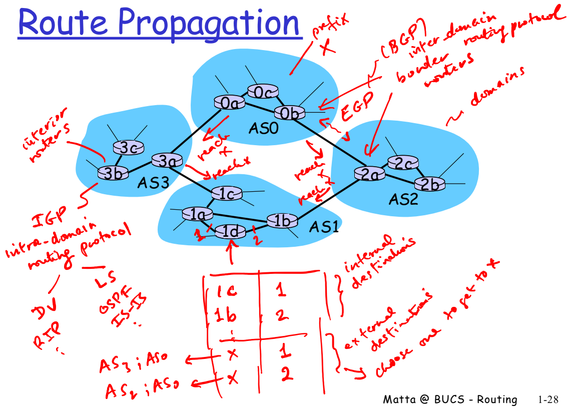
Route Propagation



28

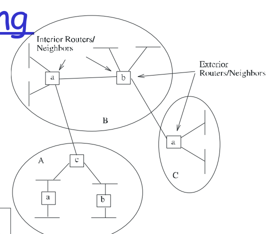
Route Propagation

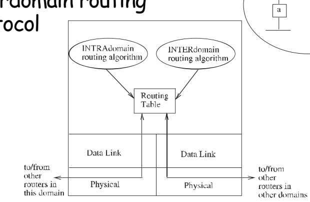
- Idea: Impose a second hierarchy on the network that limits which routers talk to each other. (The first hierarchy is the address hierarchy that governs how packets are forwarded.)
- Autonomous System (AS)
  - corresponds to an administrative domain/region
  - examples: University, company, backbone network
  - assign each AS a 16-bit number
- Two-level route propagation hierarchy
  - interior gateway protocol (each AS selects its own)
  - exterior gateway protocol (Internet-wide standard)

29

Hierarchical Routing

- A.a, A.b, A.c run an intradomain routing protocol
- A.c, B.a, B.b, C.a run an interdomain routing protocol





30

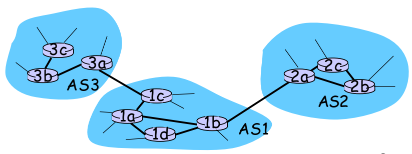
Inter-AS tasks

- Suppose router in AS1 receives datagram for which destination 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!



31

Example: Setting forwarding table in router 1d

- Suppose AS1 learns from the inter-AS protocol that net *x* is reachable from AS3 (gateway 1c) but not from AS2
- Inter-AS protocol propagates reachability info to all internal routers
- Router 1d determines from intra-AS routing info that its interface *I* is on the least cost path to 1c
- Puts in forwarding table entry *(x,I)*

32

Example: Choosing among multiple ASes

- Now suppose AS1 learns from the inter-AS protocol that net *x* is reachable from AS3 *and* from AS2
- To configure forwarding table, router 1d must determine towards which gateway it should forward packets for dest x
- This is also the job of inter-AS routing protocol!
- Hot potato routing: send packet towards closest of two routers

Learn from inter-AS protocol that net x is reachable via multiple gateways → Use routing info from intra-AS protocol to determine costs of least-cost paths to each of the gateways → Hot potato routing: Choose the gateway that has the smallest least cost → Determine from forwarding table the interface I that leads to least-cost gateway. Enter (x,I) in forwarding table

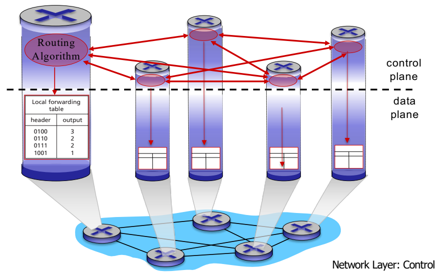
33

Software defined networking (SDN)

- Internet network layer: historically has been implemented via distributed, per-router approach
  - *monolithic* router contains switching hardware, runs proprietary implementation of Internet standard protocols (IP, RIP, IS-IS, OSPF, BGP) in proprietary router OS (e.g., Cisco IOS)
  - different "middleboxes" for different network layer functions: firewalls, NAT boxes, ..
- ~2005: renewed interest in rethinking network control plane

