CSE 422 Notes, Set 4

- These slides contain materials provided with the text: Computer Networking: A Top Down Approach, various editions, by Jim Kurose and Keith Ross, Addison-Wesley.
- Additional figures are repeated, with permission, from Computer Networks, 2nd through 4th Editions, by A. S. Tanenbaum, Prentice Hall.
- The remainder of the materials were developed by Philip McKinley at Michigan State University.

Chapters 4,5: Network Layer

- Introduction (forwarding and routing)
- Review of queueing theory
- Router design and operation
- IP: Internet Protocol
  - IPv4 (datagram format, addressing, NAT, ICMP)
  - IPv6
- Routing algorithms
  - Link state, Distance Vector
- Routing in the Internet
  - Autonomous Systems
  - Routing protocols (OSPF, BGP)
- Generalized Forwarding & SDN
Hierarchical Routing

Our routing study thus far has focused on individual networks
- network is “flat”
- all routers are identical

But, the Internet is a network of networks
- Routing is hierarchical. Why?
- Individual networks have autonomy.

Hierarchical Routing

- Regions of the Internet, managed by different organizations, are referred to as “autonomous systems” (AS)
- Routers in same AS run same routing protocol
  - “intra-AS” routing protocol
  - routers in a different AS can run a different intra-AS routing protocol

Gateway router

- Has a direct link to a router in another AS
- Gateway routers perform “inter-AS” routing (as well as intra-AS routing within their own AS)
**Autonomous System**

- Collection of network prefixes
  - E.g., 207.73.136/22, 132.111.96/19
- An ISP might own and manage multiple ASes
- Each AS has an **Autonomous System Number (ASN)**
- ASNs are used to identify elements of paths across the Internet

**Interconnected Autonomous Systems**

- A router’s forwarding table is configured by both intra- and inter-AS routing algorithm
  - intra-AS protocol sets entries for internal destinations
  - inter-AS & intra-AS routing determines entries for external destinations
FYI: ASN and MSU

- MSU networking is part of Merit, non-profit started in 1960s for sharing of mainframe computer cycles among MSU, UM, WSU.
- Merit is longest running regional network in U.S.
- MSU CSE is within AS237 (one of several operated by Merit), which includes:
  - 35.8.0.0/13
- If interested, information on ASNs, prefixes, connectivity is available from various sources:

Intra-AS Routing

- also known as Interior Gateway Protocols (IGP)
- most common Intra-AS routing protocols:
  - RIP: Routing Information Protocol (DV)
  - OSPF: Open Shortest Path First* (LS)
  - IGRP: Interior Gateway Routing Protocol (DV) (Cisco proprietary until 2016)
OSPF (Open Shortest Path First)

- OPEN means publicly available, not proprietary
- uses Link State algorithm
  - LS packet dissemination
  - topology map at each node
  - route computation using Dijkstra’s algorithm
- Link-state advertisements disseminated to entire AS
  - via flooding
  - carried in OSPF messages directly over IP, rather than TCP or UDP

OSPF “advanced” features (not in RIP)

- security: all OSPF messages authenticated (to prevent malicious intrusion)
- multiple same-cost paths allowed
- For each link, multiple cost metrics for different types of service (e.g., satellite link cost set “low” for best effort; high for real time)
- integrated unicast and multicast support:
  - Multicast OSPF (MOSPF) uses same topology database as OSPF
- hierarchical OSPF in large domains.
Hierarchical OSPF

- two-level hierarchy: local area, backbone.
  - Link-state advertisements only in area
  - each node has detailed area topology; only know direction (shortest path) to nets in other areas.
- area border routers: “summarize” distances to nets in own area, advertise to other Area Border routers.
- backbone routers: run OSPF routing limited to backbone.
- boundary routers: connect to other AS’ s.
Inter-AS routing: BGP

- **BGP (Border Gateway Protocol):** the de facto standard
- **BGP** provides each AS a means to:
  1. Obtain subnet reachability information from neighboring autonomous systems
  2. Propagate reachability information to all AS-internal routers.
  3. Determine “good” routes to subnets based on reachability information and policy.
- Allows subnet (prefix) to advertise its existence to rest of Internet: “I am here”
- Resembles distance-vector, but based on paths
- Advantage?

