Mobile Wireless Networking
The University of Kansas EECS 882
MANET Routing

James P.G. Sterbenz

Department of Electrical Engineering & Computer Science
Information Technology & Telecommunications Research Center
The University of Kansas

jgps@eecs.ku.edu

http://www.ittc.ku.edu/~jgps/courses/mwnets

29 October 2009

rev. 1.1

© 2004–2009 James P.G. Sterbenz
Mobile Wireless Networking
MANET Routing Algorithms and Protocols

MR.1  Routing algorithm alternatives
MR.2  Example protocols
MANET Algorithms and Protocols

Introduction

- Mobile ad hoc network (MANET) Lecture AH
  - mobile: node and groups of nodes move
    - subject to a mobility model Lecture LM
  - wireless: mobility implies mostly wireless links
    - weak, asymmetric, episodic connectivity Lecture PL
  - ad hoc: little or no reliance on network infrastructure
    - from Latin: for this (purpose)
MANET Routing

MR.1 Routing algorithm alternatives

MR.2 Example protocols
MANET Routing

Challenges

- Routing algorithm

*purpose and distinction from forwarding?*
MANET Routing

Challenges

- Routing algorithm discovers path *EECS780 lecture NR*
  - between source(s) and destination(s)
MANET Routing

Challenges

• Routing algorithm discovers path
  – between source(s) and destination(s)

• Challenges in MANETs
  – episodic connectivity and mobility

*implications?
MANET Routing
Challenges

• Routing algorithm discovers path
  – between source(s) and destination(s)

• Challenges in MANETs
  – episodic connectivity and mobility
  – topology and link state keeps changing
  – difficult or impossible to maintain consistent information
MANET Routing

Challenges

• Routing algorithm discovers path
  – between source(s) and destination(s)

• Challenges in MANETs
  – episodic connectivity and mobility
  – topology and link state keeps changing
  – difficult or impossible to maintain consistent information

• Conventional routing algorithms do not work well

  *why?*
MANET Routing

Challenges

- Routing algorithm discovers path
  - between source(s) and destination(s)
- Challenges in MANETs
  - episodic connectivity and mobility
  - topology and link state keeps changing
  - difficult or impossible to maintain consistent information
- Conventional routing algorithms do not work well
  - assume relatively stable topologies and link state
  - convergence difficult or impossible
MANET Routing
Algorithm vs. Protocol

- Routing algorithm
  - algorithmic formalism to describe path discovery
    *example: link state / Dijkstra shortest path algorithm*
MANET Routing
Algorithm vs. Protocol

- Routing algorithm
  - algorithmic formalism to describe path discovery

- Routing protocol
  - specification of algorithm as state machines and PDUs
  - permits interoperable operation among nodes
  - example: OSPF protocol using link state & Dijkstra
  - common to be sloppy about algorithm/protocol difference
MANET Routing
Algorithm vs. Protocol

• Routing algorithm
  – algorithmic formalism to describe path discovery

• Routing protocol
  – specification of algorithm as state machines and PDUs
  – permits interoperable operation among nodes

• Routing Implementation
  – software or hardware implementation of protocol
  – code and data structures compliant to protocol specification

  example: OSPF protocol implementation in Cisco IOS
MANET Routing Algorithms

Design Alternatives

- Routing algorithm type
- Topological structure
- Routing update mechanism
- Resource and context awareness
MANET Routing Algorithms

Design Alternatives: Algorithm Type

- Routing algorithm type *EECS780 lecture NR*
  - distance vector
  - link state
  - source route
- Topological structure
- Routing update mechanism
- Resource and context awareness
MANET Routing Algorithms
Algorithm Type: Distance Vector

- Distance vector
  *operation and relative advantages?*

- Link state
- Source route
MANET Routing Algorithms
Algorithm Type: Distance Vector

• Distance vector
  – nodes maintain vectors of shortest next hops per destination
  – Bellman-Ford path computation
  – minimises number of hops
  – slow to converge for large networks
  – subject to routing loops and count-to-infinity
  – only appropriate for small networks
  
  (wired) examples?

• Link state
• Source route
MANET Routing Algorithms
Algorithm Type: Distance Vector

- **Distance vector**
  - nodes maintain vectors of shortest next hops per destination
  - Bellman-Ford path computation
  - minimises number of hops
  - slow to converge for large networks
  - subject to routing loops and count-to-infinity
  - only appropriate for small networks
  - (wired) examples: RIP, IGRP, EIGRP

- **Link state**
- **Source route**
MANET Routing Algorithms
Algorithm Type: Link State

- Distance vector
- Link state
  
  *operation and relative advantages*

- Source route
MANET Routing Algorithms

Algorithm Type: Link State

- Distance vector
- Link state
  - every node maintains complete topology database
  - Dijkstra shortest path algorithm
  - flooding of link state advertisements (LSAs)
  - rapid convergence
  - better for large networks than distance vector
    *(wired) examples?*
- Source route
MANET Routing Algorithms

Algorithm Type: Link State

- **Distance vector**
- **Link state**
  - every node maintains complete topology database
  - Dijkstra shortest path algorithm
  - flooding of link state advertisements (LSAs)
  - rapid convergence
  - better for large networks than distance vector
  - (wired) examples: OSPF, ISIS
- **Source route**
MANET Routing Algorithms
Algorithm Type: Source Routed

- Distance vector
- Link state
- Source route

operation and relative advantages?
MANET Routing Algorithms

Algorithm Type: Source Routed

- Distance vector
- Link state
- Source route
  - source constructs the sequence of nodes to destination
MANET Routing Algorithms
Algorithm Type: Source Routed

- Distance vector
- Link state
- Source route
  - source constructs the sequence of nodes to destination
  - *route discovery* algorithm needed
  - state carried in packet headers
    - rather than maintained in forwarding tables

*(wired) examples?*
MANET Routing Algorithms
Algorithm Type: Source Routed

- Distance vector
- Link state
- Source route
  - source constructs the sequence of nodes to destination
  - *route discovery* algorithm needed
  - state carried in packet headers
    - rather than maintained in forwarding tables
  - (wired) examples: IP source route option (rarely used)
    future Internet proposals: NewArch, PoMo
MANET Routing Algorithms

Design Alternatives: Topological Structure

- Routing algorithm type
- Topological structure
  - flat
  - hierarchical
- Routing update mechanism
- Resource and context awareness
MANET Routing Algorithms

Topological Structure: Flat

- Flat
  - all nodes have a uniform view of identifier space
  - limitations?
- Hierarchical
MANET Routing Algorithms

Topological Structure: Flat

- Flat
  - all nodes have a uniform view of identifier space
  - appropriate for small networks
  - (wired) examples?

