooding a query through the entire network
  - Merging existing networks is difficult
- MANETconf
  - Configured “initiator” nodes act like a server that can assign addresses to “requesters” who arrive later
  - Configured node floods notification and assigns address if no nodes respond negatively
  - Merging existing networks is difficult

PACMAN
[Weniger, JSAC 2005]

- PACMAN = Passive Auto-Configuration for Mobile Ad hoc Networks
  - Architecture for efficient distributed MANET address auto-configuration

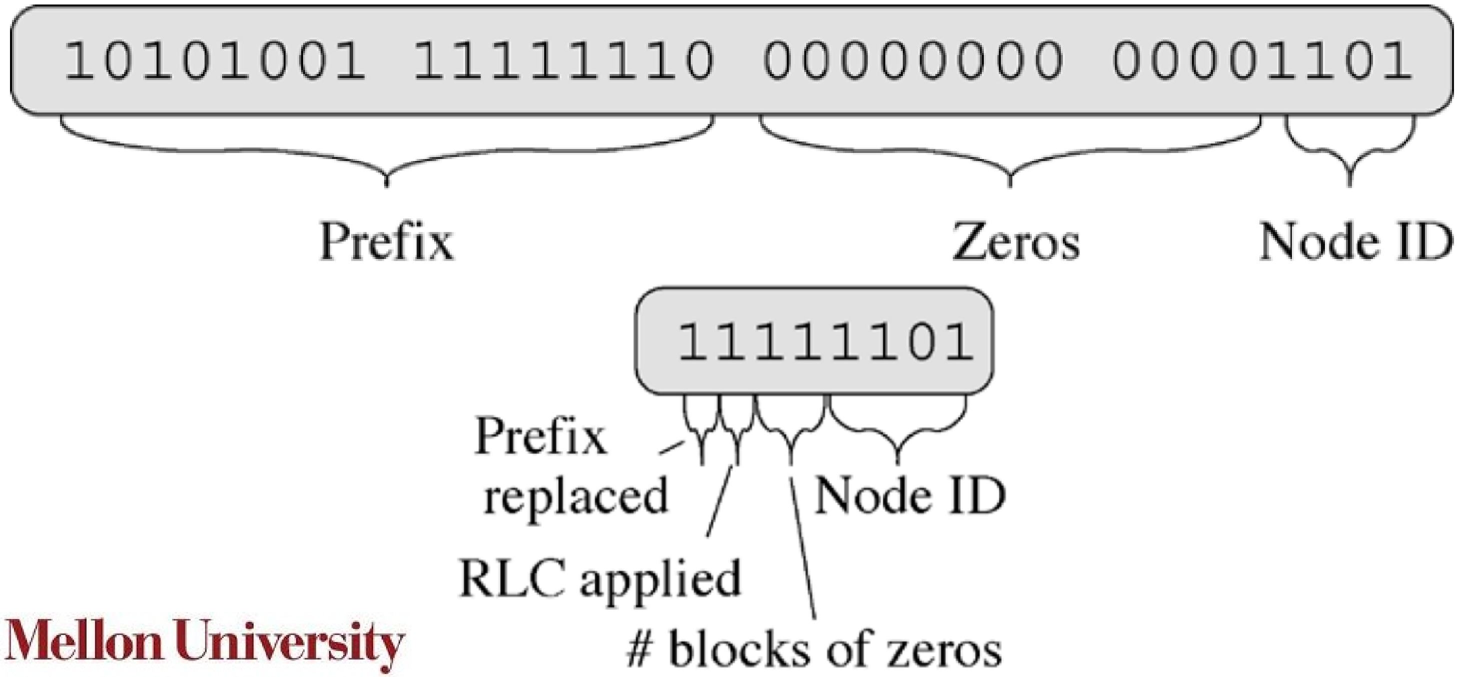


Address Assignment

- To avoid overhead of flooding network to check address uniqueness, PACMAN assigns addresses passively and relies on network to expose conflicts
  - Each node chooses its address using a probabilistic algorithm and a list of known used addresses
  - See the paper for details

