Chapter 5 (Final Study Guide)

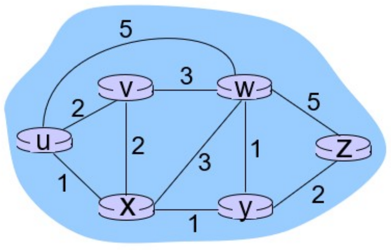
PRE ROUTER CONTROL PLANE

-Individual routing algorithm components in each and every router interact in the control plane

ROUTING PROTOCOLS

Routing protocol goal: determine "good" paths (equivalently, routes), from sending hosts to receiving host, through network of routers

GRAPH ABSTRACTION OF THE NETWORK



routing algorithm: algorithm that finds that least cost path

ROUTING ALGORITHM CLASSIFICATION

Q: global or decentralized information?

global:

- all routers have complete topology, link cost info
- "link state" algorithms

decentralized:

- router knows physically-connected predecessors, link costs to predecessors
- iterative process of computation, exchange of info with predecessors
- "distance vector" algorithms

Q: static or dynamic?

static:

-routes change slowly over time

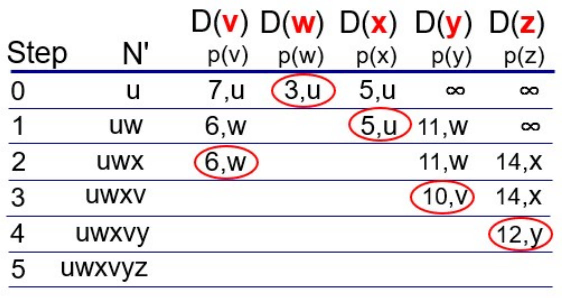
dynamic:

- routes change more quickly
    - periodic update
    - in response to link cost changes

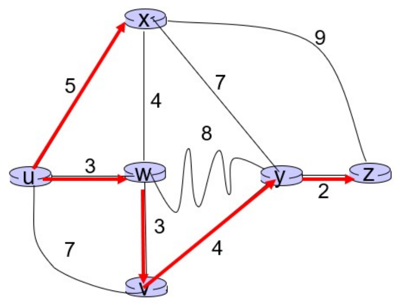
Dijkstra's algorithm (EXAM QUESTION)

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



Step	N'	D(v) p(v)	D(w) p(w)	D(x) p(x)	D(y) p(y)	D(z) p(z)
0	u	7,u	3,u	5,u	∞	∞
1	uw	6,w		5,u	11,w	∞
2	uwx	6,w			11,w	14,x
3	uwxv				10,v	14,x
4	uwxvy					12,y
5	uwxvyz					



NOTATION

c(x,y): link cost from node x to y; = ∞ if not direct predecessors
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
D(v) = min(D(v), D(w) + c(w,v))

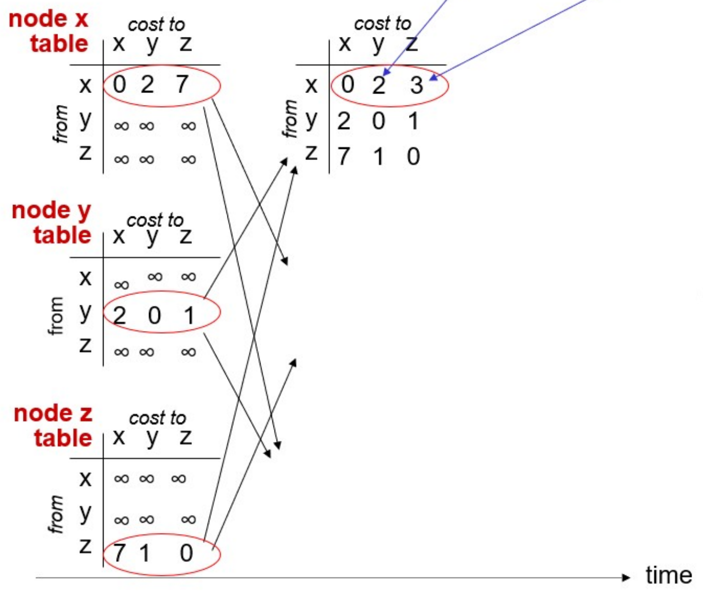
DISTANCE VECTOR ALGORITHM

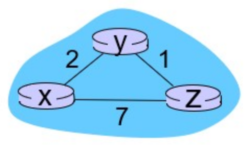
key idea:

