The Network Layer: Control Plane

Raj Jain
Washington University in Saint Louis
Saint Louis, MO 63130
Jain@wustl.edu

Audio/Video recordings of this lecture are available on-line at:
http://www.cse.wustl.edu/~jain/cse473-16/

Network Layer Functions

- Forwarding: Deciding what to do with a packet using a routing table ⇒ Data plane
- Routing: Making the routing table ⇒ Control Plane

Routing Algorithms

1. Graph abstraction
2. Distance Vector vs. Link State
3. Dijkstra’s Algorithm
4. Bellman-Ford Algorithm

Note: This class lecture is based on Chapter 5 of the textbook (Kurose and Ross) and the figures provided by the authors.
Rooting or Routing

- **Rooting** is what fans do at football games, what pigs do for truffles under oak trees in the Vaucluse, and what nursery workers intent on propagation do to cuttings from plants.

- **Routing** is how one creates a beveled edge on a table top or sends a corps of infantrymen into full scale, disorganized retreat.


Routeing or Routing

- Routeing: British
- Routing: American

Since Oxford English Dictionary is much heavier than any other dictionary of American English, British English generally prevails in the documents produced by ISO and CCITT; wherefore, most of the international standards for routing standards use the routeing spelling.

Graph abstraction

- **Graph:** \( G = (N,E) \)
- **N** = Set of routers
  = \{ u, v, w, x, y, z \}
- **E** = Set of links
  = \{(u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z)\}
- Each link has a cost, e.g., \( c(w,z) = 5 \)
- Cost of path \( (x_1, x_2, \ldots, x_p) = c(x_1, x_2) + c(x_2, x_3) + \ldots + c(x_{p-1}, x_p) \)
- Routing Algorithms find the least cost path
- We limit to “Undirected” graphs, i.e., cost is same in both directions

Distance Vector vs. Link State

**Distance Vector:**

- Vector of distances to all nodes, e.g.,
  - \( u: \{ u:0, v:2, w:5, x:1, y:2, z:4 \} \)
- Sent to neighbors, e.g.,
  - \( u \) will send to \( v, w, x \)
- Large vectors to small \# of nodes
  - Tell about the world to neighbors
- Older method. Used in RIP.

**Link State:**

- Vector of link cost to neighbors, e.g, \( u: \{ v:2, w:5, x:1 \} \)
- Sent to all nodes, e.g., \( u \) will send to \( v, w, x, y, z \)
- Small vectors to large \# of nodes
  - Tell about the neighbors to the world
- Newer method. Used in OSPF.
**Dijkstra’s Algorithm**

- **Goal:** Find the least cost paths from a given node to all other nodes in the network.
- **Notation:**
  - \( c(i,j) = \text{Link cost from } i \text{ to } j \text{ if } i \text{ and } j \text{ are connected} \)
  - \( D(k) = \text{Total path cost from } s \text{ to } k \)
  - \( N' = \text{Set of nodes so far for which the least cost path is known} \)
- **Method:**
  1. **Initialize:** \( N' = \{u\} \), \( D(v) = c(u,v) \) for all neighbors of \( u \)
  2. **Repeat until \( N \) includes all nodes:**
     - Find node \( w \in N' \), whose \( D(w) \) is minimum
     - Add \( w \) to \( N' \)
     - Update \( D(v) \) for each neighbor of \( w \) that is not in \( N' \)
       \[ D(v) = \min[D(v), D(w) + c(w,v)] \]

---

**Dijkstra’s Algorithm: Example**

<table>
<thead>
<tr>
<th>( N' )</th>
<th>( D(v) )</th>
<th>Path</th>
<th>( D(w) )</th>
<th>Path</th>
<th>( D(x) )</th>
<th>Path</th>
<th>( D(y) )</th>
<th>Path</th>
<th>( D(z) )</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>{u}</td>
<td>2</td>
<td>u-v</td>
<td>5</td>
<td>u-w</td>
<td>1</td>
<td>u-x</td>
<td>5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>{u, x}</td>
<td>2</td>
<td>u-v</td>
<td>4</td>
<td>u-x-w</td>
<td>2</td>
<td>u-x-y</td>
<td>5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>{u, x, y}</td>
<td>2</td>
<td>u-v</td>
<td>3</td>
<td>u-x-y-w</td>
<td>4</td>
<td>u-x-y-z</td>
<td>5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>{u, x, y, v}</td>
<td>3</td>
<td>u-x-y-w</td>
<td>4</td>
<td>u-x-y-z</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>{u, x, y, v, w}</td>
<td>4</td>
<td>u-x-y-z</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