- Hierarchical
MANET Routing Algorithms

Topological Structure: Flat

- Flat
  - all nodes have a uniform view of identifier space
  - appropriate for small networks
  - (wired) examples: RIP, IGRP

- Hierarchical
MANET Routing Algorithms

Topological Structure: Hierarchical

- Flat
- Hierarchical

motivation?
MANET Routing Algorithms
Topological Structure: Hierarchical

- Flat
- Hierarchical
  - allows *scalable* construction of large networks
    - hierarchy is almost always the first answer to network scaling

*implementation?*
MANET Routing Algorithms
Topological Structure: Hierarchical

- **Flat**
- **Hierarchical**
  - allows *scalable* construction of large networks
    - hierarchy is almost always the first answer to network scaling
  - nodes divided into clusters
  - each cluster has uniform access to other members
    - e.g. LSA flooding and topology databases
MANET Routing Algorithms
Topological Structure: Hierarchical

- Flat
- Hierarchical
  - allows *scalable* construction of large networks
    - hierarchy is almost always the first answer to network scaling
  - nodes divided into clusters
  - each cluster has uniform access to other members
    - e.g. for link state advertisement flooding and topo databases
  - clusters abstracted into virtual nodes at next level
MANET Routing Algorithms

Topological Structure: Hierarchical

- Flat
- Hierarchical
  - allows *scalable* construction of large networks
    - hierarchy is almost always the first answer to network scaling
  - nodes divided into clusters
  - each cluster has uniform access to other members
    - e.g. for link state advertisement flooding and topo databases
  - clusters abstracted into virtual nodes at next level

*(wired) examples?*
MANET Routing Algorithms

Topological Structure: Hierarchical

- Flat
- Hierarchical
  - allows *scalable* construction of large networks
    - hierarchy is almost always the first answer to network scaling
  - nodes divided into clusters
  - each cluster has uniform access to other members
    - e.g. for link state advertisement flooding and topo databases
  - clusters abstracted into virtual nodes at next level
  - (wired) examples: OSPF (2-level), P-NNI (n-level)
MANET Routing Algorithms
Design Alternatives: Update Mechanism

- Routing algorithm type
- Topological structure
- Routing update mechanism
  - proactive or table-driven
  - reactive or on-demand
  - predictive
- Resource and context awareness
MANET Routing Algorithms

Update Mechanism: Proactive / Table-Driven

- Proactive or table-driven
  - compute routes that *may* be needed (*proactive*)
  - insert into forwarding tables (*table-driven*)

*sound familiar?*
MANET Routing Algorithms

Update Mechanism: Proactive / Table-Driven

- Proactive or table-driven
  - compute routes that *may* be needed (proactive)
  - insert into forwarding tables (table-driven)
  - conventional routing technique used in Internet

*advantages and disadvantages?*
MANET Routing Algorithms
Update Mechanism: Proactive / Table-Driven

• Proactive or table-driven
  – compute routes that *may* be needed (*proactive*)
  – insert into forwarding tables (*table-driven*)
  – conventional routing technique used in Internet
  + low communication startup latency
  – overhead of continuous route maintenance
    • routing must *reconverge* in response to topology changes
MANET Routing Algorithms

Update Mechanism: Reactive / On-Demand

- Proactive or table-driven
- Reactive or on-demand

difference from proactive?
MANET Routing Algorithms
Update Mechanism: Reactive / On-Demand

- Proactive or table-driven
- Reactive or on-demand
  - compute paths *only when* needed (*reactive on-demand*)
  - may insert into forwarding tables or drive source routing
MANET Routing Algorithms
Update Mechanism: Reactive / On-Demand

- Proactive or table-driven
- Reactive or on-demand
  - compute paths only when needed (reactive on-demand)
  - may insert into forwarding tables or drive source routing
  - requires route discovery signalling

advantages and disadvantages?
MANET Routing Algorithms

Update Mechanism: Reactive / On-Demand

- Proactive or table-driven
- Reactive or on-demand
  - compute paths *only when* needed (*reactive on-demand*)
  - may insert into forwarding tables or drive source routing
  - requires *route discovery* signalling
- lower overall overhead
  - assumption?
MANET Routing Algorithms
Update Mechanism: Reactive / On-Demand

- Proactive or table-driven
- Reactive or on-demand
  - compute paths only when needed (reactive on-demand)
  - may insert into forwarding tables or drive source routing
  - requires route discovery signalling
+ lower overall overhead
  - assuming new route request rate not very high
  - higher startup delay
    why?
MANET Routing Algorithms
Update Mechanism: Reactive / On-Demand

- Proactive or table-driven
- Reactive or on-demand
  - compute paths only when needed (reactive on-demand)
  - may insert into forwarding tables or drive source routing
  - requires route discovery signalling
+ lower overall overhead
  - assuming new route request rate not very high
- higher startup delay
  - each communication flow must wait for route discovery mitigation?
MANET Routing Algorithms

Update Mechanism: Reactive / On-Demand

- Proactive or table-driven
- Reactive or on-demand
  - compute paths *only when* needed (*reactive on-demand*)
  - may insert into forwarding tables or drive source routing
  - requires *route discovery* signalling
  + lower overall overhead
    - assuming new route request rate not very high
  - higher startup delay
    - each communication flow must wait for route discovery
    - unless route *cached* from recent flow between *same* node pair
  - overhead of route maintenance
    *why?*
MANET Routing Algorithms

Update Mechanism: Reactive / On-Demand

- Proactive or table-driven
- Reactive or on-demand
  - compute paths *only when* needed (*reactive on-demand*)
  - may insert into forwarding tables or drive source routing
  - requires *route discovery* signalling
  + lower overall overhead
    - assuming new route request rate not very high
  - higher startup delay
    - each communication flow must wait for route discovery
    - unless route *cached* from recent flow between *same* node pair
  - overhead of route maintenance
    - signalling to alter routes with topology changes if rate high
MANET Routing Algorithms

Update Mechanism: Predictive

- Proactive or table-driven
- Reactive or on-demand
- Predictive

*what and why?*
MANET Routing Algorithms

Update Mechanism: Predictive

- Proactive or table-driven
- Reactive or on-demand
- Predictive
  - predict where packets go in advance
    
    why?
MANET Routing Algorithms

Update Mechanism: Predictive

- Proactive or table-driven
- Reactive or on-demand
- Predictive
  - predict where packets go in advance
  - necessary when mobility exceeds ability to converge
  - *how?*
MANET Routing Algorithms

Update Mechanism: Predictive

- Proactive or table-driven
- Reactive or on-demand
- Predictive
  - predict where packets go in advance
  - necessary when mobility exceeds ability to converge
  - location management with trajectory prediction
    - exploit knowledge about mobility model

*lecture LM*
• Routing and forwarding expect mobility
• Use location/trajectory information where available
  – unicast when predictable (e.g. planetary or racetrack UAV)
• Routing and forwarding expect mobility
• Use location/trajectory information where available
  – unicast when predictable (e.g. planetary or racetrack UAV)
MANET Routing Algorithms
Update Mechanism: Predictive