Address Encoding

- To minimize overhead, PACMAN encodes MANET IP addresses
  - MANET uses a fixed IP prefix (2B for IPv4, 8B for IPv6)
  - Node ID only needs $\log_2 N$ bits to support N nodes
  - Pad with lots of zeros, but only need to know #0s



Passive DAD

- PDAD relies on observation of events that:
  - Never occur in case of unique addresses but always occur in case of address duplication
    - e.g., receiving a route reply when no request was sent
  - Usually don't occur in case of unique addresses but sometimes occur in case of address duplication
    - e.g., link states in a route reply change completely
- Upon detection of duplication, at least one node can reinitialize the address assignment
  - This also allows relatively easy management of network split and merge events

Security Issues

- PACMAN and many similar approaches were not designed with malicious behaviors in mind
- Threats [Wang et al., 2005]:
  - Address spoofing – attacker spoofs the IP address of a victim and hijacks its traffic
  - False address conflict – attacker injects conflict messages (or events) to a target victim
  - Address exhaustion – attacker claims many addresses to deny service or prevent nodes from joining
  - Negative reply – in cases where approval is needed to join, attacker can prevent nodes from joining

Secure MANET Auto-Conf

[Wang et al., 2005]

- Bind the IP address to a public key to authenticate auto-configuration processes
  - New node A chooses an IP address as the hash of its public key
  - A sends a query to the network for the IP address using a signed, time-stamped Duplicate Address Probe
    - If a receiving node B has an IP conflict, it checks signatures (authenticity, replay prevention, etc.) and conditionally replies with a signed, time-stamped Address Conflict Notice
    - If A receives ACN from B, it checks signatures and conditionally starts over with a new key pair
    - If no reply within a fixed time period, A joins the network using the generated IP address

Benefits of the Approach

- Forces the attacker to find a public key that hashes to a victim's IP address before launching the attack
  - Even with relatively small address space, computation/storage overhead is prohibitive
  - Detailed analysis in the paper

Trust-Based Auto-Conf

[Hu & Mitchell, 2005]

- Misbehavior in the “requester-initiator” model (MANETconf)
  - Initiator can intentionally assign conflicting address
  - Requester can flood requests repeatedly, causing resource depletion and/or DoS
  - Malicious node can falsely claim that candidate addresses are already in use, causing excess request floods, resource depletion, and DoS
- Instead of relying on arbitrary nodes, keep track of which nodes are “good” and which are “bad”
  - A's trust in B is given by $T_A(B)$, computed based on past behaviors/interactions

Choosing a Trustable Initiator

- New requester N broadcasts a Neighbor_Query with its threshold T_N^*
- Each receiver sends N a InitREP reply with neighbor IDs who have trust values $\geq T_N^*$
- N can chooses its initiator as the neighbor appearing in the most InitREP messages
- Malicious node is unlikely to be chosen unless majority of neighbors are malicious

Duplicate Address Check

- If initiator A gets an Add_Collision message from a node B in response to an Initiator_Request:
  - If B has been previously blacklisted, ignore it
  - If $T_A(B) \geq T_A^*$, then believe B and start over
  - Otherwise, declare B malicious, add B to the blacklist, and send a Malicious_Suspect message about B to other nodes
    - Other nodes only believe A's Malicious_Suspect message if their trust value in A is high enough

3rd-Party Duplication Check

- If a node B detects an address collision between two other nodes, it notifies both of them
- If a receiving node A gets such a notification from node B:
  - If B has been previously blacklisted, ignore it
  - If $T_A(B) \geq T_A^*$, believe B and choose a new address
  - Otherwise, add B to the blacklist

Summary

- Discussed distributed addressing, threats, and a few approaches to secure auto-configuration
  - PACMAN: Passive auto-configuration for MANETs
    - [Weniger; JSAC 2005]
  - Secure address auto-configuration for MANETs
    - [Wang, Reeves, and Ning; MobiQuitous 2005]
  - Secure auto-configuration in MANETs using trust
    - [Hu and Mitchell; MSN 2005]

On to routing security – let's start with some basics of MANET routing

Popular Routing Protocols

- Link State (LS) routing
  - Optimized Link State Routing (OLSR)