---

**Complexity and Oscillations**

- **Algorithm complexity:** \( n \) nodes
  1. Each iteration: need to check all nodes, \( w \), not in \( N \)
  2. \( n(n+1)/2 \) comparisons: \( O(n^2) \)
  3. More efficient implementations possible: \( O(n \log n) \)
- **Oscillations Possible:** e.g., support link cost equals amount of carried traffic

---

**Homework 5A**

Prepare the routing calculation **table** for node 1 in the following network using Dijkstra’s algorithm.
Bellman-Ford Algorithm

- Notation:
  - $u =$ Source node
  - $c(i,j) =$ link cost from $i$ to $j$
  - $h =$ Number of hops being considered
  - $D_u(n) =$ Cost of $h$-hop path from $u$ to $n$

- Method:
  1. Initialize: $D_u(n) = \infty$ for all $n \neq u$; $D_u(u) = 0$
  2. For each node: $D_u(n) = \min_j [D_u(j) + c(j,n)]$
  3. If any costs change, repeat step 2

Bellman Ford Example 1

<table>
<thead>
<tr>
<th>node x table</th>
<th>cost to</th>
<th>node y table</th>
<th>cost to</th>
<th>node z table</th>
<th>cost to</th>
</tr>
</thead>
<tbody>
<tr>
<td>from x to 0</td>
<td>0</td>
<td>from y to 2</td>
<td>7</td>
<td>from z to 1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td>1</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

Bellman-Ford Example 2

A. B. C. D.

Bellman-Ford: Tabular Method

If cost changes
⇒ Recompute the costs to all neighbors

<table>
<thead>
<tr>
<th>h</th>
<th>D(v) Path</th>
<th>D(w) Path</th>
<th>D(x) Path</th>
<th>D(y) Path</th>
<th>D(z) Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>u-v</td>
<td>u-w</td>
<td>u-x</td>
<td>u-x-y</td>
<td>u-w-z</td>
</tr>
<tr>
<td>2</td>
<td>u-v</td>
<td>u-x-w</td>
<td>u-x-z</td>
<td>u-x-y</td>
<td>u-x-y-z</td>
</tr>
<tr>
<td>3</td>
<td>u-v</td>
<td>u-x-y-w</td>
<td>u-x-y-z</td>
<td>u-x-y-z</td>
<td>u-x-y-z</td>
</tr>
</tbody>
</table>
Counting to Infinity Problem

R1 loses A

R1 hears from R2 → (Before it tells R2)

Routing Algorithms: Summary

1. Distance Vectors: Distance to all nodes in the network sent to neighbors. Small # of large messages.
2. Link State: Cost of link to neighbors sent to entire network. Large # of small messages.
3. Dijkstra’s algorithm is used to compute shortest path using link state
4. Bellman Ford’s algorithm is used to compute shortest paths using distance vectors
5. Distance Vector algorithms suffer from the count-to-infinity problem

Ref: Read Section 5.2 of the textbook and try review questions R3-R6.

Homework 5B

Prepare the routing calculation table for node 1 in the following network using the Bellman-Ford Algorithm.

Routing Protocols

1. Autonomous Systems (AS)
2. Open Shortest Path First (OSPF)
   - OSPF Areas
3. Border Gateway Protocol (BGP)
Autonomous Systems

- An internet connected by homogeneous routers under the administrative control of a single entity

Routing Protocols

- Interior Router Protocol (IRP): Used for passing routing information among routers internal to an autonomous system. Also known as IGP.
  - Examples: RIP, OSPF, IGRP
- Exterior Router Protocol (ERP): Used for passing routing information among routers between autonomous systems. Also known as EGP.
  - Examples: EGP, BGP, IDRP
  
  Note: EGP is a class as well as an instance in that class.

Open Shortest Path First (OSPF)

- Uses true metrics (not just hop count)
- Uses subnet masks
- Allows load balancing across equal-cost paths
- Supports type of service (ToS)
- Allows external routes (routes learnt from other autonomous systems)
- Authenticates route exchanges
- Quick convergence
- Direct support for multicast
- Link state routing  Each router broadcasts its connectivity with neighbors to entire network

OSPF Areas

- Large networks are divided into areas to reduce routing traffic.
- LSAs are flooded throughout the area
- Area border routers (ABRs) summarize the topology and transmit it to the backbone area
- Backbone routers forward it to other areas
- ABRs connect an area with the backbone area. ABRs contain OSPF data for two areas. ABRs run OSPF algorithms for the two areas.
- If there is only one area in the AS, there is no backbone area and there are no ABRs.
Border Gateway Protocol

- Inter-autonomous system protocol [RFC 1267]
- Used since 1989 but not extensively until recently
- Runs on TCP (segmentation, reliable transmission)
- Advertises all transit ASs on the path to a destination address
- A router may receive multiple paths to a destination \( \Rightarrow \) Can choose the best path
- iBGP used to forward paths inside the AS.
- eBGP used to exchange paths between ASs.