- Routing and forwarding expect mobility
- Use location/trajectory information where available
  - unicast when predictable (e.g. planetary or racetrack UAV)
  - multicast to area of expected location (spray routing)
• Routing and forwarding expect mobility
• Use location/trajectory information where available
  – unicast when predictable (e.g. planetary or racetrack UAV)
  – multicast to area of expected location (spray routing)
Routing and forwarding expect mobility

Use location/trajectory information where available
  - unicast when predictable (e.g. planetary or racetrack UAV)
  - multicast to area of expected location (spray routing)

cluster may have inherent broadcast or epidemic routing
MANET Routing Algorithms

Update Mechanism: Predictive Scenario

- Very high relative velocity
  - Mach 7 ≈ 10 s contact
  - dynamic topology
- Communication channel
  - limited spectrum
  - asymmetric links
    - data down omni
    - C&C up directional
- Multihop
  - among ANs
  - through relay nodes

AN – airborne node
RN – relay node
GS – ground station
GW – gateway
Internet

ANs
RN
GS
GW
## MANET Routing Algorithms

### Update Mechanism: Predictive Scenario

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Transmit Range [nmi]</th>
<th>Relative Velocity [knots]</th>
<th>Contact Duration [sec]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-Hop Best Case</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GS – AN</td>
<td>140</td>
<td>400</td>
<td>2520</td>
</tr>
<tr>
<td>AN – AN</td>
<td>15</td>
<td>800</td>
<td>135</td>
</tr>
<tr>
<td>Single-Hop Worst Case</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GS – AN</td>
<td>100</td>
<td>Mach 3.5</td>
<td>300</td>
</tr>
<tr>
<td>AN – AN</td>
<td>10</td>
<td>Mach 7.0</td>
<td>15</td>
</tr>
</tbody>
</table>

- Multihop case significantly harder
  - probability of stable end-to-end path very low
MANET Routing Algorithms
Update Mechanism: Predictive Example

- AeroRP airborne routing protocol
  - [Jabbar-Sterbenz-2009]
- Proactive with limited updates
  - eliminates delay of reactive
  - while limiting overhead of proactive
- Exploiting cross-layer information
  - explicit cross-layering provided by AeroNP
  - predictive using geolocation and trajectory information
- Snooping to assist neighbor discovery
  - overheard transmissions indicate neighbor presence
MANET Routing Algorithms
Design Alternatives: Resource / Context Aware

- Routing algorithm type
- Topological structure
- Routing update mechanism
- Resource and context awareness
  - power- and energy-aware
  - geographical
  - communication environment
MANET Routing Algorithms
Design Alternatives: Resource Aware

- Resource and context awareness
  - consider resources in routing algorithm
  - generally to conserve battery usage *lecture EM*
    - processing, memory, bandwidth, transmission power
    - remaining battery life
MANET Routing Algorithms

Design Alternatives: Context Aware

- Resource and context awareness
  - consider context of node in routing algorithm
  - geographical location-based routing
    - geolocation as addressing (e.g. sensor networks)  
    - motion trajectories for high-mobility scenarios

lecture SN
MANET Routing Algorithms
Design Alternatives: Context Aware

- Resource and context awareness
  - consider context of node in routing algorithm
  - communications environment
    - disruption-tolerant routing *lecture RS*
MANET Routing

MR.2  Example Protocols

MR.2.1  DSDV
MR.2.2  AODV
MR.2.3  DSR
MR.2.4  ZRP
MR.2.5  OLSR
What is the simplest possible routing algorithm?
MANET Routing Algorithms

Flooding

- Flooding: send all packets to all nodes
  - simplest possible routing algorithm
MANET Routing Algorithms

Flooding

- Flooding: send all packets to all nodes
  - simplest possible routing algorithm
MANET Routing Algorithms

Flooding

- Flooding: send all packets to all nodes
  - simplest possible routing algorithm
MANET Routing Algorithms

Flooding

- Flooding: send all packets to all nodes
  - simplest possible routing algorithm

Diagram of a network showing node connections and routing paths.
MANET Routing Algorithms

Flooding

- Flooding: send all packets to all nodes
  - simplest possible routing algorithm
MANET Routing Algorithms

Flooding

- Flooding: send all packets to all nodes
  - simplest possible routing algorithm
MANET Routing Algorithms

Flooding

- Flooding: send all packets to all nodes
  - simplest possible routing algorithm

*Advantages?*
Flooding

- Flooding: send all packets to all nodes
- Advantages:
  - simplest possible routing algorithm
  - use sequence numbers to avoid forwarding duplicates
    - each node discards an incoming packet already forwarded
  - may increase probability of delivery
    *why?*
MANET Routing Algorithms

Flooding

- Flooding: send all packets to all nodes
- Advantages:
  + simplest possible routing algorithm
  + use sequence numbers to avoid forwarding duplicates
    - each node discards an incoming packet already forwarded
  + may increase probability of delivery
    - every possible path explored

Disadvantages?
MANET Routing Algorithms

Flooding

- Flooding: send all packets to all nodes
- Advantages:
  - simplest possible routing algorithm
  - use sequence numbers to avoid forwarding duplicates
  - may increase probability of delivery
- Disadvantages
  - many packets sent that are not needed
  - bandwidth intensive in a bandwidth constrained environment
  - may reduce probability of delivery

*why?
MANET Routing Algorithms

Flooding

- Flooding: send all packets to all nodes
- Advantages:
  - simplest possible routing algorithm
  - use sequence numbers to avoid forwarding duplicates
  - may increase probability of delivery
- Disadvantages
  - many packets sent that are not needed
  - bandwidth intensive in a bandwidth constrained environment
  - may reduce probability of delivery

Alternative?
MANET Routing Algorithms
Flooding Alternatives

• Alternative to flooding
  – use *routing algorithm* to find possible path
  – *forward* data packets directly along that path

*How to find possible paths?*
MANET Routing Algorithms

Flooding Alternatives

- Alternative to flooding
  - use routing algorithm to find possible path
  - forward data packets directly along that path

- To find possible paths
  - flood control messages: *route discovery*

  *how is this different?*
MANET Routing Algorithms
Flooding Alternatives

- Alternative to flooding
  - use routing algorithm to find possible path
  - forward data packets directly along that path

- To find possible paths
  - flood control messages
    - still flooding, but much lower overhead than every data packet
    - reuse path for packets in flow
    - cache route for subsequent flows
  - exploit other knowledge
    - location management so sender can efficiently track receiver
MANET Routing Algorithms
Flooding Alternatives

- Alternative to flooding
  - use routing algorithm to find possible path
  - forward data packets directly along that path
- To find possible paths
  - flood control messages
    - still flooding, but much lower overhead than every data packet
    - reuse path for packets in flow
    - cache route for subsequent flows
  - exploit other knowledge
    - location management so sender can efficiently track receiver
- There are many proposed MANET routing protocols
MANET Routing Protocols

Examples: IETF Experimental

- There are *many* proposed MANET routing protocols
- A few are part of IETF MANET working group
  