- Distance Vector (DV) routing
  - Destination Sequenced Distance Vector (DSDV)
  - Ad hoc On-demand Distance Vector (AODV)
  - Dynamic Source Routing (DSR)

Proactive

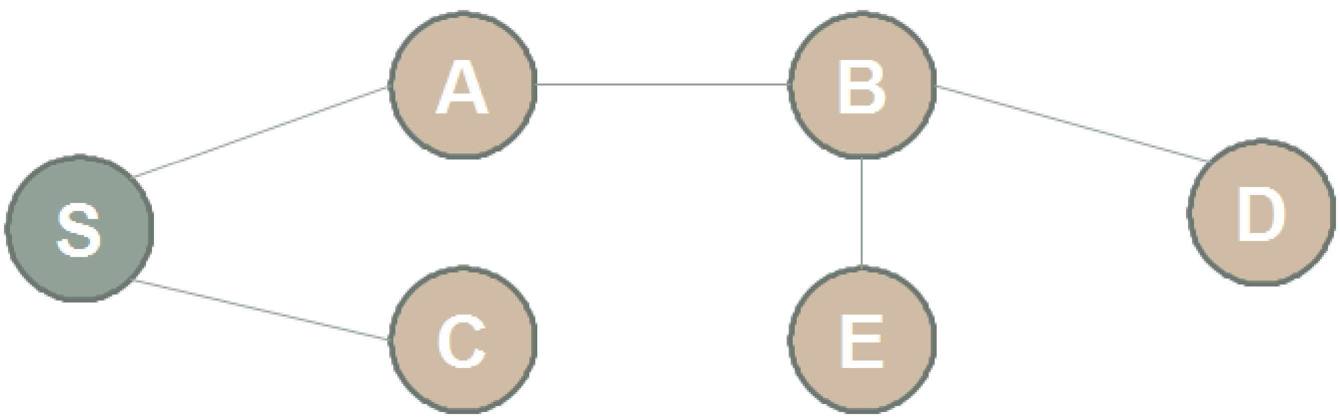
On Demand

On-Demand Routing

- On-demand routing has several advantages and disadvantages in MANETs
  - Efficiency:
    - (+) Routing information isn't constantly collected and updated, only when needed
    - (-) One-time cost of info collection can be higher
  - Security:
    - (+) Source nodes are aware of the entire path, unlike fully distributed algorithms that just focus on next hop
    - (-) Long-term information typically isn't available
  - Overall, advantages outweigh the disadvantages, so on-demand routing (esp. source routing) is popular

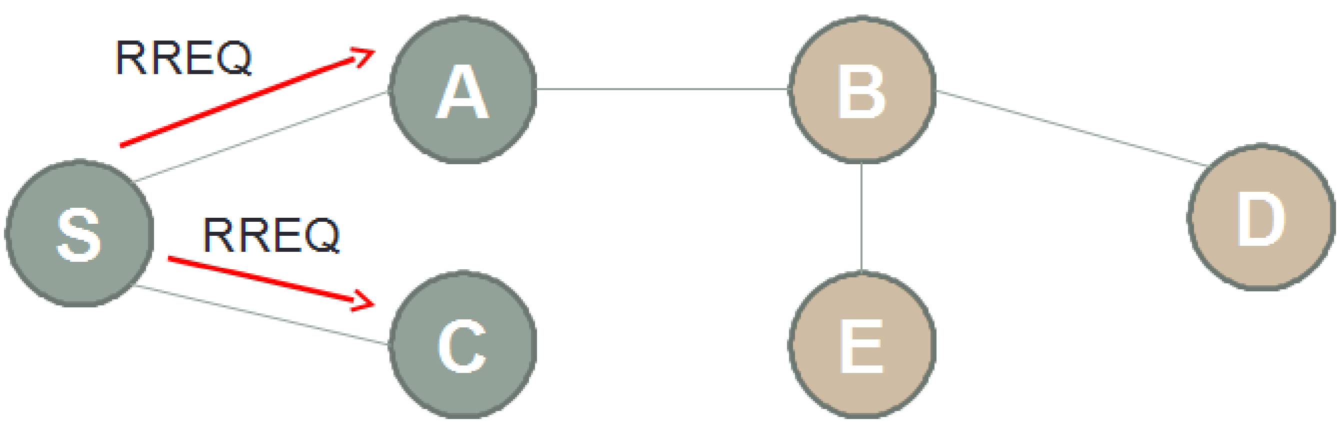
Route Discovery

- Source *S* and neighboring nodes use control message exchanges to discover a route from *S* to destination *D*



