Algorithms for Radio Networks
Routing, Flooding, DSR
### Protocols of the Internet

<table>
<thead>
<tr>
<th>Layer</th>
<th>Protocols</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Application</strong></td>
<td>Telnet, FTP, HTTP, SMTP (E-Mail), ...</td>
</tr>
<tr>
<td><strong>Transport</strong></td>
<td>TCP (Transmission Control Protocol)</td>
</tr>
<tr>
<td></td>
<td>UDP (User Datagram Protocol)</td>
</tr>
<tr>
<td><strong>Network</strong></td>
<td>IP (%Internet Protocol%)</td>
</tr>
<tr>
<td></td>
<td>+ ICMP (%Internet Control Message Protocol%)</td>
</tr>
<tr>
<td></td>
<td>+ IGMP (%Internet Group Management Protocol%)</td>
</tr>
<tr>
<td><strong>Host-to-Network</strong></td>
<td>LAN (e.g. Ethernet, Token Ring etc.)</td>
</tr>
</tbody>
</table>
TCP/IP Layers

1. Host-to-Network
   • Not specified, depends on the local network, e.g. Ethernet, WLAN 802.11, PPP, DSL

2. Routing Layer/Network Layer (IP - Internet Protocol)
   • Defined packet format and protocol
   • Routing
   • Forwarding

3. Transport Layer
   • TCP (Transmission Control Protocol)
     - Reliable, connection-oriented transmission
     - Fragmentation, Flow Control, Multiplexing
   • UDP (User Datagram Protocol)
     - Hands packets over to IP
     - Unreliable, no flow control

4. Application Layer
   • Services such as TELNET, FTP, SMTP, HTTP, NNTP (for DNS), ...
Example: Routing between LANs

- FTP client
- FTP protocol
- FTP server
- TCP
- TCP protocol
- IP
- IP protocol
- router
- IP
- IP protocol
- Ethernet driver
- Ethernet protocol
- token ring driver
- token ring protocol
- token ring

Stevens, TCP/IP Illustrated
Routing Tables and Packet Forwarding

- **IP Routing Table**
  - contains for each destination the address of the next gateway
  - destination: host computer or sub-network
  - default gateway

- **Packet Forwarding**
  - IP packet (datagram) contains start IP address and destination IP address
    - if destination = my address then hand over to higher layer
    - if destination in routing table then forward packet to corresponding gateway
    - if destination IP subnet in routing table then forward packet to corresponding gateway
    - otherwise, use the default gateway
IP Packet Forwarding

- IP -Packet (datagram) contains...
  - TTL (Time-to-Live): Hop count limit
  - Start IP Address
  - Destination IP Address

- Packet Handling
  - Reduce TTL (Time to Live) by 1
  - If TTL ≠ 0 then forward packet according to routing table
  - If TTL = 0 or forwarding error (buffer full etc.):
    - delete packet
    - if packet is not an ICMP Packet then
      * send ICMP Packet with
        * start = current IP Address
        * destination = original start IP Address
Static and Dynamic Routing

- **Static Routing**
  - Routing table created manually
  - used in small LANs

- **Dynamic Routing**
  - Routing table created by Routing Algorithm
  - Centralized, e.g. Link State
    - Router knows the complete network topology
  - Decentralized, e.g. Distance Vector
    - Router knows gateways in its local neighborhood
Intra-AS Routing

- **Routing Information Protocol (RIP)**
  - Distance Vector Algorithmus
  - Metric = hop count
  - exchange of distance vectors (by UDP)

- **Interior Gateway Routing Protocol (IGRP)**
  - successor of RIP
  - different routing metrics (delay, bandwidth)

- **Open Shortest Path First (OSPF)**
  - Link State Routing (every router knows the topology)
  - Route calculation by Dijkstra’s shortest path algorithm
Distance Vector Routing Protocol

- **Distance Table data structure**
  - Each node has a
    - Line for each possible destination
    - Column for any direct neighbors

- **Distributed algorithm**
  - Each node communicates only with its neighbors

- **Asynchronous operation**
  - Nodes do not need to exchange information in each round

- **Self-terminating**
  - Exchange unless no update is available
Distance Vector Routing Example

Graph Diagram:

- Nodes: A, B, C, D, E
- Edges and Weights:
  - A to B: 1
  - A to C: 3
  - B to C: 5
  - B to D: 1
  - B to E: 7
  - C to D: 1
  - C to E: 1

Routing Table:

<table>
<thead>
<tr>
<th>from A to</th>
<th>via</th>
<th>entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>1</td>
<td>B</td>
</tr>
<tr>
<td>C</td>
<td>6</td>
<td>C</td>
</tr>
<tr>
<td>D</td>
<td>2</td>
<td>B</td>
</tr>
<tr>
<td>E</td>
<td>7</td>
<td>C</td>
</tr>
</tbody>
</table>
Distance Vector Routing

<table>
<thead>
<tr>
<th>from A to</th>
<th>via</th>
<th>entry</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>B</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>-</td>
<td>3 C</td>
</tr>
<tr>
<td>D</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>E</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>from B to</th>
<th>via</th>
<th>entry</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>-</td>
<td>3 C</td>
</tr>
<tr>
<td>D</td>
<td>-</td>
<td>1 C</td>
</tr>
<tr>
<td>E</td>
<td>-</td>
<td>- 8 D</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>from C to</th>
<th>via</th>
<th>entry</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>E</td>
<td>-</td>
</tr>
<tr>
<td>A</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>5 B</td>
</tr>
<tr>
<td>D</td>
<td>-</td>
<td>8 E</td>
</tr>
<tr>
<td>E</td>
<td>-</td>
<td>1 E</td>
</tr>
</tbody>
</table>
“Count to Infinity” - Problem

- **Good news travels fast**
  - A new connection is quickly at hand

- **Bad news travels slowly**
  - Connection fails
  - Neighbors increase their distance mutually
  - "Count to Infinity" Problem
“Count to Infinity” - Problem

Routing Table entry

<table>
<thead>
<tr>
<th>from A</th>
<th>via</th>
<th>Routing Table entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>to B</td>
<td>2</td>
<td>B</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>B</td>
</tr>
</tbody>
</table>

Routing Table entry

<table>
<thead>
<tr>
<th>from B</th>
<th>via</th>
<th>Routing Table entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>to A</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>5</td>
<td>A</td>
</tr>
</tbody>
</table>

Routing Table entry

<table>
<thead>
<tr>
<th>from A</th>
<th>via</th>
<th>Routing Table entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>to B</td>
<td>2</td>
<td>B</td>
</tr>
<tr>
<td>C</td>
<td>7</td>
<td>B</td>
</tr>
</tbody>
</table>

Routing Table entry

<table>
<thead>
<tr>
<th>from B</th>
<th>via</th>
<th>Routing Table entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>to A</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>5</td>
<td>A</td>
</tr>
</tbody>
</table>

Routing Table entry

<table>
<thead>
<tr>
<th>from A</th>
<th>via</th>
<th>Routing Table entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>to B</td>
<td>2</td>
<td>B</td>
</tr>
<tr>
<td>C</td>
<td>7</td>
<td>B</td>
</tr>
</tbody>
</table>

Routing Table entry

<table>
<thead>
<tr>
<th>from B</th>
<th>via</th>
<th>Routing Table entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>to A</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>9</td>
<td>-</td>
</tr>
</tbody>
</table>

∞
Link-State Protocol

- **Link state routers**
  - exchange information using Link State Packets (LSP)
  - each node uses shortest path algorithm to compute the routing table
- **LSP contains**
  - ID of the node generating the packet
  - Cost of this node to any direct neighbors
  - Sequence-no. (SEQNO)
  - TTL field for that field (time to live)
- **Reliable flooding (Reliable Flooding)**
  - current LSP of each node are stored
  - Forward of LSP to all neighbors
    - except to be node where it has been received from
  - Periodically creation of new LSPs
    - with increasing SEQNO
  - Decrement TTL when LSPs are forwarded
Characteristics of routing in mobile ad hoc networks

- **Movement of participants**
  - Reconnecting and loss of connection is more common than in other wireless networks
  - Especially at high speed

- **Other performance criteria**
  - Route stability in the face of mobility
  - energy consumption
Unicast Routing

- Variety of protocols
  - Adaptations and new developments
- No protocol dominates the other in all situations
  - Solution: Adaptive protocols?
Routing in MANETs

- **Routing**
  - Determination of message paths
  - Transport of data

- **Protocol types**
  - **proactive**
    - Routing tables with updates
  - **reactive**
    - Repair of message paths only when necessary
  - **hybrid**
    - Combination of proactive and reactive
Routing Protocols

- **Proactive**
  - Routes are demand independent
  - Standard Link-State and Distance-Vector Protocols
    - Destination Sequenced Distance Vector (DSDV)
    - Optimized Link State Routing (OLSR)