  
  - currently all RFCs are "experimental" status
    
    - increases chance of implementation in products
    - RFC 3561: AODV (ad hoc on-demand distance vector protocol)
    - RFC 3626: OLSR (optimized link state routing protocol)
    - RFC 3684: TBRPF (topology broadcast based on RPF)
      
      - RPF (reverse path forwarding)
    - RFC 4728: DSR (dynamic source routing protocol)
MANET Routing Protocols
Examples: Current IETF Internet Drafts

• There are *many* proposed MANET routing protocols
• A few are part of IETF MANET working group
  – currently all RFCs are “experimental” status
  – some in proposals are currently in Internet Draft form
    • DYMO (dynamic MANET on-demand routing) – AODV successor
    • OLSR version 2
    • SMF (simplified multicast forwarding)
  – IDs (Internet drafts)
    • can be part of IETF working group agenda
    • can be sponsored by individuals
MANET Routing Protocols
Examples: Dead IETF Internet Drafts

- There are *many* proposed MANET routing protocols
- A few are part of IETF MANET working group
  - currently all RFCs are “experimental” status
  - some in proposals are currently in Internet Draft form
  - some proposals never became RFCs
    - ABR (associativity based routing)
    - MM (mobile mesh)
    - TORA (temporally ordered routing algorithm)
    - ZRP (zone routing protocol)
    etc.
There are many proposed MANET routing protocols. A few are part of IETF MANET working group. Many more research proposals:

- DSDV (destination sequenced distance vector)
- WRP (wireless routing protocol)
  etc.
- some designed for very specialised domains
  - tactical military networks
  - vehicles and aircraft (e.g. AeroRP)
  - disruption-tolerant networks (e.g. P-WARP)
- IETF RFC status ≠ probability of deployment
MANET Routing Protocols

Examples

• Each proposed protocol could be an entire lecture
  – we will only provide an overview of a selection
  – somewhat arbitrary choice
    • based on historical significance
    • based on how well known
  – remember: *none* of these have seen significant deployment
### MANET Routing Protocols Examples

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Routing Algorithm</th>
<th>Topological Structure</th>
<th>Update Mechanism</th>
<th>Resource Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSDV</td>
<td>distance vector</td>
<td>flat</td>
<td>proactive</td>
<td>none</td>
</tr>
<tr>
<td>AODV</td>
<td>distance vector</td>
<td>flat</td>
<td>reactive</td>
<td>none</td>
</tr>
<tr>
<td>DSR</td>
<td>source routing</td>
<td>flat</td>
<td>reactive</td>
<td>none</td>
</tr>
<tr>
<td>ZRP</td>
<td>varies</td>
<td>flat</td>
<td>hybrid</td>
<td>none</td>
</tr>
<tr>
<td>OLSR</td>
<td>link state</td>
<td>flat</td>
<td>proactive</td>
<td>none</td>
</tr>
</tbody>
</table>
MANET Routing Protocols

Network Layer Functionality

- **Addressing:** node identifiers
- **Routing:** path discovery
- **Forwarding:** next-hop decision and transfer
- **Signalling:** MANET control messages
- **Traffic management**
MANET Routing Protocols

Addressing Alternatives

- Addressing: node identifiers

alternative possibilities?
MANET Routing Protocols
Addressing Alternatives

- **Addressing**: node identifiers
  - alternative possibilities
- **IEEE 48b MAC addresses**
  - de facto standard for unique network interface addressing
  - much larger addresses than needed for typical MANET
- **IPv4 or IPv6 addresses**
  - de facto standard for network addresses
  - useful when MANET connected to Internet
- **MANET protocol-specific identifiers**
  - flat: address space related to maximum size of MANET
  - hierarchical: size relate to cluster size and number of levels
MANET Routing Protocols
Implementation Alternatives

- Implementation alternatives
  - standalone
  - IP integration
  - IP encapsulation
  - UDP/IP encapsulation
MANET Routing Protocols

Implementation Options: Standalone

- **Standalone**
  - MANET packet (encapsulated in a link/MAC frame)
  - no dependencies on TCP/IP
- **Node addresses in MANET header**
- **Alternative address schemes:**
  - protocol-specific identifiers (flat or hierarchical)
  - IPv4, IPv6, or IEEE 48b MAC addresses can still be used
- **Payload (determined by header bit) consists of either**
  - MANET signalling message
  - data
MANET Routing Protocols
Implementation Options: IP Integration

- IP integration
  - MANET packet (encapsulated in a link/MAC frame)
- Node addresses in IP header (originating, target)
  - other IP fields used such as TTL
  - extended by MANET subheader
  - additional addresses and control fields
- Payload consists of either
  - MANET signalling message
- Example: DSR (IP protocol ID=48)
  - MANET subheader contains protocol ID of payload
MANET Routing Protocols

Implementation Options: IP Encapsulation

- MANET signalling message encapsulated in IP
  - IP protocol number defines MANET protocol
- MANET data messages conventional UDP
  - using IP source = origin node, destination = target node
- IP protocol IDs
  - 138: generic MANET
MANET Routing Protocols

Implementation Options: UDP Encapsulation

<table>
<thead>
<tr>
<th>link</th>
<th>IP</th>
<th>UDP</th>
<th>MANET</th>
<th>payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>269</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **MANET signalling message** encapsulated in **UDP**
  - UDP port number defines MANET protocol
- **MANET data messages** conventional **UDP**
  - using IP source = origin node, destination = target node
- **Examples and UDP ports**
  - 269: generic MANET
  - 654: AODV
  - 698: OLDR
MANET Routing Protocols

MR.2.2 DSDV

MR.1 Routing algorithm alternatives
MR.2 Example protocols
  MR.2.1 DSDV
  MR.2.2 AODV
  MR.2.3 DSR
  MR.2.4 ZRP
  MR.2.5 OLSR
DSDV
Overview

• DSDV: destination sequenced distance vector
  – one of the first MANET routing protocols [Perkins 1994]
DSDV
Overview

• DSDV: destination sequenced distance vector
  – one of the first MANET routing protocols [Perkins 1994]
• Table driven / proactive
  – each node maintains table
  – row for each possible destination
    • next hop to reach destination
    • number of hops to destination
  – sequence number to know which is most recent update
DSDV
Overview

• DSDV: destination sequenced distance vector
  – one of the first MANET routing protocols [Perkins 1994]
• Table driven / proactive
• Updates exchanged between immediate neighbours
  – single update when topology change at a given node
  – full dump when significant topology change at a given node
  – periodic synchronisation
DSDV
Example Network and Table

<table>
<thead>
<tr>
<th>Dest</th>
<th>Next</th>
<th>Dist</th>
<th>Seq#</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>22</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>2</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>1</td>
<td>134</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>1</td>
<td>144</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>3</td>
<td>162</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>3</td>
<td>170</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>4</td>
<td>186</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>2</td>
<td>142</td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td>3</td>
<td>176</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>3</td>
<td>190</td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td>4</td>
<td>198</td>
</tr>
<tr>
<td>14</td>
<td>6</td>
<td>3</td>
<td>214</td>
</tr>
<tr>
<td>15</td>
<td>5</td>
<td>4</td>
<td>256</td>
</tr>
</tbody>
</table>
DSDV