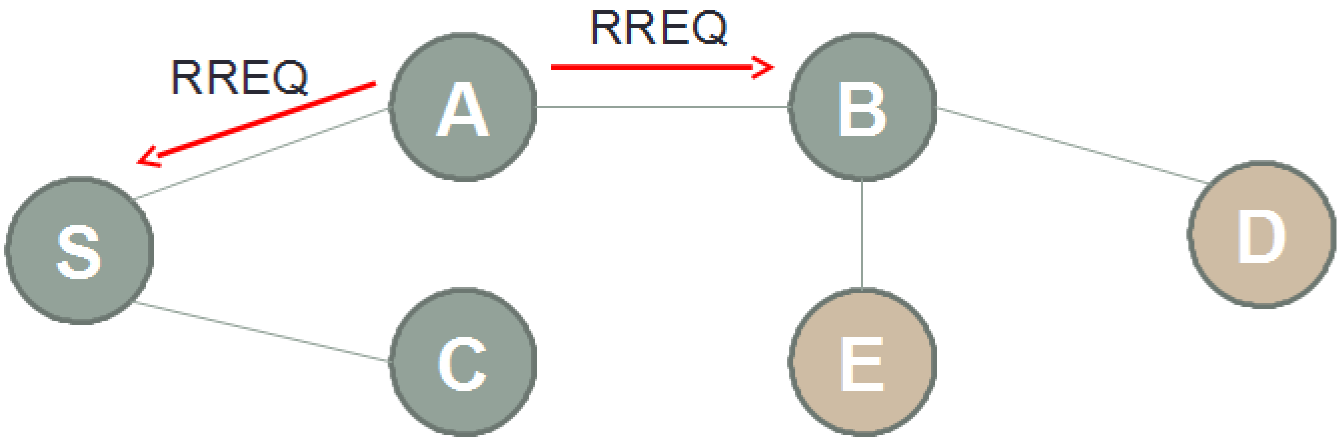
Route Discovery

- Route request flooding:
  - Source *S* broadcasts a Route Request (RREQ) packet to its neighbors



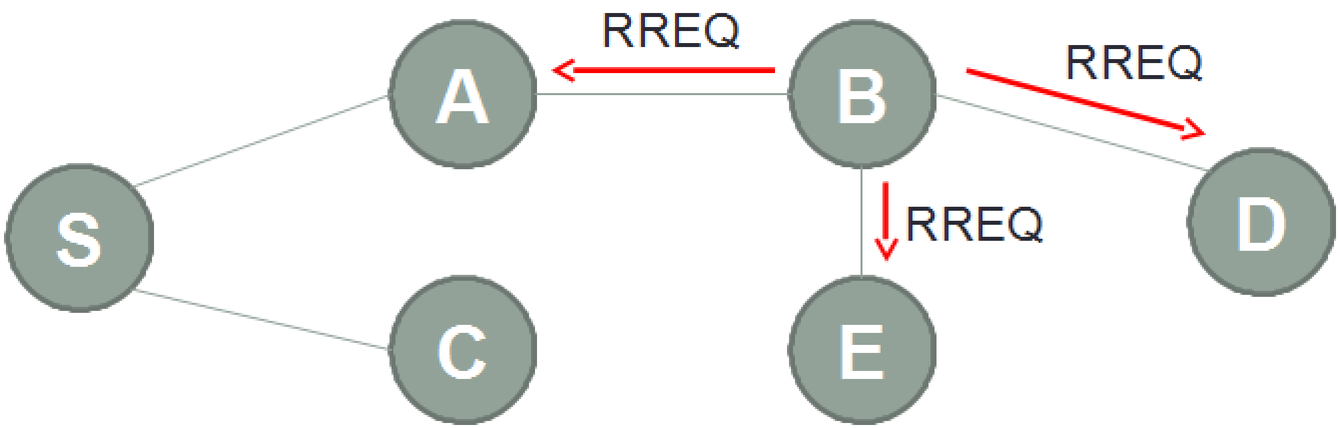
Route Discovery

- RREQ forwarding:
  - If the neighbor has no prior relationship with the destination, it will further broadcast the RREQ



