Lecture #18: Principles of Network Routing

Contents

- Basic Routing Concepts
- Routing Features and Elements
- Static Routing
- Adaptive Routing
- Distance Vector and Link State Routing
- Hierarchical Routing
- Broadcasting and Multicasting
- Routing in Internet
Basic Routing Concepts

Network layer functions

- **Connection control**: ...
- **Routing**: considerations associated with hop-by-hop services transparent to the underlying resources such as data link connections.
- **Addressing**: ...

Basic Concepts

- routing algorithms are main part of the NL
- on local level routing consist of the decision
  - on which output line to transmit an arriving packet (by datagram exchange model)
  - which input/output pair to assign to a virtual circuit during its set up (by VC model; session routing)
- broadcasting messages also require routing when the destination is another network
- formal goal of routing algorithms: finding the sink tree for each router
Routing Functions & Elements

- **Routing Functions**
  - Correctness
  - Simplicity
  - Robustness vs. Stability
    - too quick reflections of changed traffic conditions may cause instability
  - Fairness vs. Optimality
    - the optimization criteria may favor specific part of the traffic (e.g. local traffic)

- **Routing Elements**
  - Performance optimization criteria
  - Decision time:
    - packet (datagram) or session (vc)
  - Decision place
    - Each node (distributed control)
    - Central node (centralized control)
    - Originating node (isolated control)
Routing Elements (cont.)

- **Status information source** - options:
  - no status information is used
  - local information
  - adjacent nodes
  - nodes along the route
  - all nodes

- **Routing strategy**
  - static routing
    - fixed
    - flooding
    - random
  - adaptive (dynamic) - update periods:
    - continuous
    - periodic
    - considerable load change
    - architecture updates (topology, throughput, etc.)
Static Routing

- **Features**
  - fixed routes: no use of status variables; DGs and VCs follow the same route
  - all static routes are maintained in Central Routing Directory (Routing Control Center)
  - local part of the routes are stored at each node directory (consists of destinations and next node to them)
  - simple, smallest overload, no flexibility
  - application for reliable networks with small variation of traffic parameters

- **Examples**
  - Generic fixed routing (see 18/2)
  - Shortest path routing
  - Flooding
  - Flow-based routing
  - Multipath random routing

*Single path static routing*
Shortest Path Routing

- Features
  - based on [weighted] graph representation of the subnet (node/router, arc/link)
  - shortest path is based on
    - the number of arcs between two routers (i.e. number of hops) - uses unmarked graph
    - the accumulated “length” of links (length is measured by the transmission time, transmission price, distance, etc.) - uses weighted graph (labeled arcs)
  - algorithms for shortest path from given node to ALL other:
    - Dijkstra’s algorithm - developing path in order of increasing path length; each node is labeled with the couple \((S_{i}, N_{i-1})\), iterations follow the shortest path to the next node
    - Bellman-Ford algorithm - iterative search for all the nodes that are distanced by 0, 1, 2, ... MAX hops
  - Dijkstra vs. Bellman-Ford
    - Computational Complexity: Dijkstra’s \(O(N^2)\) computations, Bellman-Ford \(O(N^3)\)
    - D’s algorithm requires knowledge of all costs - better suited to centralized routing decisions
    - BF’s only requires knowledge of link costs to neighboring nodes - may be implemented in a distributed way
    - Both can adapt to slowly changing link costs
Flooding

- **Features**
  - based on propagation of every incoming packet to each output line excluding the delivery line
  - modification: propagation is pointed to some “proper” subset of output lines - **selective flooding**

- **Effects:**
  - exponential growth of the number of packets
  - duplicate packets ☹ hop limitation
  - guaranteed finding of the shortest path
  - ☑ the flooding gives all the shortest paths in the graph

- **Application:**
  - trouble sensitive applications
  - concurrent updates in distributed databases
Flow-based Routing

