Today’s Lecture Objectives

- Network Layer
  - Routing

- Small note going before the lecture
  - Lots more slides today, there mostly to help you review
Routing vs. Forwarding

• Forwarding
  – Decided once a packet arrives
  – Given a packet’s destination, which outgoing interface gets it closer to its destination
  – Uses the routing table (should probably be called the forwarding table)

• Routing
  – The process of discovering neighbors and gathering information about in which direction destinations lie

• THE Question
  – On what outgoing interface should a packet be sent to get it closer to its destination
  – Not an end-to-end decision: the tool is always the routing table
Routing vs. Forwarding

Routing algorithm

<table>
<thead>
<tr>
<th>header value</th>
<th>output link</th>
</tr>
</thead>
<tbody>
<tr>
<td>0100</td>
<td>3</td>
</tr>
<tr>
<td>0101</td>
<td>2</td>
</tr>
<tr>
<td>0111</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1</td>
</tr>
</tbody>
</table>

value in arriving packet's header
Routing Algorithm Classification

Centralized, Global or Decentralized?

Centralized:
• One central node gathers information, makes decisions, and informs routers

Global:
• All routers have complete topology, link cost info

Decentralized:
• Router knows physically-connected neighbors, link costs to neighbors
• Iterative process of computation, exchange of info with neighbors

Static or Dynamic?

Static:
• Routes are set and not changed
• Works well when routes change slowly over time

Dynamic:
• Routes are set and then periodically updated
• Works well when routes change more quickly
  – periodic update
  – in response to link cost changes
Routing Basics

• THE Question
  – On what outgoing interface should a packet be sent to get it closer to its destination
  – Not an end-to-end decision: the tool is always the routing table

• Basic Process
  – Router interface is configured for a range of addresses
  – Routers discover neighbors
  – Routers start advertising “local prefixes” to the world
**Distance Vector**

1. Discover neighbors
2. Build routing table
3. Advertise to neighbors
4. Receive advertisements
5. Run Bellman-Ford algorithm—update table

**Link State**

1. Discover neighbors
2. Build routing table
3. Flood through network
4. Receive advertisements
5. Run Dijkstra’s algorithm—update table
Graph Abstraction

- Useful for understanding Bellman-Ford and Dijkstra

- Graph: \( G = (N,E) \)
- \( N \) = set of routers = \{ u, v, w, x, y, z \}
  - “routers” translate into real-world address ranges
- \( E \) = set of links =\{ (u,v),(u,x),(v,x),(v,w),(x,w),(x,y),(w,y),(w,z),(y,z) \}
Examples

• Bellman-Ford

• Dijkstra
Distance Vector Algorithm: Bellman-Ford

Iterative, asynchronous:
• Each local iteration caused by:
  – local link cost change
  – DV update message from neighbor

Distributed:
• Each node notifies neighbors only when its DV changes
  – neighbors then notify their neighbors if necessary

Each node:
1. wait for (change in local link cost of msg from neighbor)
2. recompute estimates
3. if DV to any dest has changed, notify neighbors
### node x table

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>from</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>0</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>y</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>z</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

### node y table

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>from</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>y</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>z</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

### node z table

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>from</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>y</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>z</td>
<td>7</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

---

**Diagram:**

A network diagram showing nodes x, y, and z with connections and costs. The diagram includes arrows indicating the direction of flow or cost from one node to another. The costs are indicated in the table above the diagram.
Link-State Routing Algorithm: Dijkstra

Dijkstra’s algorithm

- net topology, link costs known to all nodes
  - accomplished via “link state broadcast”
  - all nodes have same info
- computes least cost paths from one node (‘source”) to all other nodes
  - gives forwarding table for that node
- iterative: after k iterations, know least cost path to k dest.’s

Notation:

- $c(x,y)$: link cost from node $x$ to $y$; $= \infty$ if not direct neighbors
- $D(v)$: current value of cost of path from source to dest. $v$
- $p(v)$: predecessor node along path from source to $v$
- $N'$: set of nodes whose least cost path definitively known
Dijsktra’s Algorithm

1 *Initialization:*
2 \( N' = \{u\} \)
3 for all nodes \( v \)
4 if \( v \) adjacent to \( u \)
5 then \( D(v) = c(u,v) \)
6 else \( D(v) = \infty \)
7
8 *Loop*
9 find \( w \) not in \( N' \) such that \( D(w) \) is a minimum
10 add \( w \) to \( N' \)
11 update \( D(v) \) for all \( v \) adjacent to \( w \) and not in \( N' \) :
12 \[ D(v) = \min( D(v), D(w) + c(w,v) ) \]
13 /* new cost to \( v \) is either old cost to \( v \) or known shortest path cost to \( w \) plus cost from \( w \) to \( v \) */
14
15 *until all nodes in \( N' \)*
### Example of Dijkstra’s Algorithm