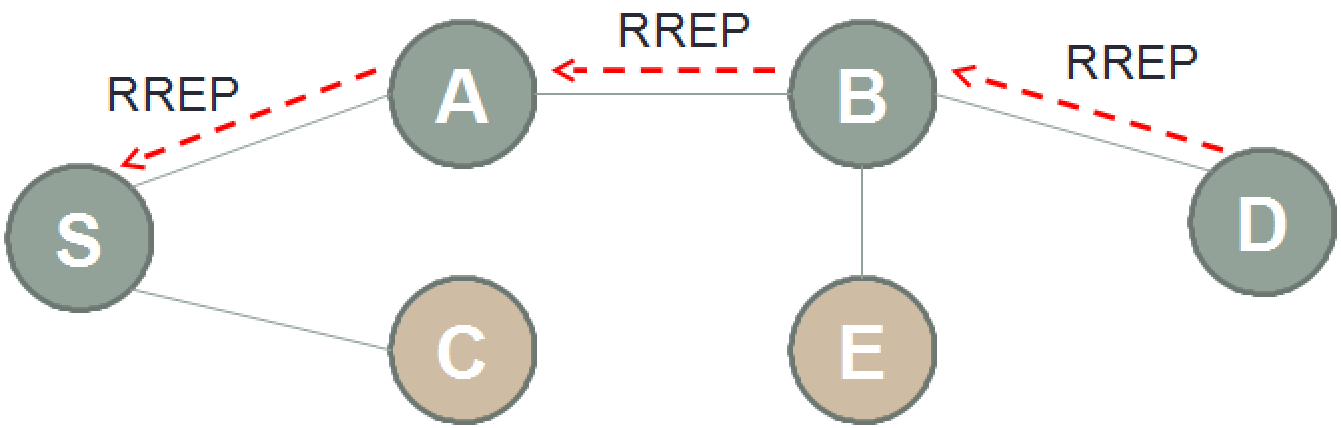
Route Discovery

- Flooding of control packets to discover routes
  - Once the RREQ packet reaches the destination, or a node that knows the destination, the node will unicast a RREP packet to the source via the routed path



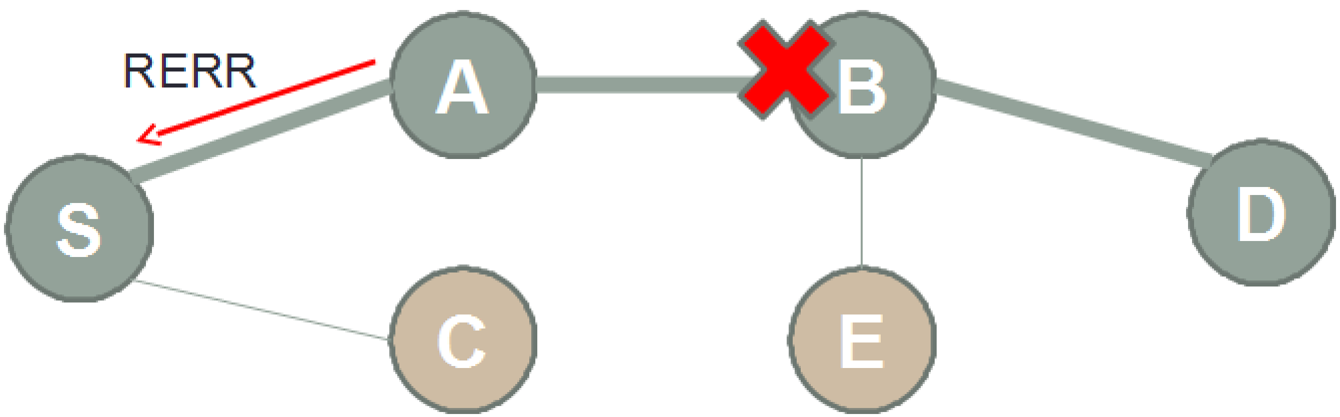
Route Discovery

- Upon receiving the RREQ, *D* (or another node that knows *D*) will unicast a Route Reply (RREP) back to *S* along the found path