Example Network and Table

<table>
<thead>
<tr>
<th>Dest</th>
<th>Next</th>
<th>Dist</th>
<th>Seq#</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>22</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>2</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>1</td>
<td>134</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>1</td>
<td>144</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>3</td>
<td>162</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>3</td>
<td>170</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>4</td>
<td>186</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>2</td>
<td>142</td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td>3</td>
<td>176</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>3</td>
<td>190</td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td>4</td>
<td>198</td>
</tr>
<tr>
<td>14</td>
<td>6</td>
<td>3</td>
<td>214</td>
</tr>
<tr>
<td>15</td>
<td>5</td>
<td>4</td>
<td>256</td>
</tr>
</tbody>
</table>
DSDV
Example Network and Table

<table>
<thead>
<tr>
<th>Dest</th>
<th>Next</th>
<th>Dist</th>
<th>Seq#</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>22</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>2</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>1</td>
<td>134</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>1</td>
<td>144</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>3</td>
<td>162</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>3</td>
<td>170</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>4</td>
<td>186</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>2</td>
<td>142</td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td>3</td>
<td>176</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>3</td>
<td>190</td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td>4</td>
<td>198</td>
</tr>
<tr>
<td>14</td>
<td>6</td>
<td>3</td>
<td>214</td>
</tr>
<tr>
<td>15</td>
<td>5</td>
<td>4</td>
<td>256</td>
</tr>
</tbody>
</table>
DSDV

Example Network and Table

<table>
<thead>
<tr>
<th>Dest</th>
<th>Next</th>
<th>Dist</th>
<th>Seq#</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>22</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>2</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>1</td>
<td>134</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>1</td>
<td>144</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>3</td>
<td>162</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>3</td>
<td>170</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>4</td>
<td>186</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>2</td>
<td>142</td>
</tr>
<tr>
<td>11</td>
<td>5</td>
<td>4</td>
<td>180</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>3</td>
<td>190</td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td>4</td>
<td>198</td>
</tr>
<tr>
<td>14</td>
<td>6</td>
<td>3</td>
<td>214</td>
</tr>
<tr>
<td>15</td>
<td>5</td>
<td>4</td>
<td>256</td>
</tr>
</tbody>
</table>
DSDV

Advantages and Disadvantages

- Advantages
  - minor adaptation of wired distance vector protocol
DSDV

Advantages and Disadvantages

• Advantages
  – minor adaptation of wired distance vector protocol

• Disadvantages
  – high overhead with non-trivial changes
    • mobility
    • episodic link connectivity
  – stale information before updates propagate
    • packets may be forwarded along wrong path
WRP
Overview

- **WRP**: wireless routing protocol
  - one of the first MANET routing protocols [Murthy, JJ 1996]
  - similar in concept to DSDV

- **Table driven / proactive**
  - each node maintains table
  - multiple tables for more accurate information
    - distance
    - routing: distance, predecessor, successor, status
    - link cost
    - message retransmission list
MANET Routing Protocols

**MR.2.2** AODV

**MR.1** Routing algorithm alternatives

**MR.2** Example protocols

**MR.2.1** DSDV

**MR.2.2** AODV

**MR.2.3** DSR

**MR.2.4** ZRP

**MR.2.5** OLSR
AODV
Overview

• AODV (ad hoc on-demand distance vector)
  – early MANET routing protocol [Perkins, Belding-Royer 1999]
  – adopted by IETF MANET working group [RFC 3561]

• On demand / reactive successor to DSDV
  – each node has forwarding table
  – paths only included when needed (reactive)
AODV
Message Types

- AODV packets are for control only
  - encapsulate in UDP in IP in MAC frame
  - UDP port 654 for AODV
- Message types (AODV is only a routing protocol)
  1 = RREQ route request to discover path
  2 = RREP route reply to confirm path
  3 = RERR route error when link goes down
  4 = RREP-ACK route reply ACK
AODV Operation

Route Discovery

- Flood route request: RREQ message path not known
  - using broadcast IP address 255.255.255.255
  - IP TTL determines scope of flood
- Route discovery occurs when:
  - destination node previously unknown to node
  - previously valid route expires
  - previously valid route marked as invalid by RERR
- RREQ for *destination node* by *originating node*
AODV Route Discovery

RREQ Packet Format

- **RREQ**: type = 01
- **Flags**
  - **J**: join (multicast)
  - **R**: repair (multicast)
  - **G**: gratuitous RREP should be unicast to destination
  - **D**: destination only can reply to this RREQ
  - **U**: unknown destination seq #
  - remaining bits reserved

<table>
<thead>
<tr>
<th>type = 01</th>
<th>JRGDU</th>
<th>hop count</th>
</tr>
</thead>
<tbody>
<tr>
<td>RREQ ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>destination IP address</td>
<td></td>
<td></td>
</tr>
<tr>
<td>destination sequence number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>originator IP address</td>
<td></td>
<td></td>
</tr>
<tr>
<td>originator sequence number</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
AODV Route Discovery
RREQ Packet Format

- **RREQ**: type = 01
- **Flags**
- **Hop count**
  - number of hops from originator to this node
- **RREQ ID**
  - identifier of RREQ for given originator
  - in combination with IP address unique ID for each RREQ

<table>
<thead>
<tr>
<th>type = 01</th>
<th>JRGDU</th>
<th>hop count</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RREQ ID</td>
<td></td>
</tr>
<tr>
<td></td>
<td>destination IP address</td>
<td></td>
</tr>
<tr>
<td></td>
<td>destination sequence number</td>
<td></td>
</tr>
<tr>
<td></td>
<td>originator IP address</td>
<td></td>
</tr>
<tr>
<td></td>
<td>originator sequence number</td>
<td></td>
</tr>
</tbody>
</table>
### AODV Route Discovery

**Operation: RREQ Packet Format**

- **RREQ**: type = 01
- Flags
- Hop count
- RREQ ID
- Destination IP address of desired path
- Destination sequence number
- Originator IP address of RREQ
- Originator sequence number

<table>
<thead>
<tr>
<th>type = 01</th>
<th>JRGDU</th>
<th>hop count</th>
</tr>
</thead>
<tbody>
<tr>
<td>RREQ ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>destination IP address</td>
<td></td>
<td></td>
</tr>
<tr>
<td>destination sequence number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>originator IP address</td>
<td></td>
<td></td>
</tr>
<tr>
<td>originator sequence number</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
AODV Operation
Route Discovery: RREQ Flood

• Determine sequence numbers and ID
  – dest seq# is last used from routing table or set U flag if none
  – orig seq#
  – RREQ ID is incremented from last used by originator

• Rate
  – rate of RREQ messages limited to RREQ_RATELIMIT
    • default = 10 msg/sec
AODV
Operation: Route Discovery