- **Reactive**
  - Route are determined when needed
    - Dynamic Source Routing (DSR)
    - Ad hoc On-demand Distance Vector (AODV)
    - Dynamic MANET On-demand Routing Protocol
    - Temporally Ordered Routing Algorithm (TORA)

- **Hybrid**
  - combination of reactive and proactive
    - Zone Routing Protocol (ZRP)
    - Greedy Perimeter Stateless Routing (GPSR)

Source: Christian Schindelhauer

Algorithms for Radio Networks
Computer Networks and Telematics
University of Freiburg
Trade-Off

- Latency because of route discovery
  - Proactive protocols are faster
  - Reactive protocols need to find routes
- Overhead of Route discovery and maintenance
  - Reactive protocols have smaller overhead (number of messages)
  - Proactive protocols may have larger complexity
- Traffic-Pattern and mobility
  - decides which type of protocol is more efficient
Flooding

- **Algorithm**
  - Sender S broadcasts data packet to all neighbors
  - Each node receiving a new packet
    - broadcasts this packet
    - if it is not the receiver
- **Sequence numbers**
  - identifies messages to prevent duplicates
- **Packet always reaches the target**
  - if possible
Flooding Example
Flooding for Data Delivery

Broadcast

Packet for Receiver F
Flooding for Data Delivery

Possible collision at B
Flooding for Data Delivery

Nodes G, H, I do not receive the packet

Receiver F gets packet and stops
Flooding

- **Advantage**
  - simple and robust
  - the best approach for short packet lengths, small number of participants in highly mobile networks with light traffic

- **Disadvantage**
  - High overhead
  - Broadcasting is unreliable
    - lack of acknowledgements
    - hidden, exposed terminals lead to data loss or delay
Flooding

- Produces too many unnecessary (long) data packets
  - in the worst case, each participant sends each packet
  - many long transmissions collisions lead to long waiting times in the medium access

- Better approach:
  - Use of control packets for route determination
  - Flooding of control packet leads to DSR
Dynamic Source Routing (DSR)

- Johnson, Maltz

- Algorithm
  - Sender initiates route discovery by flooding of **Route-Request (RREQ)**-packets
    - Each forwarding node appends his ID to the RREQ-packet
  - The receiver generates the routing information from the RREQ packet by producing a **Route-Reply (RREP)**-packet
    - using the route information of the packet is sent back to the sender
  - Transmitter sends **data packet** along with route information to the receiver
DSR Example

A S B E F
C D G
H I
DSR Example

Broadcast

RREQ[S]

H
I

A

S

C

D

E

F

G

RREQ[S]
DSR Example

RREQ[S,A]

RREQ[S,C]

RREQ[S,D]
DSR Example

A

S

B

C

D

E

F

G

H

I

RREQ[S,A,E]

RREQ[S,C,B]
DSR Example

A

B

C

D

E

F

G

H

I

RREP[S,A,E,F]
DSR Example

RREP[S,A,E,F]
DSR Example

RREP[S,A,E,F]
DSR Example

Datenpaket[S,A,E,F]
Requirements

› Route Reply
  • requires bidirectional connections
  • unidirectional links
    - must be tested for symmetry
    - or Route-Reply must trigger its own route-request

› Data packet has all the routing information in the header
  • hence: Source-Routing

› Route determination
  • if no valid route is known
DSR Extensions and Modifications

- Intermediate nodes can cache information RREP
  - Problem: stale information
- Listening to control messages
  - can help to identify the topology
- Random delays for answers
  - To prevent many RREP-packets (Reply-Storm)
  - if many nodes know the answer (not for media access)
- Repair
  - If an error is detected then usually: route recalculation
  - Instead: a local change of the source route
- Cache Management
  - Mechanisms for the deletion of outdated cache information
DSR Optimization
Route Caching

- Each node stores information from all available
  - Header of data packets
  - Route Request
  - Route-Reply
  - partial paths
- From this information, a route reply is generated
DSR Route Caching

- [S, A, E, F]
- [A, S]
- [A, E, F]
- [E, A, S]
- [F, E, A, S]
- [B, A, S]
- [B, C, S]
- [D, C, S]
- [C, S]
DSR Route Caching