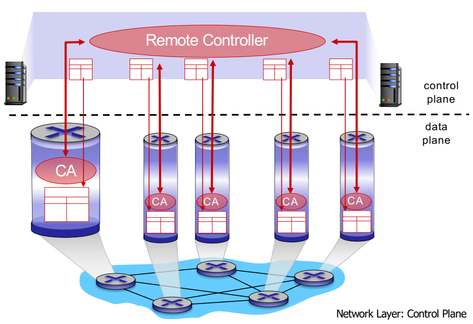
Recall: per-router control plane

Individual routing algorithm components *in each and every router* interact with each other in control plane to compute forwarding tables



Generalized Forwarding and SDN

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



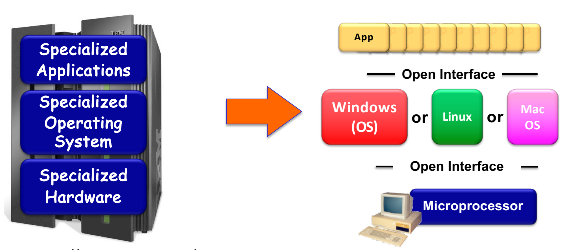
Software defined networking (SDN)

Why a *logically centralized* control plane?

- easier network management: avoid router misconfigurations, greater flexibility of traffic flows
- centralized "programming" easier: compute tables centrally and distribute
- open (non-proprietary) implementation of control plane

37

Analogy: mainframe to PC evolution*

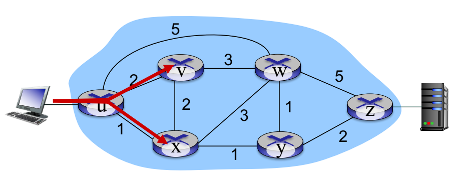


Vertically integrated		Horizontal
Closed, proprietary	→	Open interfaces
Slow innovation		Rapid innovation
Small industry		Huge industry

* Slide courtesy: N. McKeown

38

Traffic engineering: difficult

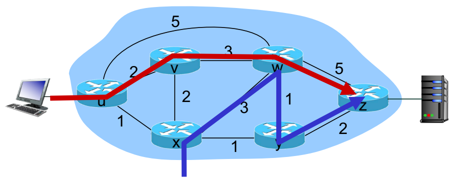


Q: what if network operator wants to split u-to-z traffic along uvwz *and* uxyz (load balancing)?

A: can't do it (or need a new routing algorithm)

39

Traffic engineering: difficult

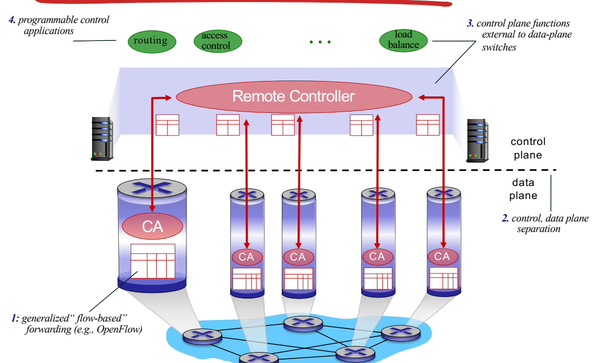


Q: what if w wants to route blue and red traffic differently?

A: can't do it (with destination based forwarding, and LS, DV routing)

40

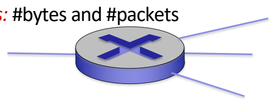
Software defined networking (SDN)



41

OpenFlow data plane abstraction

- *flow*: defined by header fields
- generalized forwarding: simple packet-handling rules
  - *Pattern:* match values in packet header fields
  - *Actions: for matched packet:* drop, forward, modify, matched packet or send matched packet to controller
  - *Priority*: disambiguate overlapping patterns
  - *Counters:* #bytes and #packets

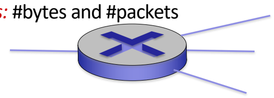


Flow table in a router (computed and distributed by controller) define router's match+action rules