Ref: Section 5.4 of the textbook.

BGP Operations

- BGP systems initially exchange entire routing tables. Afterwards, only updates are exchanged.
- BGP messages have the following information:
  - Origin of path information: RIP, OSPF, …
  - AS Path: List of ASs on the path to reach the dest
  - Next_Hop: IP address of the border router to be used as the next hop to reach the dest
  - Unreachable: If a previously advertised route has become unreachable
- BGP speakers generate update messages to all peers when it selects a new route or some route becomes unreachable.

BGP Routing Policy Example

- A, B, C are provider networks
- X, W, Y are customer (of provider networks)
- X is dual-homed: attached to two networks
  - X does not want to route from B via X to C
  - .. so X will not advertise to B a route to C

A advertises path A-W to B
B advertises path B-A-W to X
Should B advertise path B-A-W to C?
  - No way! B gets no “revenue” for routing C-B-A-W since neither W nor C are B’s customers
  - B wants to force C to route to w via A
  - B wants to route only to/from its customers!
Intra- vs. Inter-AS Routing

- **Policy:**
  - Inter-AS: admin wants control over how its traffic routed, who routes through its net.
  - Intra-AS: single admin, so no policy decisions needed

- **Scale:**
  - Hierarchical routing saves table size, reduced update traffic

- **Performance:**
  - Intra-AS: can focus on performance
  - Inter-AS: policy may dominate over performance

Routing Protocols: Summary

1. OSPF uses link-state routing and divides the autonomous systems into multiple areas. Area border router, AS boundary router, designated router
2. BGP is an inter-AS protocol ⇒ Policy driven

SDN Control Plane

1. S1, experiencing link failure using OpenFlow port status message to notify controller
2. SDN controller receives OpenFlow message, updates link status info
3. Dijkstra’s routing algorithm application has previously registered to be called when ever link status changes. It is called.
4. Dijkstra’s routing algorithm access network graph info, link state info in controller, computes new routes

Controller Example: OpenDaylight
OpenDaylight SDN Controller

- Multi-company collaboration under Linux foundation
- Many projects including OpenDaylight Controller
- Dynamically linked in to a Service Abstraction Layer (SAL) ⇒ SAL figures out how to fulfill the service requested by higher layers irrespective of the southbound protocol
- Modular design
- A rich set of North-bound APIs via RESTful (Web page like) services

Ref: Read Section 5.5 and try review questions R14-R18.

ICMP

- Internet Control Message Protocol
- Required companion to IP. Provides feedback from the network.
- ICMP: Used by IP to send error and control messages
- ICMP uses IP to send its messages (Not UDP)
- ICMP does not report errors on ICMP messages.
- ICMP reports error only on the first fragment

ICMP Header

ICMP Data

IP Header

IP Data

Datalink Header

Datalink Data

ICMP: Message Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Echo reply</td>
</tr>
<tr>
<td>3</td>
<td>Destination unreachable</td>
</tr>
<tr>
<td>4</td>
<td>Source quench</td>
</tr>
<tr>
<td>5</td>
<td>Redirect</td>
</tr>
<tr>
<td>8</td>
<td>Echo request</td>
</tr>
<tr>
<td>11</td>
<td>Time exceeded</td>
</tr>
<tr>
<td>12</td>
<td>Parameter unintelligible</td>
</tr>
<tr>
<td>13</td>
<td>Time-stamp request</td>
</tr>
<tr>
<td>14</td>
<td>Time-stamp reply</td>
</tr>
<tr>
<td>15</td>
<td>Information request</td>
</tr>
<tr>
<td>16</td>
<td>Information reply</td>
</tr>
<tr>
<td>17</td>
<td>Address mask request</td>
</tr>
<tr>
<td>18</td>
<td>Address mask reply</td>
</tr>
</tbody>
</table>

ICMP Messages

- Source Quench: Please slow down! I just dropped one of your datagrams.
- Time Exceeded: Time to live field in one of your packets became zero,” or “Reassembly timer expired at the destination.
- Fragmentation Required: Datagram was longer than MTU and “No Fragment bit” was set.
- Address Mask Request/Reply: What is the subnet mask on this net? Replied by “Address mask agent”
- PING uses ICMP echo
- Tracert uses TTL expired