Route Maintenance

- If a node can no longer reach the next hop
  - Sends Route Error (RERR) control packet to inform upstream neighbors
  - Route cache alternative (DSR) or rediscovery



AODV vs. DSR

AODV	DSR
Routing tables • one route per destination	Routing caches • multiple routes per destination
Always chooses fresher routes • Sequence numbers	Does not have explicit mechanism to expire stale routes
More frequent discovery flood to ensure freshness	Source Routing • Intermediate nodes learn routes in 1 discovery cycle

Now, how could an attacker interfere with or manipulate MANET routing?

Modification Attacks

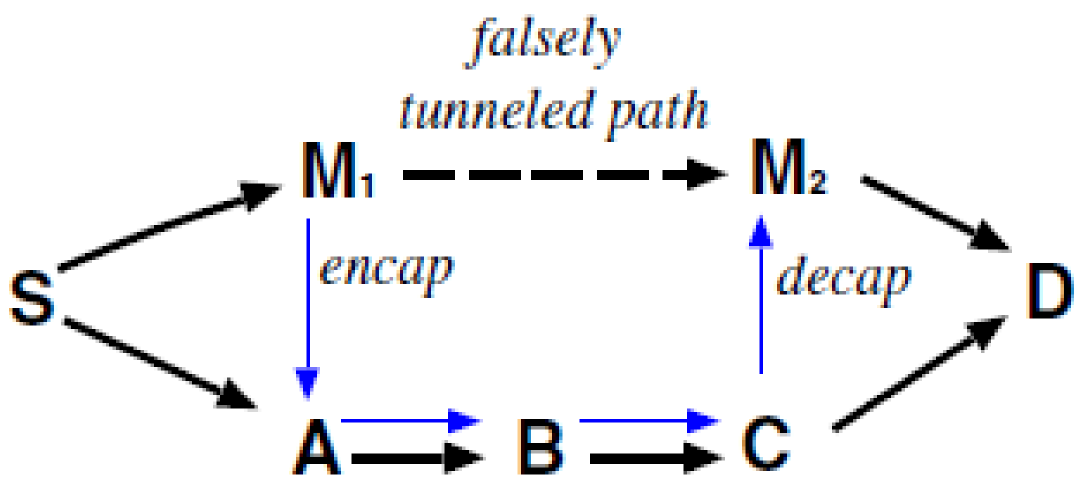
- AODV seq# modification
  - AODV uses seq# as a timestamp (high seq# → fresh)
  - Attacker can raise seq# to make its path attractive

- DSR hop count modification
  - DSR uses #hops for efficiency (low #hops → cheap)
  - Attacker can lower/raise #hops to attract/repel

Modification Attacks

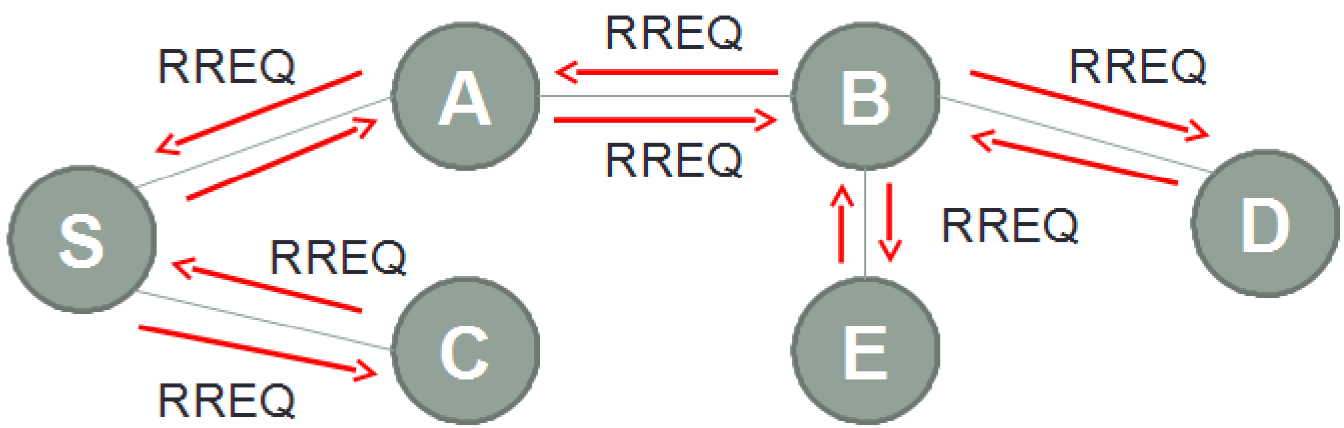
- DSR route modification
  - Non-existent route (DoS)
  - Loops (resource exhaustion, DoS)
  - No control to prevent loops after route discovery (more of a data plane attack, we'll get there later)