<table>
<thead>
<tr>
<th>Step</th>
<th>N'</th>
<th>D(v),p(v)</th>
<th>D(w),p(w)</th>
<th>D(x),p(x)</th>
<th>D(y),p(y)</th>
<th>D(z),p(z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>u</td>
<td>2,u</td>
<td>5,u</td>
<td>1,u</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>1</td>
<td>ux</td>
<td>2,u</td>
<td>4,x</td>
<td>2,x</td>
<td>∞</td>
<td>4,y</td>
</tr>
<tr>
<td>2</td>
<td>uxy</td>
<td>2,u</td>
<td>3,y</td>
<td>4,y</td>
<td>4,y</td>
<td>4,y</td>
</tr>
<tr>
<td>3</td>
<td>uxyv</td>
<td>3,y</td>
<td>4,y</td>
<td>4,y</td>
<td>4,y</td>
<td>4,y</td>
</tr>
<tr>
<td>4</td>
<td>uxyvw</td>
<td>3,y</td>
<td>4,y</td>
<td>4,y</td>
<td>4,y</td>
<td>4,y</td>
</tr>
<tr>
<td>5</td>
<td>uxyvwz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Diagram: A graph with nodes u, v, w, x, y, and z connected by edges with weights labeled on the diagram. The steps in the algorithm are illustrated by the progression of nodes added to the set N' and the updated distances (D) and predecessors (p) for each node.
Hierarchical Routing

- Our routing study thus far - idealization
  - All routers identical
  - Network “flat” … one single network
  - *Not* true in practice

Scale:
- with 200 million destinations:
- can’t store all dest’s in routing tables!
- routing table exchange would swamp links!

Administrative Autonomy:
- internet = network of networks
- each network admin may want to control routing in its own network
Hierarchical Routing

- Aggregate routers into regions, called “autonomous systems” (ASes)

- Routers in the same AS run same routing protocol
  - “intra-AS” routing
  - routers in different AS can run different intra-AS routing protocol

Gateway router
- Direct link to router in another AS
- Run special protocol between Ases
  - “inter-AS” routing
Interconnected ASes

• Forwarding table is configured by both intra- and inter-AS routing algorithm
  – Intra-AS sets entries for internal dests
  – Inter-AS & Intra-As sets entries for external dests
**Inter-AS tasks**

- Suppose router in AS1 receives datagram for which dest is outside of AS1
  - Router should forward packet towards one of the gateway routers, but which one?

**AS1 needs:**
1. to learn which dests are reachable through AS2 and which through AS3
2. to propagate this reachability info to all routers in AS1

Job of inter-AS routing!
Routing Protocol Hierarchy

• Intra-Domain
  – also called intra-AS (Autonomous System)
  – also called Interior Gateway Protocols (IGPs)
  – focused only on routing within a domain

• Inter-Domain
  – also called inter-AS (Autonomous System)
  – also called Exterior Gateway Protocols (EGPs)
  – focused on routing between domains
  – treat each entire AS as a single node

• Intra-Domain vs. Inter-Domain
  – difference in size
  – difference in level of trust (security requirements)
  – difference in policy
Algorithms vs. Protocols

- When a routing protocol is developed, there are multiple decisions to be made
  - Choose an algorithm
  - Chose a message format
  - Make decisions about when/how to do updates
  - Maybe build in some security
Types of Routing Protocols

IGP
- Distance Vector: RIP
- Link State: OSPF

EGP
- Path Vector: BGP

BGP
Intra-AS Routing Protocol Details
Intra-AS Routing

- Also known as Interior Gateway Protocols (IGP)
- Most common Intra-AS routing protocols:
  - RIP: Routing Information Protocol
  - OSPF: Open Shortest Path First
  - IGRP: Interior Gateway Routing Protocol (Cisco proprietary)
RIP (Routing Information Protocol)

- Distance vector algorithm
- Included in BSD-UNIX Distribution in 1982
- Distance metric: # of hops (max = 15 hops)

<table>
<thead>
<tr>
<th>destination</th>
<th>hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>u</td>
<td>1</td>
</tr>
<tr>
<td>v</td>
<td>2</td>
</tr>
<tr>
<td>w</td>
<td>2</td>
</tr>
<tr>
<td>x</td>
<td>3</td>
</tr>
<tr>
<td>y</td>
<td>3</td>
</tr>
<tr>
<td>z</td>
<td>2</td>
</tr>
</tbody>
</table>
RIP Advertisements

• Distance vectors: exchanged among neighbors every 30 sec via Response Message (also called advertisement)
• Each advertisement: list of up to 25 destination nets within AS
RIP: Link Failure and Recovery

If no advertisement heard after 180 sec -->
neighbor/link declared dead
  – routes via neighbor invalidated
  – new advertisements sent to neighbors
  – neighbors in turn send out new advertisements (if tables changed)
  – link failure info quickly propagates to entire net
  – poison reverse used to prevent ping-pong loops (infinite distance = 16 hops)
Sample Protocol Header: RIPv2