- **Features**
  - considers both network topology (+ link capacity!) and some statistical steady-state data (e.g. average flow, mean delay) about the traffic between hops
  - the goal is minimization of the mean packet delay but not number of hops, kilometers, etc., i.e. minimization in time, not in space
  - requisites:
    - topology matrix
    - traffic matrix
    - links’ capacity matrix
    - some static routing
  - method: check of all [finite number] possible routes by calculation of mean residence time of the packets in the subnet

- **Effects and Application:**
  - best static single-path routing
  - applicable always when requisite set is available and the temporary status of the network does not deviate substantially from a steady-state
Multipath Random Routing

- **Features**
  - uses several paths when routing from source to destination
  - splits the outgoing traffic under control of a probability distribution function (PDF), which usually favors one/some of paths according some metrics ("shortest", less delay, bigger capacity, lower price, etc.)

  - requisites:
    - routing matrix for each node with one or more choice probabilities

- **Effects and Application:**
  - higher reliability, better congestion control and performance
  - applicable by networks with bigger traffic variances and when priority services have to be implemented
Dynamic [Adaptive] Routing

- Dynamic routing reflects the changes in traffic/network conditions (as stored in the status variables)

**Effects:**
- more complex decision phase of the algorithms
- additional communication overhead due to status information gathering [and exchange]: depends on range and responsiveness

**Information and control types** depending on status information gathering and connected decision making

- **isolated** (closer to the static routing)
  - local status information
  - independent local decision making
- **distributed** (best applicable for moderate communication load)
  - adjacent nodes status information (sometimes wider)
  - local decision making according common rules
- **centralized** (best applicable for heavy but stable load)
  - global status gathering (i.e. subnet range or internet range)
  - decision making by devoted node
Dynamic Routing Methods

- **Isolated routing**
  - “Hot-potato” algorithms (synonym JSQ - join shortest queue).
  - Extensions: table of associated biases to the routes (the biases act as delayed queue thresholds)
  - suitable for heavy/oscillating communication load at the routers

- **Distributed routing**
  - local decision based on the neighbor/all routers’ status
  - Example methods:
    - distance vector routing
    - link state routing

- **Centralized routing**
  - information and decision procedures are run by the network RCC (Routing Control Center), which updates local routing tables
  - quasi-optimal routing decisions and simplified routers cost communication overhead and fault/load-oscillation vulnerability
Distance Vector Routing

- Each router maintains “vector”-table of the distances to the rest of routers in its AS; distances are measured in # of hops, mean transmission time delay (measured by ECHO packets), local queue length to the remote router, monetary cost, etc. Synonym: dynamic Bellman-Ford routing.

- In first step distances are predefined (e.g. ∞), estimated or assumed to be known.

- Each router periodically (i.e. time-driven control!) initiates transfer of his vector table to the immediate neighbor routers only.

- Each recipient adds its own distance vector table and forwards it to its neighbors.

- The step-by-step process continues until each node update its distance vector table; i.e. information range is neighborhood; decision range is local.

- Applicable for smaller networks and the results are suboptimal at best but often problematical.
Distance Vector Convergence

• Distance vector routing (the information gathering procedure) is vulnerable during the convergence process by failures/changes in the network - inconsistent routing and infinite loops ("Count-to-infinity").

• Modifications - split horizon - forwarding to neighbors in unidirectional manner (the way back is reported infinity)

• Performance and implementations
  • status propagation in $D$ steps where $D$ is the “diameter” i.e. the longest distance in the connectivity graph
  • delay metric is distance but not the bandwidth
  • simple application but suitable for small networks with not more than few redundant paths
  • RIP
Link State Routing

- **Synonym SPF**: shortest path first
- **Decision range** is local and **information range** covers all the AS routers
- Distance metrics considers the link bandwidth
- Consists in development and maintaining of the extensive knowledge of the network routers via exchange of **LSA** (Link State Advertisement)
- Phases:
  - *link metrics* in neighborhood range (round trip ECHO packets with or without load consideration depending on stability requirements)
  - **AS wide propagation** of the results of *☆* in link state packets (**forward flooding** method controlled by the 16b sequence number of current packet and its age)
  - **local SPF computation** to any of AS routers:
    - reconstruction of the weighted net topology graph with averaged throughput arc labels according to the both ends’ states packets
    - Dijkstra’s SPF over the current topology
Link State and Hybrid Routing