- Tunneling



RREQ Flooding

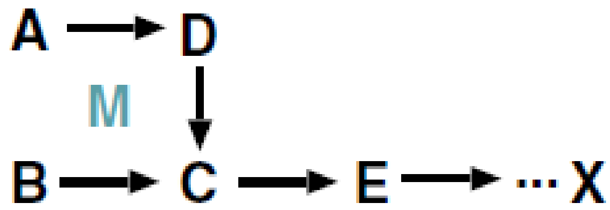
- Flood the network with RREQs to an unreachable destination address



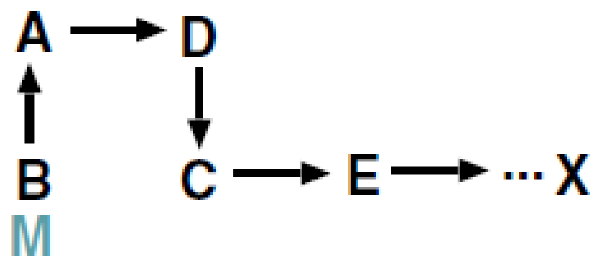
Example : S continuously send RREQ packet to destination X

AODV/DSR Spoofing

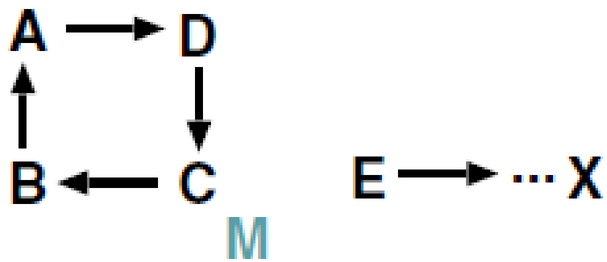
- Attacker listens for RREQ/RREP from neighbors