![DSR Route Caching Diagram](image-url)
DSR Route Caching

![Diagram of a network with nodes A, B, C, D, E, F, G, and H, and edges connecting them. The diagram includes route caching information such as [A,S], [A,E,F], [E,B,A,S], [E,A,S], [E,F], [F,E,A,S], RREP[G,F,E,A], [S,A,E,F], [B,A,S], [B,C,S], [C,S], [D,C,S], and [E,F].]
Data packet [G,F,E,A]
DSR Optimization
Route Caching

› If any information is incorrect
  • because a route no longer exists
  • then this path is deleted from the cache
  • alternative paths are used
  • or RREQ is generated

› Missing links are distributed by (RERR) packets in the network
Route Error

![Diagram of a network with nodes A, B, C, D, E, F, S, H, I, and G. The diagram shows a route error between nodes E and F, indicated by the RERR[E,F] message.]
DSR Discussion

- **Benefits**
  - Routes are maintained only between communicating nodes
  - Route caching reduces route search
  - Caches help many alternative routes to find

- **Disadvantages**
  - Header size grows with distance
  - Network may be flooded with route requests
  - Route-Reply-Storm
  - Outdated information may cause cache overhead
AODV

- Perkins, Royer
  - Ad hoc On-Demand Distance Vector Routing, IEEE Workshop on Mobile Computing Systems and Applications, 1999
- Reaktives Routing-Protokoll
- Reactive routing protocol
  - Improvement of DSR
  - no source routing
  - Distance Vector Tables
    - but only for nodes with demand
  - Sequence number to help identify outdated cache info
  - Nodes know the origin of a packet and update the routing table
AODV

- **Algorithm**
  - Route Request (RREQ) like in DSR
  - Intermediate nodes set a reverse pointer towards the sender
  - If the target is reached, a Route Reply (RREP) is sent
  - Route Reply follow the pointers

- **Assumption: symmetric connections**
AODV: Example
AODV: Example

Broadcast

H
I
S
A
B
C
D
E
F
G
AODV: Example
AODV: Example
AODV: Example
AODV: Example
AODV: Example
AODV: Example

![AODV Diagram]

RREP

S S F S S F

AODV: Example

H I

A B C D E F G

Network diagram with nodes A, B, C, D, E, F, G, S, and H. The diagram shows the process of route discovery in AODV protocol.
AODV: Example

Data Packet[F]
Route Reply in AODV

- **Intermediate nodes**
  - may send route-reply packets, if their cache information is up-to-date

- **Destination Sequence Numbers**
  - measure the up-to-dateness of the route information
  - AODV uses cached information less frequently than DSR
  - A new route request generates a greater destination sequence number
  - Intermediate nodes with a smaller sequence number may not generate a route reply (RREP) packets
Timeouts

- **Reverse pointers are deleted after a certain time**
  - RREP timeout allows the transmitter to go back
- **Routing table information to be deleted**
  - if they have not been used for some time
  - Then a new RREQ is triggered
Link Failure Reporting

- Neighbors of a node X are active,
  - if the routing table cache are not deleted
- If a link of the routing table is interrupted,
  - then all active neighbors are informed
- Link failures are distributed by Route Error (RERR) packets to the sender
  - also update the Destination Sequence Numbers
  - This creates new route request
AODV: Example

AODV: Example

A
S
B
E
F
G
H
I
S
S
S
S
S
S
F
RERR
S
F
S

Algorithms for Radio Networks
Christian Schindelhauer

Computer Networks and Telematics
University of Freiburg
Detection of Link Failure

- **Hello messages**
  - neighboring nodes periodically exchange hello packets from
  - Absence of this message indicates link failure

- **Alternative**
  - use information from MAC protocol
Sequence Numbers

- When a node receives a message with destination sequence number $N$
  - then this node sets its number to $N$
  - if it was smaller before
- In order to prevent loops
  - If A has not noticed the loss of link (C, D)
    - (for example, RERR is lost)
  - If C sends a RREQ
    - on path C-E-A
  - Without sequence numbers, a loop will be constructed
    - since A "knows" a path to D, this results in a loop (for instance, CEABC)
Sequence Numbers
Optimization
Expanding Ring Search

- Route Requests
  - *start with small time-to-live value (TTL)*
  - if no Route Reply (RREP) is received, the value is increased by a constant factor and resent
- This optimization is also applicable for DSR
Relax

Diagram with labeled points and connections.
Bellman Ford

A \rightarrow C -16
A \rightarrow B 2
B \rightarrow A: x-8
B \rightarrow C: x-6
C \rightarrow A: -10
C \rightarrow B: x-4

negative cost cycle
SOURCE ROUTING

Home

School