- **Link State Routing Performance:**
  - **complex** link-, memory- and processor-intensive algorithm. Internal link data presentation is of $R^*L_i$ dimension (#routers by #links).
  - **performance degradation** during initial flooding of routing state packets, esp. in large networks with small bandwidth links
  - **event driven** control: convergence process begins and proceeds quickly after the change event without waiting timeouts; no overhead due to periodical routines
  - best **scalability** algorithm

- **Hybrid versions** of distributed protocols (Distance Vector and Link State) family of protocols that use extended version of DV metrics, event driven control like LS
  - implementations: Cisco firmware EIGRP (Enhanced Interior Gateway Routing Protocol)
  - performance: better information convergence than DV and better bulk overhead than LS
Hierarchical Routing

- Routing algorithms’ *scalability* limits the range of effective (or tolerably ineffective) routing
- Hierarchical approach divides the network in *regions* that are ranges of the network graph i.e. in related subsets of routers
- Each local routing table includes:
  - all the links to the routers in the region
  - one or few links to node[s] in the neighbor regions
- Multilevel hierarchy possible
- Iterregion routing may be distributed or centralized
- Hierarchy optimization (Kleinrock) for \( N \)-router network:
  - optimal # of levels \( L = \ln N \)
  - required # of table entries \( t = e \ln N = e \ L \)
  - for 5/17 example \( L = \ln 17 = 2.83, \ t = 7.701 \)
Broadcasting and Multicasting

- **Broadcast** routing methods:
  - 1. Separate packets to all destinations
  - 2. Flooding
  - 3. Multi- or omni-destination routing:
    - Packets contain enumeration of addresses or address mask over the multi-destination field
    - The router propagates such packet to each output line locally known as best route for some subset of the destinations; the destination enumeration/mask is reduced to the destinations reachable on that line
    - Splitting continues until each packet has single destination

- **Spanning tree** broadcasting: moderated flooding to all routers without loops. Applicable by LSR but not by DVR

- **Reverse path forwarding.** Topology information of 4 is replaced by keeping history track of routers’ *receiving* link with exclusive propagation or discarding.

- **Multicast Routing:**
  - **Groups’ spanning tree** method:
    - Each router maintains and propagate dynamically list of groups’ members among adjacent hosts
    - A pruned spanning tree is calculated for each group and the traffic is rerouted to and by the root node of the tree
TCP/IP Routing Protocols

- Internet routing concepts
  - Autonomous systems (AS)
  - internal and external routing - standardization and code reusability: interior and exterior gateway routing protocols

- Internet IGRPs
  - RIP (obsolete) - Bellman-Ford DVRP, time-driven: updates in 30 sec
  - OSPF (Open SPF) - multimetric, adaptive, LS method, event-driven:
    - service oriented (consider real-time and priority of the IP datagrams)
    - congestion control (load balancing) considerations
    - conservative: frequent alternative events are ignored!
    - support of hierarchical routing: splitting of AS in areas and backbones
    - security: routing table protection
    - tunneling and transparent service to the outside network requests

- OSPF application:
  - Point-to-point networks
  - multiaccess networks with broadcasting (LANs) and packet-switching
  - 2 modes of OSPF application:
    - intra-area routing
    - inter-area routing
    - backbone routers
    - inter-AS routing - interfacing some EGRP
TCP/IP Routing Protocols

• **OSPF (cont.):**
  
  • message exchange in local range (between adjacent designated routers - one in a LAN if more presented)

• **EGRP**s: Border Gateway Protocol (**BGP**)
  
  • extends the principle of hierarchical routing in Internet
  
  • applies different kind of routing considerations on inter-AS level
  
  • any pair of BGPs is interfaced via TCP connection by at least one common network that can be
    