---

**eBGP, iBGP connections**

- BGP routers (“peers”) exchange BGP messages over semi-permanent **TCP (!) connections** (on port 179):
  - exterior to other AS (eBGP), interior within an AS (iBGP)
  - “keep-alive” messages sent every 60 seconds

---

[Diagram showing BGP connectivity between AS 1, AS 2, and AS 3.]

gateway routers run both eBGP and iBGP protocols
BGP Evolution

- BGP-4 has been running since mid-1990s
  - Based on CIDR prefixes
- Revised over the years
- Current version of BGP-4:
  - Defined in 2006 (RFC 4271)

Path attributes and BGP routes

- advertised prefix includes BGP attributes
  - prefix + attributes = “route”
- two important attributes:
  - AS-PATH: list of ASes through which prefix advertisement has passed (list of ASNs, not prefixes)
  - NEXT-HOP: indicates specific internal-AS router to next-hop AS
- Policy-based routing:
  - gateway receiving route advertisement uses import policy to accept/decline path (e.g., never route through AS Y).
  - AS policy also determines whether to advertise path to other other neighboring ASes
**BGP path advertisement**

- **BGP session**: BGP routers advertise *paths* to different destination network prefixes (i.e., BGP is a *path* vector protocol, rather than distance vector)

- When AS3 gateway router 3a advertises path AS3,X to AS2 gateway router 2c:
  - AS3 *promises* to AS2 it will forward datagrams towards X

![Diagram showing BGP path advertisement](image)

**BGP path advertisement**

- AS2 router 2c receives path advertisement AS3,X (via eBGP) from AS3 router 3a

- Based on AS2 policy, AS2 router 2c accepts path AS3,X, propagates (via iBGP) to all AS2 routers

- Based on AS2 policy, AS2 router 2a advertises (via eBGP) path AS2, AS3, X to AS1 router 1c

![Diagram showing BGP path advertisement](image)
Example path advertisement

Hot Potato Routing

Basically, get packets out of autonomous system ASAP.

- 2d learns (via iBGP) it can route to X via 2a or 2c
- *Hot potato routing*: choose local gateway that has least intra-domain cost (e.g., 2d chooses 2a, even though more AS hops to X): don’t worry about inter-domain cost!
**BGP path advertisement**

A gateway router may learn about multiple paths to destination:

- AS1 gateway router 1c learns path AS2,AS3,X from 2a
  - AS1 gateway router 1c learns path AS3,X from 3a
  - AS1 gateway router 1c chooses path AS3,X, and advertises path within AS1 via iBGP

**BGP route selection**

Route is a list of ASNs, not list of prefixes

If a router learns about more than one route to destination AS, it stores all the routes but selects the one to use based on:

1. Select path with highest local **preference** value attribute, i.e., a policy decision
2. If ties in (1), select shortest AS-PATH
3. If ties in (2), select closest NEXT-HOP router (hot potato routing)
4. If ties in (3), use additional criteria such as a tie breaker (e.g., BGP identifier)
Achieving policy via advertisements

Suppose an ISP only wants to route traffic to/from its customer networks (does not want to carry transit traffic between other ISPs)

- A advertises path Aw to B and to C
- B *chooses not to advertise* BAw to C:
  - B gets no “revenue” for routing CBAw, since none of C, A, w are B’s customers
  - So, C does not learn about CBAw path
- Hence, C will route packets to w on CAw (not using B)

---

Achieving policy via advertisements

Suppose an ISP only wants to route traffic to/from its customer networks (does not want to carry transit traffic between other ISPs)

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

- A remote controller computes routes and sends to every router
- Was used in some very early networks
- Problems?

Distributed Routing

- As we have seen, modern routing algorithms are distributed
  - OSPF (link-state)
  - RIP (distance vector)
  - BGP (“path vector”)
- Routers exchange state info and compute routes, more robust than centralized routing
- Results are entered into routing tables
- Hence, two functions of the network layer, routing and forwarding, are tightly coupled.
Data Plane vs. Control Plane

Starting in the early 2000s, researchers and developers began to explore decoupling:

**Data plane**
- local, per-router function
- determines how datagram arriving on router input port is **forwarded** to router output port
- forwarding function

**Control plane**
- network-wide logic
- determines how datagram is **routed** among routers along end-end path from source host to destination host

Traditional Routing

- Routing algorithm components **in each and every router** interact with each other in control plane to compute forwarding tables
Software Defined Networking

