Wireless Sensor Networks

5. Routing

Christian Schindelhauer
Technische Fakultät
Rechnernetze und Telematik
Albert-Ludwigs-Universität Freiburg
- Application
  - Data transmission, e-mail, terminal, remote login
- Presentation
  - System-dependent presentation of the data (EBCDIC / ASCII)
- Session
  - Start, end, restart
- Transport
  - Segmentation, congestion
- Network
  - Routing
- Data Link
  - Checksums, flow control
- Physical
  - Mechanics, electrics
# 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

Stevens, TCP/IP Illustrated
- **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

\[ n! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot \ldots \cdot n \]
\[ 2^n = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 - \ldots \cdot 2 \]

\[ \frac{n \cdot (n-1)}{2} = \binom{n}{2} \]
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 Table for A

<table>
<thead>
<tr>
<th>from A</th>
<th>B</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>to</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>14</td>
</tr>
<tr>
<td>D</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>E</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

Distance Table for C

<table>
<thead>
<tr>
<th>from C</th>
<th>B</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>to A</td>
<td>3</td>
<td>11</td>
<td>18</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>9</td>
<td>21</td>
</tr>
<tr>
<td>D</td>
<td>6</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>E</td>
<td>7</td>
<td>5</td>
<td>10</td>
</tr>
</tbody>
</table>

Routing Table entry:

- B
- B
- B
- D
Inter-AS
Distance Vector Routing Example

![Graph with nodes A, B, C, D, and E connected by links with distances labeled.]

<table>
<thead>
<tr>
<th>from A</th>
<th>via</th>
<th>entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>C</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>E</td>
<td>7</td>
<td>4</td>
</tr>
</tbody>
</table>
### Table: Routing Table

<table>
<thead>
<tr>
<th>from B</th>
<th>via</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>C</td>
<td>-</td>
<td>D</td>
</tr>
<tr>
<td>D</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>E</td>
<td>-</td>
<td>8</td>
</tr>
</tbody>
</table>

### Diagram: Network Routing

![Network Diagram]

- **Path from B to C:**
  - B > A > C

- **Path from C to B:**
  - C > D > B

- **Path from C to E:**
  - C > D > E

- **Path from E to C:**
  - E > D > C

- **Path from E to B:**
  - E > D > B
“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

<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>to C</td>
<td>3</td>
<td>B</td>
</tr>
</tbody>
</table>

<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>to C</td>
<td>5</td>
<td>-</td>
</tr>
</tbody>
</table>

- (B, A) no route
- (B, A, B, C) no route

### Routing Table

<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>to C</td>
<td>7</td>
<td>B</td>
</tr>
</tbody>
</table>

<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>to C</td>
<td>5</td>
<td>-</td>
</tr>
</tbody>
</table>

- (B, A) no route
- (B, A, B, C) no route

### Routing Table

<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>to C</td>
<td>7</td>
<td>B</td>
</tr>
</tbody>
</table>

<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>to C</td>
<td>9</td>
<td>-</td>
</tr>
</tbody>
</table>

- (B, A) no route
- (B, A, B, C) no route

**Path Vector**

- B6DP
- B6DP
- B6DP

**Bouw Gateway Protocol**
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 und 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 und proactive
    - Zone Routing Protocol (**ZRP**)
    - Greedy Perimeter Stateless Routing (**GPSR**)
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
Packet for Receiver F
Possible collision at B
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