    • **common multiconnected network** could be used for transit traffic, refuse
    
    • **devoted transit network** backbones, which are willing to handle third-party packets, possibly with some restrictions, and usually for pay
    
    • **stub networks** - have single connection to some BGP router (can’t transit traffic)
    
    • hybrid algorithm: applies DV information/propagation strategy but keeping and propagating the exact route to each destination.
<table>
<thead>
<tr>
<th>Upper layer</th>
<th>Type of subnet</th>
<th>Datagram</th>
<th>Virtual circuit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connectionless</td>
<td>UDP over IP</td>
<td>UDP over IP</td>
<td>UDP over IP over ATM</td>
</tr>
<tr>
<td>Connection-oriented</td>
<td>TCP over IP</td>
<td>ATM AAL1 over ATM</td>
<td></td>
</tr>
</tbody>
</table>
Resource conflict

Sink tree
Shortest Path Routing
<table>
<thead>
<tr>
<th>i</th>
<th>Line</th>
<th>$\lambda_i$ (pkts/sec)</th>
<th>$C_i$ (kbps)</th>
<th>$\mu C_i$ (pkts/sec)</th>
<th>$T_i$ (msec)</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AB</td>
<td>14</td>
<td>20</td>
<td>25</td>
<td>91</td>
<td>0.171</td>
</tr>
<tr>
<td>2</td>
<td>BC</td>
<td>12</td>
<td>20</td>
<td>25</td>
<td>77</td>
<td>0.146</td>
</tr>
<tr>
<td>3</td>
<td>CD</td>
<td>6</td>
<td>10</td>
<td>12.5</td>
<td>154</td>
<td>0.073</td>
</tr>
<tr>
<td>4</td>
<td>AE</td>
<td>11</td>
<td>20</td>
<td>25</td>
<td>71</td>
<td>0.134</td>
</tr>
<tr>
<td>5</td>
<td>EF</td>
<td>13</td>
<td>50</td>
<td>62.5</td>
<td>20</td>
<td>0.159</td>
</tr>
<tr>
<td>6</td>
<td>FD</td>
<td>8</td>
<td>10</td>
<td>12.5</td>
<td>222</td>
<td>0.098</td>
</tr>
<tr>
<td>7</td>
<td>BF</td>
<td>10</td>
<td>20</td>
<td>25</td>
<td>67</td>
<td>0.122</td>
</tr>
<tr>
<td>8</td>
<td>EC</td>
<td>8</td>
<td>20</td>
<td>25</td>
<td>59</td>
<td>0.098</td>
</tr>
</tbody>
</table>
### J $\uparrow$ G new route calculation

<table>
<thead>
<tr>
<th></th>
<th>J $\uparrow$ via</th>
<th>via $\uparrow$ G</th>
<th>comul.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>8</td>
<td>18</td>
<td>26</td>
</tr>
<tr>
<td>I</td>
<td>10</td>
<td>31</td>
<td>41</td>
</tr>
<tr>
<td>H</td>
<td>12</td>
<td>6</td>
<td>18</td>
</tr>
<tr>
<td>K</td>
<td>6</td>
<td>31</td>
<td>37</td>
</tr>
</tbody>
</table>

#### Figure (a)

A diagram showing a network of routers labeled A, B, C, D, E, F, G, H, I, J, K, and L. The diagram includes lines connecting these routers, indicating potential routes.

#### Figure (b)

A table showing the delays for each route. The table includes columns for the delays to A, I, H, and K, with corresponding delays for J's four neighbors. The delays are summarized at the bottom of the table, indicating the new estimated delay from J to each neighbor, and the new routing table for J.

- Vectors received from J's four neighbors:
  - JA delay: 8
  - JI delay: 10
  - JH delay: 12
  - JK delay: 6

- New routing table for J:
  - JA: 8
  - JI: 10
  - JH: 12
  - JK: 6
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>Inf</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>Inf</td>
</tr>
</tbody>
</table>

Initially

After 1 exchange

After 2 exchanges

After 3 exchanges

After 4 exchanges

(a)

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>Inf</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>Inf</td>
</tr>
</tbody>
</table>

Initially

After 1 exchange

After 2 exchanges

After 3 exchanges

After 4 exchanges

After 5 exchanges

After 6 exchanges

(b)
Routing probability matrix for node 4.