- Although interest started around 2004, only gained traction after 2011 or so.
- Driving forces for new network architecture:
  - Data centers: lots of “east-west” traffic in satisfying user requests (as opposed to mainly client-server patterns)
  - Mobile device access to corporate networks
  - Cloud services and virtualization
  - Big Data and hyper-scale processing, networking
- Remote “logically centralized” routing facilitates more control/adaptation in routing, to meet specific needs of an organization

Illustration

- A distinct (typically remote) controller interacts with local control agents (CAs) in routers to compute forwarding tables

Diagram showing a network with remote controller interacting with local control agents (CAs) in routers.
SDN Elements

- Generalized forwarding of packets
  - a.k.a. flow-based forwarding
  - use multiple header fields in deciding packet action
- Separation of data plane (forwarding) and control plane (routing)
- Network control software external to switches (distributed but logically centralized from switch perspective)
- Programmable network via remote applications for routing, access control, load balancing, etc.

Illustration

1. generalized "flow-based" forwarding (e.g., OpenFlow)
2. control, data plane separation
3. control plane functions external to data-plane switches
4. programmable control applications
Generalized Forwarding

- Traditional packet forwarding based solely on destination IP address
- Recent exceptions:
  - NAT - rewrite IP address and port numbers
  - Firewalls: block packets based on IP addresses, port numbers, track TCP connections, etc.
  - Load balancers: reroute packets among multiple devices that collectively provide a service
- Generalized forwarding expands and formalizes this “match-plus-action” behavior

Packet Switches and OpenFlow

- Devices that perform generalized forwarding are referred to as packet switches
  - As opposed to layer 3 routers and layer 2 switches
- OpenFlow standard: generalized forwarding
  - Switch has flow table of match-plus-action entries
  - Each entry
    - Set of header field values (fast matching hardware in switch)
    - Set of counters for various matches, etc.
    - Set of actions (forward, drop, modify field, etc.)
  - In OpenFlow 1.0, 11 fields can be involved in matching
    - Can include wildcards (e.g., 129.119.*.*)
    - Entries are prioritized, highest priority matching entry rules
OpenFlow Entries

<table>
<thead>
<tr>
<th>Rule</th>
<th>Action</th>
<th>Stats</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Packet + byte counters</td>
</tr>
</tbody>
</table>

1. Forward packet to port(s)
2. Encapsulate and forward to controller
3. Drop packet
4. Send to normal processing pipeline
5. Modify Fields

Applications of match+action

- **Router** (layer 3)
  - **match**: longest destination IP prefix
  - **action**: forward out a link
- **Switch** (layer 2)
  - **match**: destination MAC address
  - **action**: forward or flood
- **Firewall**
  - **match**: IP addresses and TCP/UDP port numbers
  - **action**: permit or deny
- **NAT**
  - **match**: IP address and port
  - **action**: rewrite address and port
OpenFlow Communication

- OpenFlow standard also defines communication between “controller” and packet switches
- Controller-to-switch, e.g.,
  - Get/set switch configuration params
  - Add/delete/modify flow table entries
  - Send enclosed packet on specific port
- Switch-to-controller, e.g.,
  - Send packet to controller (e.g., unmatched)
  - Port status has changed
  - Etc.

SDN Controller Components

- Interface layer to network control apps: abstractions
  - API
- Network-wide state management layer:
  - state of networks links, switches, services: a distributed database
  - OpenFlow
- Communication layer:
  - communicate between SDN controller and controlled switches
  - SNMP

Network Layer
**SDN Controller**

- OpenFlow is almost universally used to realize the "southbound" API between controller and switches
- State management layer elements are distributed and likely contain full copies of flow tables and other state
- Northbound API
  - Typically applications register to be notified when state changes occur
  - Can actually contain multiple APIs

**SDN Routing Example**

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
Network Layer Summary

- Introduction (forwarding and routing)
- Review of queueing theory
- Router design and operation
- IP: Internet Protocol
  - IPv4 (datagram format, addressing, NAT, ICMP)
  - IPv6 (maybe premature considering CIDR, NAT, SDN?)
- Routing algorithms
  - Link state, Distance Vector
- Routing in the Internet
  - Autonomous Systems
  - Routing protocols (OSPF, BGP)
- Generalized Forwarding & SDN
- Takeaways?