- Encapsulated in UDP
  - IP protocol 17 (UDP), then port 520
- OSPF is encapsulated in IP
  - IP protocol type 89
- BGP runs on top of TCP
  - IP protocol 6 (TCP), then port 179
**OSPF (Open Shortest Path First)**

- “Open”: means publicly available

- Uses link state algorithm
  - LS packet dissemination
  - Topology map at each node
  - Route computation using Dijkstra’s algorithm

- OSPF advertisement inc. one entry per neighbor router

- Advertisements disseminated to entire AS (via flooding)
  - Carried in OSPF messages directly over IP (rather than TCP or UDP)
Inter-AS Routing Protocol Details
Internet Inter-AS Routing: BGP

• BGP (Border Gateway Protocol): the de facto inter-domain standard

• BGP provides each AS a means to:
  1. Obtain subnet reachability information from neighboring ASs.
  2. Propagate the reachability information to all routers internal to the AS.
  3. Determine “good” routes to subnets based on reachability information and policy.

• Allows a subnet to advertise its existence to rest of the Internet: “I am here”
**BGP Basics**

- Pairs of routers (BGP peers) exchange routing info over semi-permanent TCP connections: **BGP sessions**

- Note that BGP sessions do not correspond to physical links.

- When AS2 advertises a prefix to AS1, AS2 is *promising* it will forward any datagrams destined to that prefix towards the prefix.
  - AS2 can aggregate prefixes in its advertisement
BGP

• Terminology
  – **Stub network**: only one connection
  – **Multi-connected**: more than one connection, but no transit
  – **Transit network**: carry non-local traffic
**BGP and Path Vectors**

- **Path Vectors**
  - Carries AS path in advertisement message
  - Includes per-hop attributes
  - Can be used to enforce policy

- Forwarding table still looks the same!!!
Path Attributes & BGP Routes

• When advertising a prefix, advert includes BGP attributes.
  – prefix + attributes = “route”

• Two important attributes:
  – **AS-PATH**: contains the ASs through which the advert for the prefix passed: AS 67 AS 17
  – **NEXT-HOP**: Indicates the specific internal-AS router to next-hop AS. (There may be multiple links from current AS to next-hop-AS.)

• When gateway router receives route advert, uses **import policy** to accept/decline.
**BGP Route Selection**

- Router may learn about more than 1 route to some prefix. Router must select route.

- **Elimination rules:**
  1. Local preference value attribute: policy decision
  2. Shortest AS-PATH
  3. Closest NEXT-HOP router: hot potato routing
  4. Additional criteria
BGP Messages

• BGP messages exchanged using TCP

• BGP messages:
  – OPEN: opens TCP connection to peer and authenticates sender
  – UPDATE: advertises new path (or withdraws old)
  – KEEPALIVE keeps connection alive in absence of UPDATES; also ACKs OPEN request
  – NOTIFICATION: reports errors in previous msg; also used to close connection
**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
BGP Routing Policy Example #2

- A advertises to B the path AW
- B advertises to X the path BAW
- Should B advertise to C the path BAW?
  - No way! B gets no “revenue” for routing CBAW 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!
Why Different Intra- and 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, reduces update traffic

Performance:
• Intra-AS: can focus on performance
• Inter-AS: policy may dominate over performance
Summary
Outline

• Review: Routing v. Forwarding
• Routing Algorithms
  – Distance Vector
  – Link State
• Routing Between Domains
• Algorithms v. Protocols
• Intra-AS Routing Protocol Details
• Inter-AS Routing Protocol Details
• Summary and Notables
Process Summary

• ASes choose their intra-domain routing protocol
  – Exchange information about neighbors and determine routing topology
  – If a stub network, often include a “default route” to the Internet
  – Have a gateway/border router that is part of internal network
    • But acts as a proxy for the rest of the Internet (“send that traffic to me”)

• ASes have border routers that speak BGP
  – Less complex for stub networks
  – Backbone networks have to decide with whom they exchange traffic

• Always keep in mind: however routing table is built, there is always a forwarding table and it always looks the same
  – The forwarding table always indicates the next hop
  – Note: the forwarding table might have been built using network-wide information, but the decision of which outgoing interface to take is still the decision of each router along a path
Types of Routing Protocols

IGP
- Distance Vector
  - RIP
- Link State
  - OSPF

EGP
- Path Vector
  - BGP
Routing Summary Checkoff

• Routing vs. Forwarding

• Intra-AS Basics
  – Distance vector vs. link state (including tradeoffs between the two)
  – Algorithm vs. protocol (know what algorithms go with which protocols)

• Inter-AS Basics
  – What BGP does and how it is different than intra-AS routing

• Reasons for why routing is broken into intra- and inter-AS routing