- Send an "attractive" RREP with spoofed ID

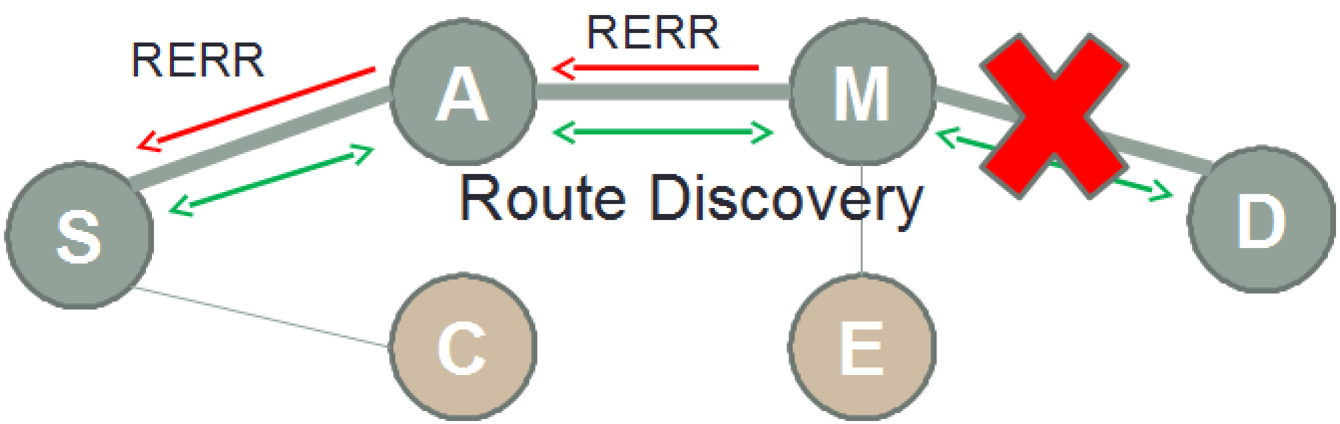


- Spoof more IDs with interesting results



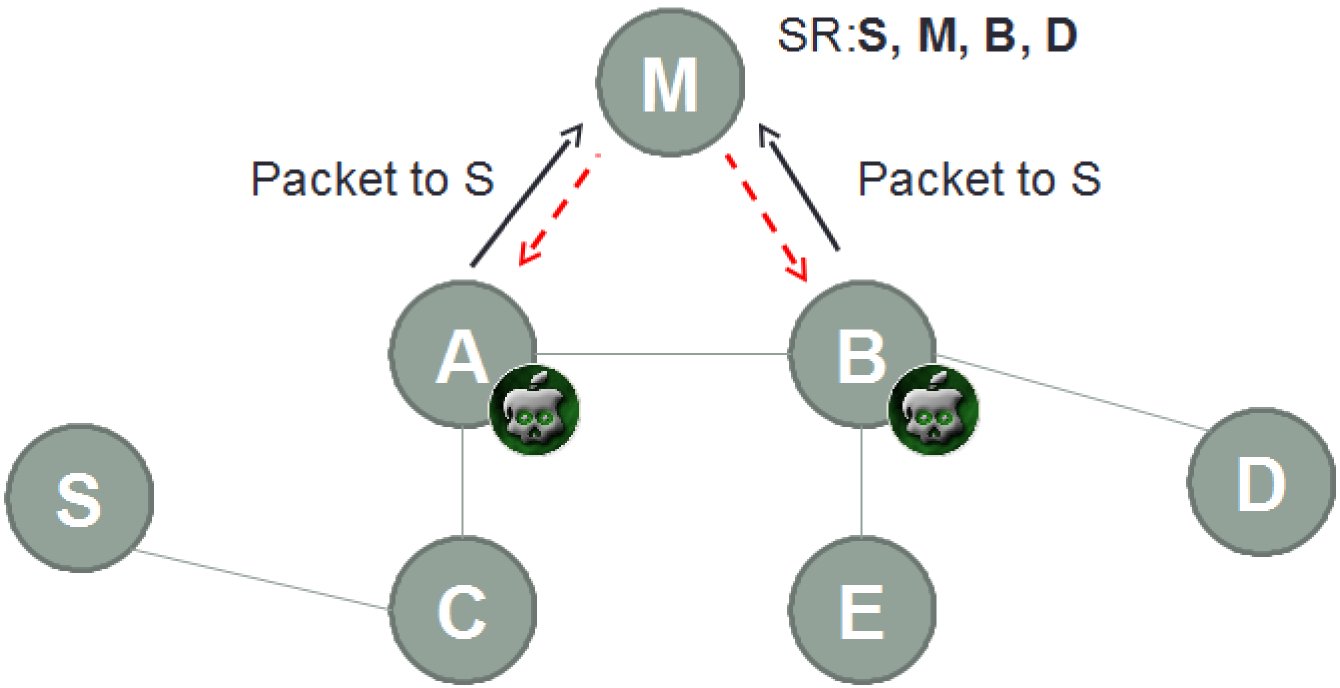
Fabrication Attacks

- DoS against AODV/DSR by falsifying route errors



Fabrication Attacks

- DSR route cache poisoning



Control-Plane Security

- How to guarantee that an established path can be efficient (e.g., short) and/or reliable?
- How to prevent attackers from manipulating path discovery/construction?
- What metrics can be used to quantify the value of a path?
  - Length? Latency? Trust?

Feb 23:

Forwarding Security

Feb 25:

SoW Presentations;
Network Privacy & Anonymity