• Every intermediate node stores reverse path
  – bidirectional links required
• RREP returned to originator IP address when
  – fresh route located in intermediate node
  – destination receives the *first* RREQ from a given originator
  – this constructs distance vector (not Bellman-Ford)
• RREP uses reverse pointers in each node
  – establishes predecessor and successor nodes for a flow
• Forwarding table entries used by multiple flows
  – soft state: entries time out
AODV Route Discovery

RREP Packet Format

- **RREP**: type = 02
- **Flags**
  - **R**: repair (multicast)
  - **A**: ACK required
- **Prefix size**
- **Hop count**
  - # hops orig→dest
- **Lifetime**
  - timer to prevent stale RREPs from being used

<table>
<thead>
<tr>
<th>type = 02</th>
<th>RA</th>
<th>prefix sz</th>
<th>hop count</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>destination IP address</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>destination sequence number</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>originator IP address</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>lifetime</td>
<td></td>
</tr>
</tbody>
</table>
AODV
Operation: Route Maintenance

- Adjacent nodes exchange periodic HELLO messages
  - keepalive function
  - route times out if no messages received
- Link failure detected
  - forwarding failure
  - timeout of HELLO messages
  - failure to receive MAC-layer ACKs
- Route error (RERR) messages to neighbours
  - sequence numbers avoid broken paths and loops
AODV
Operation: Data Transfer

- Data packets forwarded hop-by-hop to destination
  - each hop uses forwarding table to lookup next hop
AODV

Advantages and Disadvantages

Advantages?
AODV
Advantages

- Advantages
  + relatively simple algorithm
  + reduces flooding of control messages
    • but flooded for every source–destination pair
  + paths maintained only for needed routes
  + soft state allows amortisation over...
    • time: multiple flows
  + good for long flows in a relatively stable network

Disadvantages?
AODV
Disadvantages

- Disadvantages
  - mobility and episodic connectivity *significantly* impact
  - flooding of RREQ control messages
    - may reach all nodes in a large network
    - required for every source–destination pair
  - stale forwarding table entries lead to inconsistent routes
  - overhead of HELLO messages
AODV Deployment

- 802.11s uses HWMP (hybrid wireless mesh protocol)
  - based in part on AODV

*why is this a good choice for 802.11s mesh networking?*
AODV Deployment

- 802.11s uses HWMP (hybrid wireless mesh protocol)
  - based in part on AODV
  - AODV: self-organising
  - AODV: occasional topology changes
  - AODV: small networks
MANET Routing Protocols

MR.2.3  DSR

MR.1  Routing algorithm alternatives
MR.2  Example protocols
  MR.2.1  DSDV
  MR.2.2  AODV
  MR.2.3  DSR
  MR.2.4  ZRP
  MR.2.5  OLSR
DSR
Overview

• DSR (dynamic source routing)
  – early MANET routing protocol [Johnson, Maltz 1996]
  – adopted by IETF MANET working group [RFC 4728]

• On demand / reactive
  – source routing
DSR
Packet Format: IP Header

- **IP header**
  - prot id = 48
  - IHL = 20
  - source addr [32b]  
    source node address
  - destination addr [32b]  
    controls dissemination
  - TTL [ 8b]  
    limits scope of flooding
- Fixed option hdr [32b]
- Option
**DSR**

Packet Format: Fixed Option Header

- **IP header**
- **Fixed option header** [32b]
  - next hdr [8b]
    - IP protocol id of encapsulated transport
  - F [1b]
    - flow state header flag
  - payld len [16b]
    - length DSR options hdr
- **Option** [length]
DSR
Option Types

- DSR options are shim header following IP header
  - IP protocol ID = 48 reserved for DSR
- Option types (DSR is *only* a routing protocol)
  
  1 = RREQ route request path discovery
  2 = RREP route reply path establishment
  3 = RERR route error
  160 = acknowledgement request
  32 = acknowledgement
  96 = DSR source route option
  224 = Pad1 option
  0 = PadN option
DSR
Operation: Route Discovery

- When source does not know path to destination:
- Flood route request (RREQ)
  - every node appends its 32-bit address to RREQ
  - constructs source route while packet is travelling to target
DSR

Packet Format: RREQ Packet

- **IP header** [20B]
  - **source addr** [32b]
    originating address
  - **destination addr** [32b]
    IP broadcast to flood
    255.255.255.255
  - **TTL** = [1...255] [8b]
    controls scope of flood
- **Fixed option hdr** [4B]
- **Option** [8+4nB]

<table>
<thead>
<tr>
<th>Field</th>
<th>Bits</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>total length</td>
<td>16</td>
<td>total length of packet</td>
</tr>
<tr>
<td>TOS</td>
<td>8</td>
<td>traffic class</td>
</tr>
<tr>
<td>fragment id</td>
<td>16</td>
<td>fragment identifier</td>
</tr>
<tr>
<td>frag offset</td>
<td>16</td>
<td>fragment offset</td>
</tr>
<tr>
<td>header checksum</td>
<td>32</td>
<td>checksum</td>
</tr>
<tr>
<td>source address = orig. address</td>
<td>32</td>
<td>source address = orig. address</td>
</tr>
</tbody>
</table>
DSR
Packet Format: RREQ Packet

- IP header [20B]
- Fixed option hdr [4B]
- Option [8+4nB] 24B
  - type = 1 RREQ [8b]
  - unique pkt id [16b]
  - target address [32b]
  - addresses [32b]
    each node adds its address to this list to form source route
**DSR**

**Operation: Route Discovery**

- When source does not know path to destination:
- Flood route request (RREQ)
  - every node appends its ID to RREQ
  - constructs source route while packet is travelling to target
- Target sends route reply (RREP)
  - to first RREQ received
- Two possibilities
  - for bidirectional links (e.g. symmetric transmit power)
    - use reverse accumulated source path in RREQ
  - for asymmetric links
    - RREQ needed back to source if path not known
DSR
Packet Format: RREP Packet

- **IP header** [20B]
  - **source addr** [32b]
    - DSR target address: node replying to RREQ
  - **destination addr** [32b]
    - DSR originator address initiated RREQ
  - **TTL** = [1...255] [8b]
- **Fixed option hdr** [1B]
- **Option**

<table>
<thead>
<tr>
<th>04</th>
<th>20</th>
<th>TOS</th>
<th>total length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>fragment id</th>
<th>D M F F</th>
<th>frag offset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TTL</th>
<th>prot=48</th>
<th>header checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| source address = target node |
| destination address = orig. node |