Ref: Read Section 5.6 of the textbook and try review questions R19-R20.
Trace Route Example

C:\>tracert www.google.com

Tracing route to www.l.google.com [74.125.93.147] over a maximum of 30 hops:
1  3 ms  1 ms  1 ms  192.168.0.1
2  12 ms 10 ms  9 ms  bras4-10.stlsmo.sbcglobal.net [151.164.182.113]
3  10 ms  8 ms  8 ms  dist2-vlan60.stlsmo.sbcglobal.net [151.164.14.163]
4  9 ms  7 ms  7 ms  151.164.93.224
5  25 ms 22 ms  22 ms  151.164.251.226
6  30 ms 28 ms  28 ms  209.85.254.128
7  61 ms 57 ms  58 ms  72.14.236.26
8  54 ms 52 ms  51 ms  209.85.254.226
9  79 ms 60 ms  60 ms  72.14.236.26
10 79 ms 60 ms  60 ms  72.14.236.26
11 66 ms 57 ms  58 ms  64.233.175.14
12 60 ms 58 ms  58 ms  qw-in-f147.google.com [74.125.93.147]

Trace complete.

Lab 5A: ICMP

- Download the Wireshark traces from http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip
- Open icmp-ethereal-trace-1 in Wireshark.
  - Select View → Expand All. Answer the following questions:
    1. Examine Frame 3.
       A. What is the IP address of your host? What is the IP address of the destination host?
       B. Why is it that an ICMP packet does not have source and destination port numbers?
       C. What are the ICMP type and code numbers? What other fields does this ICMP packet have? How many bytes are the checksum, sequence number and identifier fields?

Lab 5A (Cont)

2. Examine Frame 4. What are the ICMP type and code numbers?

- Open icmp-ethereal-trace-2 in Wireshark.
  - Answer the following questions:

3. Examine Frame 2. What fields are included in this ICMP error packet?

4. Examine Frames 100, 101, and 102. How are these packets different from the ICMP error packet 2? Why are they not error packets?

Network Management

- What is Network Management?
- Components of Network Management
- How is Network Managed?
- SNMP protocol
What is Network Management?

- Traffic on Network = Data + Control + Management
  - **Data** = Bytes/Messages sent by users
  - **Control** = Bytes/messages added by the system to properly transfer the data (e.g., routing messages)
  - **Management** = Optional messages to ensure that the network functions properly and to handle the issues arising from malfunction of any component

If all components function properly, control is still required but management is optional.

Examples:
- Detecting failures of an interface card at a host or a router
- Monitoring traffic to aid in resource deployment
- Intrusion Detection

Components of Network Management

1. **Fault Management**: Detect, log, and respond to fault conditions
2. **Configuration Management**: Track and control which devices are on or off
3. **Accounting Management**: Monitor resource usage for records and billing
4. **Performance Management**: Measure, report, analyze, and control traffic, messages
5. **Security Management**: Enforce policy for access control, authentication, and authorization

FCAPS

How is Network Managed?

- Management = Initialization, Monitoring, Control
- Manager, Agents, and Management Information Base (MIB)

A MIB is defined for each device
SNMP

- Based on Simple Gateway Management Protocol (SGMP) – RFC 1028 – Nov 1987
- SNMP = Simply Not My Problem [Marshall Rose]
  
  *Simple Network Management Protocol*
- RFC 1058, April 1988
- Only Five commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>get-request</td>
<td>Fetch a value</td>
</tr>
<tr>
<td>get-next-request</td>
<td>Fetch the next value (in a tree)</td>
</tr>
<tr>
<td>get-response</td>
<td>Reply to a fetch operation</td>
</tr>
<tr>
<td>set-request</td>
<td>Store a value</td>
</tr>
<tr>
<td>trap</td>
<td>An event</td>
</tr>
</tbody>
</table>

SNMP protocol

Two ways to convey MIB info, commands:

- Request/response mode
- Trap mode

SNMP Message Formats

| PDU type (0-3) | Request ID | Error Status (0-5) | Error Index | Name | Value | Name | Value | ...
|---------------|------------|--------------------|-------------|------|-------|------|-------|----|

| PDU type 4 | Enterprise | Agent Addr | Trap Type (0-7) | Specific code | Time stamp | Name | Value | ...
|------------|------------|------------|----------------|---------------|------------|------|-------|----|

Network Management: Summary

1. Management = Initialization, Monitoring, and Control
2. Standard MIBs defined for each object
3. SNMP = Only 5 commands in the first version
Network Layer Control Plane: Summary

1. Dijkstra’s algorithm allows path computation using link state.
2. Bellman Ford’s algorithm allows path computation using distance vectors.
3. OSPF is a link state IGP.
4. BGP is an EGP and uses path vectors.
5. SDN controllers use various algorithms for centralized computation of path and other policies.
6. ICMP is IP control protocol is used to convey errors.
7. SNMP is the simple network management protocol to manage all devices and protocols in a network.