<table>
<thead>
<tr>
<th>Dest.</th>
<th>via</th>
<th>$q_{4i}$</th>
<th>bias</th>
<th>$q+b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>1.</td>
<td>$q_{41}$</td>
<td>0</td>
<td>$q_{41}+0$</td>
</tr>
<tr>
<td>2.</td>
<td>2.</td>
<td>$q_{42}$</td>
<td>0</td>
<td>$q_{42}+0$</td>
</tr>
<tr>
<td>3.</td>
<td>1.</td>
<td>$q_{43}$</td>
<td>2</td>
<td>$q_{43}+2$</td>
</tr>
<tr>
<td>4.</td>
<td>3.</td>
<td>$q_{45}$</td>
<td>2</td>
<td>$q_{45}+2$</td>
</tr>
<tr>
<td>5.</td>
<td>5.</td>
<td>$q_{43}$</td>
<td>0</td>
<td>$q_{43}+0$</td>
</tr>
<tr>
<td>6.</td>
<td>3.</td>
<td>$q_{45}$</td>
<td>2</td>
<td>$q_{45}+2$</td>
</tr>
<tr>
<td>7.</td>
<td>1.</td>
<td>$q_{41}$</td>
<td>5</td>
<td>$q_{43}+5$</td>
</tr>
</tbody>
</table>

↓ JSQ biased matrix for node 4.
Link states table
Hierarchical routing

(a)

Full table for 1A

<table>
<thead>
<tr>
<th>Dest.</th>
<th>Line</th>
<th>Hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1B</td>
<td>1B</td>
<td>1</td>
</tr>
<tr>
<td>1C</td>
<td>1C</td>
<td>1</td>
</tr>
<tr>
<td>2A</td>
<td>1B</td>
<td>2</td>
</tr>
<tr>
<td>2B</td>
<td>1B</td>
<td>3</td>
</tr>
<tr>
<td>2C</td>
<td>1B</td>
<td>3</td>
</tr>
<tr>
<td>2D</td>
<td>1B</td>
<td>4</td>
</tr>
<tr>
<td>3A</td>
<td>1C</td>
<td>3</td>
</tr>
<tr>
<td>3B</td>
<td>1C</td>
<td>2</td>
</tr>
<tr>
<td>4A</td>
<td>1C</td>
<td>3</td>
</tr>
<tr>
<td>4B</td>
<td>1C</td>
<td>4</td>
</tr>
<tr>
<td>4C</td>
<td>1C</td>
<td>4</td>
</tr>
<tr>
<td>5A</td>
<td>1C</td>
<td>4</td>
</tr>
<tr>
<td>5B</td>
<td>1C</td>
<td>5</td>
</tr>
<tr>
<td>5C</td>
<td>1B</td>
<td>5</td>
</tr>
<tr>
<td>5D</td>
<td>1C</td>
<td>6</td>
</tr>
<tr>
<td>5E</td>
<td>1C</td>
<td>5</td>
</tr>
</tbody>
</table>

Hierarchical table for 1A

<table>
<thead>
<tr>
<th>Dest.</th>
<th>Line</th>
<th>Hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1B</td>
<td>1B</td>
<td>1</td>
</tr>
<tr>
<td>1C</td>
<td>1C</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1B</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1C</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>1C</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>1C</td>
<td>4</td>
</tr>
</tbody>
</table>
AS 1

AS 2

AS 3

AS 4

Backbone

Backbone router

Area

BGP protocol connects the ASes

Internal router

Area border router

AS boundary router
(a) A set of BGP routers.  

(b) Information sent to F.

Information F receives from its neighbors about D:

- From B: "I use BCD"
- From G: "I use GCD"
- From I: "I use IFGCD"
- From E: "I use EFGCD"