<table>
<thead>
<tr>
<th>nxt=59</th>
<th>F</th>
<th>resv</th>
<th>payload length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>type=2</th>
<th>opt len</th>
<th>L</th>
<th>resv</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| address[1] |
| address[1] |
|           |
|           |

| address[n] |
|           |
|           |
DSR
Packet Format: RREP Packet

- IP header [20B]
- Fixed option hdr [1B]
- Option
  - type = 2 RREP [8b]
  - unique pkt id [16b]
  - addresses [32b]
    source route copied from RREQ returned to originator

```
<table>
<thead>
<tr>
<th></th>
<th></th>
<th>TOS</th>
<th></th>
<th>total length</th>
</tr>
</thead>
<tbody>
<tr>
<td>04</td>
<td>20</td>
<td></td>
<td>fragment id</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 D M F F</td>
<td>frag offset</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TTL</td>
<td></td>
<td>prot=48</td>
<td></td>
<td>header checksum</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>source address = target node</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>destination address = orig. node</td>
</tr>
<tr>
<td>nxt=59</td>
<td></td>
<td>F</td>
<td>resv</td>
<td>payload length</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>type=2</td>
<td>opt len</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td>resv</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>address[1]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>address[1]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>address[n]</td>
<td></td>
</tr>
</tbody>
</table>
```
DSR

Operation: Route Caching

- When source receives RREP from destination
- Route cached

why?
DSR
Operation: Route Caching

- When source receives RREP from destination
- Route cached
  - used by all packets in a flow
  - can be used by other flows to same destination
  - all intermediate sub-paths cached for other flows
    - maintained in a tree data structure
  - paths cached in both directions
  - paths cached by other nodes overhearing RREQ and RREP
- Route caching can significantly reduce flooding

*Disadvantage?*
DSR
Operation: Route Caching

- When source receives RREP from destination
- Route cached
- Route caching can significantly reduce flooding
- Stale cache significantly impacts performance
  - due to mobility or episodic connectivity
  - multiple stale routes may be tried before success or new RREQ
DSR
Operation: Data Transfer

- Data packets sent hop-by-hop to destination
  - using cached source route
  - each hop pops next hop from packet header
  - packet header length proportional to number of hops
DSR
Advantages and Disadvantages

Advantages?
DSR
Advantages

• Advantages
  – simple algorithm
  – reduces flooding of control messages
  – paths maintained only for needed routes
    • plus subpaths
  – caching allows amortisation over
    • time: multiple flows
    • space: locality of flow endpoints
  – good for long flows in a relatively stable network

Disadvantages?
DSR

Disadvantages

- Disadvantages
  - mobility and episodic connectivity *significantly* impact
  - flooding of RREQ and RREP control messages
    - may reach all nodes in a large network
    - collisions between adjacent nodes during flood
      - insert random delays to ameliorate
  - header length grows with network scale
    - cost paid on every packet
  - intermediate node may send RREP using stale cache
    - poisons other caches
    - may be ameliorated by adding cache purge messages
MANET Routing Protocols

MR.2.4 ZRP

MR.1 Routing algorithm alternatives

MR.2 Example protocols
  MR.2.1 DSDV
  MR.2.2 AODV
  MR.2.3 DSR
  MR.2.4 ZRP
  MR.2.5 OLSR
ZRP
Overview

• ZRP (zone routing protocol)
  – early MANET routing protocol [Haas 1997]
  – not adopted by IETF MANET working group
    • specification exists in expired Internet drafts

• Hybrid proactive/reactive
  – structured into 2-level zone based on hop-count $d$
  – proactive intra-zone over short distances $\leq d$
  – reactive inter-zone RREQ discovery for $> d$ hops
ZRP
Advantages and Disadvantages

• Advantages
  – reduces overhead of RREQ for intra-zone nodes

• Disadvantages
  – sensitive to zone radius for given node density
  – high overhead due to zone overlap
MANET Routing Protocols

MR.2.5 OLSR

MR.1 Routing algorithm alternatives

MR.2 Example protocols
  MR.2.1 DSDV
  MR.2.2 AODV
  MR.2.3 DSR
  MR.2.4 ZRP
  MR.2.5 OLSR
• OLSR (optimized link state routing)
  – 2nd generation MANET routing protocol [Clausen et al. 2001]
  – adopted by IETF MANET working group [RFC 3626]
OLSR
Overview

• OLSR (optimized link state routing)
  – 2nd generation MANET routing protocol [Clausen et al. 2001]
  – adopted by IETF MANET working group [RFC 3626]

• Table driven / proactive
  – link state

Why can’t we just use OSPF?
OLSR
Differences from Wired Link State

• Table driven / proactive
  – link state
• But there are no links in the conventional sense
  – OSPF and ISIS track the state of each physical link
  – MANET has only ephemeral links to reachable neighbours

Problem?
OLSR

Differences from Wired Link State

• Table driven / proactive
  – link state

• But there are no links in the conventional sense
  – OSPF and ISIS track the state of each physical link
  – MANET has only ephemeral links to reachable neighbours

• Problem: frequent topology & connectivity changes
  – result in severe overhead due to frequent flooding
  – every change in node pair connectivity would need LSA
### OLSR Information Bases

- Each node accumulates and maintains information
  - repositories of information: IBs (information bases)
- **OLSR IBs**
  - multiple interface association information base
  - local link information base
  - neighborhood information base
  - topology information base
- **Soft state that must be refreshed**
  - time fields determine when IP tuples must be removed
Multiple interface association information base
  - nodes may have multiple interface addresses

Nodes keep interface addresses from MID messages
  - neighbor tuple describing set of immediate neighbors
    - I_iface_addr  interface address of node
    - I_main_addr   main address of node
    - I_time        expiration time of this tuple to be removed
OLSR
Local Link IB

- Local link information base
  - neighbors discovered using HELLO messages
- Node records link tuples form incoming HELLOs
  - neighbor tuple describing set of immediate neighbors
  - L\_local\_iface\_addr local interface address
  - L\_neighbor\_iface\_addr neighbor interface address
  - L\_SYM\_time time until link considered symmetric
  - L\_ASYM\_time time until neighbor considered heard
  - L\_time expiration time of this tuple to be removed
OLSR

Neighborhood IB

- Neighborhood information base
  - stores information about local neighbourhood
  - neighbor set, 2-hop neighbor set, MPR set, MPR selector set
**OLSR**

**Neighborhood IB: Neighbors**

- Neighborhood information base
  - stores information about local neighborhood
- Neighbor set

- 2-hop neighbor set
- Multipoint relay (MPR) set
- MPR selector set
OLSR

Neighborhood IB: Neighbors

- Neighborhood information base
  - stores information about local neighborhood
- Neighbor set
  - neighbor tuple describing set of immediate neighbors
OLSR

Neighborhood IB: Neighbors

• Neighborhood information base
  – stores information about local neighborhood

• Neighbor set
  – neighbor tuple describing set of immediate neighbors
  – \textit{N\_neighbor\_main\_addr} node main address
  – \textit{N\_status} node symmetry
  – \textit{N\_willingness} transit traffic willingness

• 2-hop neighbor set

• Multipoint relay (MPR) set

• MPR selector set
OLSR

Neighborhood IB: 2-Hop Neighbors

- Neighborhood information base
  - stores information about local neighborhood
- Neighbor set
- 2-hop neighbor set
  - 2-hop tuple describing set of 1- and 2-hop neighbors