- from time-to-time, each node sends its own distance vector estimate to neighbors
- when x receives new DV estimate from neighbor, it updates its own DV using Bellman-Ford equation:

$$D_x(y) = \min\{c(x,y) + D_y(y),\ c(x,z) + D_z(y)\} = \min\{2+0\ ,\ 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z),\ c(x,z) + D_z(z)\} = \min\{2+1\ ,\ 7+0\} = 3$$





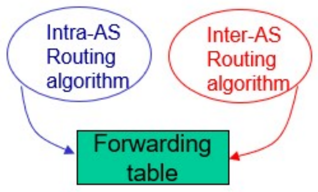
INTERNET APPROACH SCALABLE ROUTING

intra-AS routing
 -routing among hosts, routers in same AS ("network")
 -all routers in AS must run same intra-domain protocol

inter-AS routing (also known as IGP - Interior Gateway Protocols)
 -routing among AS'es
 -gateways perform inter-domain routing (as well as intra-domain routing)

Interconnected ASes

- forwarding table configured by both intra- and inter-AS routing algorithm
    - intra-AS routing determine entries for destinations within AS
    - inter-AS & intra-AS determine entries for external destinations

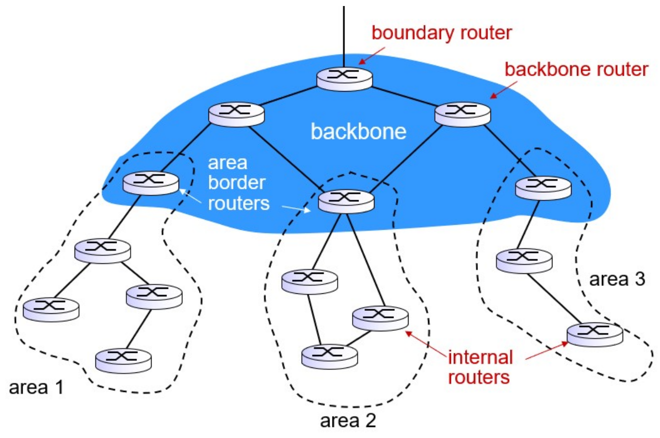


OSPF (Open Shortest Path First)

- "open": publicly available
- uses link-state algorithm
- A router broadcasts OSPF link-state advertisements to all other routers in entire AS
- IS-IS routing protocol: nearly identical to OSPF

Advanced OSPF Operations

- security: all OSPF messages authenticated (to prevent malicious intrusion)
- multiple same-cost paths allowed (only one path in RIP)
- for each link, multiple cost metrics for different TOS (e.g., satellite link cost set low for best effort ToS; high for real-time ToS)
- integrated uni- and multi-cast support:
    - Multicast OSPF (MOSPF) uses same topology data base as OSPF
- hierarchical OSPF in large domains.

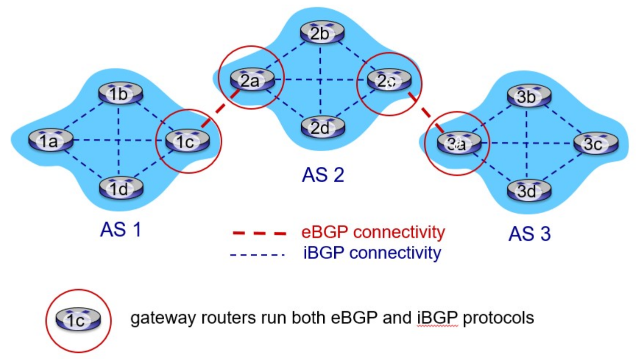


Internet inter-AS routing: BGP

BGP (Border Gateway Protocol): the de facto inter-domain routing protocol

BGP provides each AS a means to:

- eBGP: obtain subnet reachability information from neighboring ASes
- iBGP: propagate reachability information to all AS-internal routers.
- determine "good" routes to other networks based on reachability information and policy

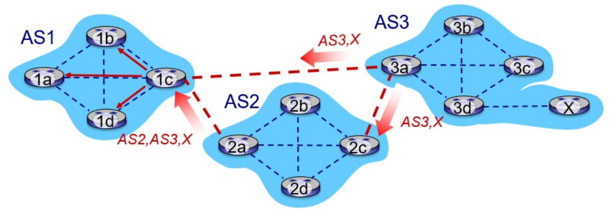


Policy-based routing:

- 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 neighboring ASes

BGP path advertisement

BGP path advertisement



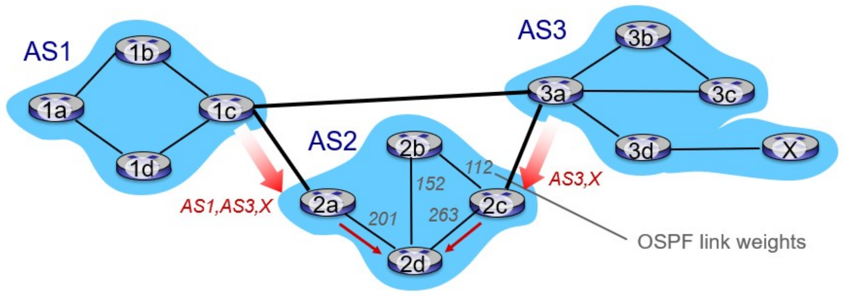
gateway router may learn about multiple paths to destination:

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

BGP MESSAGES

- BGP messages exchanged between peers over TCP connection
- BGP messages:
    - OPEN: opens TCP connection to remote BGP peer and authenticates sending BGP peer
    - 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

Hot Potato Routing



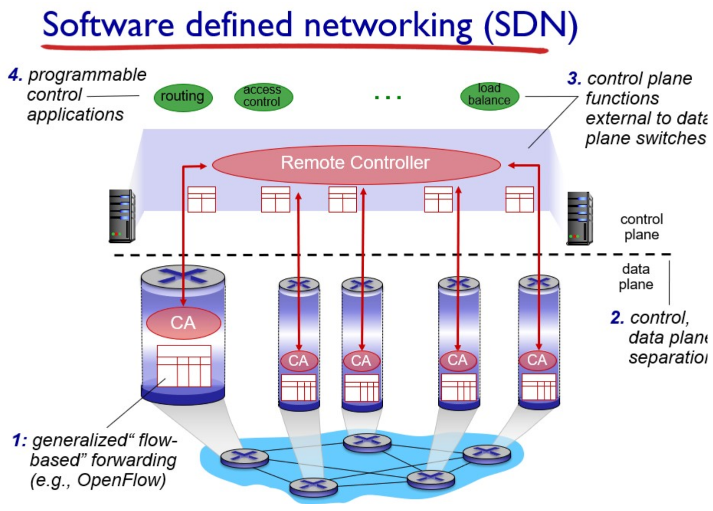
- 2d learns (via iBGP) it can route to X via 2a or 2c
- hot potato routing: choose local gateway that has least intra-domain cost (e.g., 2d chooses 2a, even though more AS hops to X): don't worry about inter- domain cost!

Software Defined Networking (SDN)

Internet network layer: historically has been implemented via distributed, per-router approach

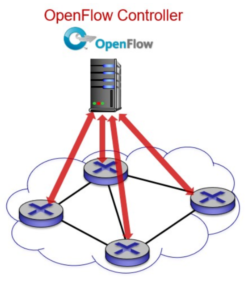
Why a logically centralized control plane?

- easier network management: avoid router misconfigurations, greater flexibility of traffic flows
- table-based forwarding allows "programming" routers
- open (non-proprietary) implementation of control plane


Software defined networking (SDN)
4. programmable control applications
routing
access control
...
load balance
3. control plane functions external to data plane switches
Remote Controller
control plane
data plane
2. control, data plane separation
CA
CA
CA
CA
CA
1: generalized" flow-based" forwarding (e.g., OpenFlow)

OpenFlow Protocol

- operates between controller, switch
- TCP used to exchange messages
    - optional encryption
- three classes of OpenFlow messages:
    - controller-to-switch
    - asynchronous (switch to controller)
    - symmetric (misc)



OpenFlow: Control to switch messages

Key controller-to-switch messages
- *features:* controller queries switch features, switch replies
- *configure:* controller queries/sets switch configuration parameters
- *modify-state:* controller adds, deletes, modifies flow entries in the OpenFlow tables
- *packet-out:* controller can send this packet out of specific switch port

OpenFlow: Switch to control messages

Key switch-to-controller messages
- *packet-in:* transfer packet (and its control) to controller. See packet-out message from controller
- *flow-removed:* inform controller that flow table entry deleted at switch
- *port status:* inform controller of a change on a port.

Network Management

- autonomous systems (aka "network"): thousands of interacting hardware/software components
- other complex systems requiring monitoring, control:
    - -jet airplane
    - - nuclear power plant
    - - others?

"***Network management*** includes the deployment, integration and coordination of the hardware, software, and human elements to monitor, test, poll, configure, analyze, evaluate, and control the network and element resources to meet the real-time, operational performance, and Quality of Service requirements at a reasonable cost."