Acronyms

- ABR: Area border router
- API: Application Programming Interface
- AS: Autonomous System
- ASBR: Autonomous System Boundary Router
- BDR: Backup Designated Router
- BGP: Border Gateway Protocol
- BR: Backbone Router
- CAPWAP: Control and Provisioning of Wireless Access Points
- CCITT: Consultative Committee for International Telegraph and Telephone (now ITU-T)
- CoAP: Constrained Application Protocol
- COPS: Common Open Policy Service
- DIDM: Device Identifier and Driver Management
- DLUX: OpenDaylight User Interface
- DOCSIS: Data over Cable Service Interface Specification
- DR: Designated Router
- eBGP: exterior BGP
- EGP: Exterial Gateway Protocol
- ERP: Exterior Router Protocol
- FCAPS: Fault Configuration Accounting Performance and Security
- FRM: Forwarding Rules Manager
- GBP: Group Based Policy
- GUI: Graphical User Interface
- HTTP: Hyper-Text Transfer Protocol
- iBGP: interior BGP
- ICMP: IP Control Message Protocol
- ID: Identifier
- IDRP: ICMP Router Discovery Protocol
- IGP: Interior Gateway Protocol
- IGRP: Interior Gateway Routing Protocol
- IP: Internet Protocol
- IRP: Interior Router Protocol
- ISO: International Standards Organization

Acronyms (Cont)

- LACP: Link Aggregation Control Protocol
- LSA: Link State Advertisements
- MIB: Management Information Base
- MTU: Maximum Transmission Unit
- NETCONF: Network Configuration Protocol
- NIC: Network Interface Card
- OSGi: Open Service Gateway Initiative
- OSI: Open Service Interconnection
- OSPF: Open Shortest Path First
- OVSDB: Open Vswitch Database
- PCEP: Path Computation Element Protocol
- PCMM: Packet Cable Multimedia
- REST: Representational State Transfer
- RESTful: Representational State Transfer
- RFC: Request for Comments
- SAL: Service Abstraction Layer
### Acronyms (Cont)

- **SDN**  
  Software Defined Networking
- **SDNI**  
  SDN domains interface
- **SFC**  
  Service Function Chaining
- **SFC**  
  Service Function Chaining
- **SGMP**  
  Simple Gateway Management Protocol
- **SNBI**  
  Secure Network Bootstrapping Interface
- **SNMP**  
  Simple Network Management Protocol
- **SXP**  
  SGT (Security Group Tags) Exchange Protocol
- **TCP**  
  Transmission Control Protocol
- **ToS**  
  Type of Service
- **TSDR**  
  Time Series Data Repository
- **TTL**  
  Time to Live
- **UDP**  
  User Datagram Protocol
- **USC**  
  Unified Secure Channel
- **VPN**  
  Virtual Private Network
- **VTN**  
  Virtual Tenant Network

### Related Modules

- **CSE 473s: Introduction to Computer Networks**  
  (Course Overview),  
  [http://www.cse.wustl.edu/~jain/cse473-16/ftp/i_0int.pdf](http://www.cse.wustl.edu/~jain/cse473-16/ftp/i_0int.pdf)

- **CSE473S: Introduction to Computer Networks (Fall 2016)**,  

- **Wireless and Mobile Networking (Spring 2016)**,  
  [http://www.cse.wustl.edu/~jain/cse574-16/index.html](http://www.cse.wustl.edu/~jain/cse574-16/index.html)

- **CSE571S: Network Security (Fall 2014)**,  
  [http://www.cse.wustl.edu/~jain/cse571-14/index.html](http://www.cse.wustl.edu/~jain/cse571-14/index.html)

- **Audio/Video Recordings and Podcasts of Professor Raj Jain's Lectures**,  
  [https://www.youtube.com/channel/UCN4-5wzNP9-ruOzQMs-8NUw](https://www.youtube.com/channel/UCN4-5wzNP9-ruOzQMs-8NUw)