- Multipoint relay (MPR) set
- MPR selector set
OLSR

Neighborhood IB: 2-Hop Neighbors

• Neighborhood information base
  – stores information about local neighborhood

• 2-hop neighbor set
  – 2-hop tuple describing set of 1- and 2-hop neighbors
OLSR

Neighborhood IB: 2-Hop Neighbors

• Neighborhood information base
  – stores information about local neighborhood

• Neighbor set

• 2-hop neighbor set
  – 2-hop tuple describing set of 1- and 2-hop neighbors
  – N_neighbor_main_addr 1-hop node main address
  – N_2hop_addr 2-hop node main address
  – N_time expiration time of this tuple to be removed

• Multipoint relay (MPR) set

• MPR selector set
OLSR

Neighborhood IB: MultiPoint Relays

- Neighborhood information base
  - stores information about local neighborhood
- Neighbor set
- 2-hop neighbor set
- Multipoint relay (MPR) set
  - neighbors selected as MPRs
  - subset of neighbors that can reach all 2-hop neighbors
  - LSAs sent only to MPR set to reduce overhead
  - low-overhead MPR election algorithm heuristic
- MPR selector set
OLSR

Neighborhood IB: MultiPoint Relays

- Neighborhood information base
  - stores information about local neighborhood
- MPR set
  - neighbors selected as MPRs
  - subset of neighbors that can reach all 2-hop neighbors
OLSR

Neighborhood IB: MPR Selector Set

- Neighborhood information base
  - stores information about local neighborhood
- Neighbor set
- 2-hop neighbor set
- Multipoint relay (MPR) set
- MPR selector set
  - tuple of neighbors which have selected this node as MPR
  - MS_neighbor_main_addr node address
  - MS_time expiration time of this tuple to be removed
OLSR

Neighborhood IB: MPR Selector Set

- Neighborhood information base
  - stores information about local neighborhood
- MPR selector set
  - tuple of neighbors which have selected this node as MPR
OLSR
Topology Information Base

- Topology information base
  - topology information for entire network
  - data structure needed by all link state routing algorithms

- Topology tuple for all nodes:
  - $T_{\text{dest\_addr}}$ node address
  - $T_{\text{last\_addr}}$ one-hop away from $T_{\text{dest\_addr}}$
  - $T_{\text{seq}}$ sequence number
  - $T_{\text{time}}$ expiration time of this tuple to be removed
OLSR
Packet Format and Header

- Packet consists of sequence of messages
- Packet header [1B]
  - length [16b]
  - seq# [16b] per node interface
- Message headers [3B]
- Message bodies
OLSR
Packet Format: Message Header

• Message header
  – type [8b]
  – Vtime [8b]
    time to be kept valid
  – size [16b]
  – orig addr [32b]
  – TTL [8b]
    limits scope of flooding
  – hop cnt [8b]
  – seq# [16b]

• Message body [size – 12B]
OLSR
Message Types

• OLSR packets are for control only
  – as for OSPF
  – may be encapsulated in another L3 protocol
    • UDP port 698 reserved for OLSR
    • typically encapsulated in UDP in IP in MAC frame

• Message types (OLSR is only a routing protocol)
  1 = HELLO neighbor discovery and keepalive
  2 = TC topology control: link state advertisements
  3 = MID multiple interface declaration (for a given node)
  4 = HNA host and network association (OLSR–external GW)
• **HELLO message**
  - **HTime** [8b]
    HELLO time interval
  - **Willing** [8b]
    willingness to forward
    [0=never ... 7=always]
  - **link code** [8b]
    type (symmetry)
  - **link msg size**
    [1 + #(intfc addr)B]
  - **neigh intfc addr** [32b]
    list of addresses
**OLSR**

Message Format: **TC**

- **HELLO message**
  - **HTime** [8b]
    - HELLO time interval
  - **Willing** [8b]
    - Willingness to forward
      - \([0=\text{never} \ldots 7=\text{always}]\)
  - **link code** [8b]
    - Type (symmetry)
  - **link msg size**
    - \([1 + \#(\text{intfc addr})B]\)
  - **neigh intfc addr** [32b]
    - List of addresses
OLSR
Packet Processing

- Node examines Packet Type of received packet
  - duplicate tuple record created to ignore subsequent dups
1. Packets with invalid short length discarded
2. Packets with TTL ≤ 0 discarded
3. Duplicate packets discarded; else Type decoded
4. Message processed, and forwarded if applicable
OLSR

Packet Forwarding

1. If sender address not 1-hop neighbor, drop
2. If duplicate, check to see if retransmission
3. Drop if duplicate not retransmitted
4. If from MPR selector and TTL > 1 forward
5. Update duplicate tuple record
6. Decrease TTL by 1
7. Increase hop count by 1
8. Broadcast on all interfaces
### OLSR

**Message Format: Topology Control**

- **TC message**
  - **ANSN** [8b]
    - advertised neighbor sequence number: incremented with every topology change so stale updates don’t affect net
  - **neigh main addr** [32b]
    - list of MPR neighbors of originating node

<table>
<thead>
<tr>
<th>Packet Length</th>
<th>Sequence #</th>
</tr>
</thead>
<tbody>
<tr>
<td>type = 02</td>
<td>Vtime</td>
</tr>
<tr>
<td></td>
<td>message size</td>
</tr>
<tr>
<td></td>
<td>originator address</td>
</tr>
<tr>
<td>TTL=255</td>
<td>hop cnt</td>
</tr>
<tr>
<td></td>
<td>msg sequence #</td>
</tr>
<tr>
<td>ANSN</td>
<td>reserved</td>
</tr>
<tr>
<td>advertised neighbor main address</td>
<td></td>
</tr>
<tr>
<td>advertised neighbor main address</td>
<td></td>
</tr>
</tbody>
</table>

---

© James P.G. Sterbenz
OLSR
Advantages and Disadvantages

Advantages?
OLSR
Advantages and Disadvantages

- Advantages
  + lower overhead than other proactive algorithms
  + scalable to large networks
    - hierarchy could be added for very large networks

Disadvantages?
OLSR
Advantages and Disadvantages

• Advantages
  + lower overhead than other proactive algorithms
  + scalable to large networks
    • hierarchy could be added for very large networks

• Disadvantages
  – maintains routes even for unneeded paths
  – link state only up or down; no notion of weak connectivity
  – overhead of periodic LSAs
OLSR Deployment

- Used in some mesh deployments
  - OpenWRT on Linksys WRT54GS
  - but this is targeted for fixed mesh networking
    - rather than MANET
MANET Routing
Further Reading

MANET Routing

Acknowledgements

Some material in these foils is based on the textbook
- Murthy and Manoj,
  Ad Hoc Wireless Networks: 
  Architectures and Protocols

Some information in these foils derived from the tutorial
- Nitin H. Vaidya
  Mobile Ad Hoc Networks: Routing, MAC, and Transport Issues
  available from

Some material in these foils enhanced from EECS 780 foils