42

OpenFlow data plane abstraction

- *flow*: defined by header fields
- generalized forwarding: simple packet-handling rules
  - *Pattern:* match values in packet header fields
  - *Actions: for matched packet:* drop, forward, modify, matched packet or send matched packet to controller
  - *Priority*: disambiguate overlapping patterns
  - *Counters:* #bytes and #packets



* : wildcard

1. src=1.2.*.*, dest=3.4.5.* → drop
2. src = *.*.*.*, dest=3.4.*.* → forward(2)
3. src=10.1.2.3, dest=*.*.*.* → send to controller

43

OpenFlow: Flow Table Entries

Rule	Action	Stats

Stats: Packet + byte counters

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

Rule:

Switch Port	VLAN ID	MAC src	MAC dst	Eth type	IP Src	IP Dst	IP Prot	TCP sport	TCP dport
	Link layer				Network layer			Transport layer	

44

Examples

Destination-based forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	51.6.0.8	*	*	*	port6

IP datagrams destined to IP address 51.6.0.8 should be forwarded to router output port 6

Firewall:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Forward
*	*	*	*	*	*	*	*	*	22	drop

do not forward (block) all datagrams destined to TCP port 22

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Forward
*	*	*	*	*	128.119.1.1	*	*	*	*	drop

do not forward (block) all datagrams sent by host 128.119.1.1

45

OpenFlow abstraction

- *match+action:* unifies different kinds of devices

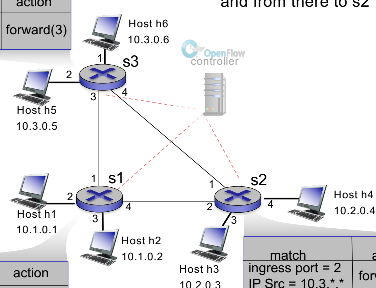
- Router
  - *match:* longest destination IP prefix
  - *action:* forward out a link
- Switch
  - *match:* destination MAC address
  - *action:* forward or flood
- Firewall
  - *match*: IP addresses and TCP/UDP port numbers
  - *action:* permit or deny
- NAT
  - *match:* IP address and port
  - *action:*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

46

OpenFlow example

Example: datagrams from hosts h5 and h6 should be sent to h3 or h4, via s1 and from there to s2

match	action
IP Src = 10.3.*.* IP Dst = 10.2.*.*	forward(3)



match	action
ingress port = 1 IP Src = 10.3.*.* IP Dst = 10.2.*.*	forward(4)

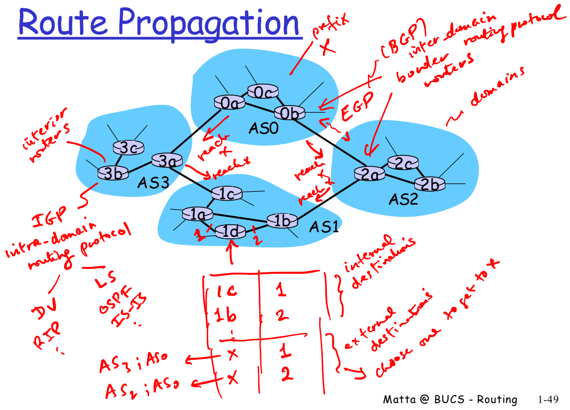
match	action
ingress port = 2 IP Src = 10.3.*.* IP Dst = 10.2.0.3	forward(3)
ingress port = 2 IP Src = 10.3.*.* IP Dst = 10.2.0.4	forward(4)

47

BACK TO BGP

48

Route Propagation



Matta @ BUCS - Routing 1-49

49

BGP: Border Gateway Protocol

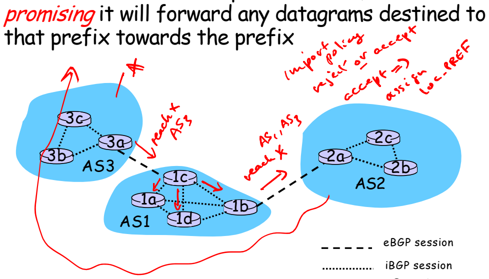
- Assumes the Internet is an arbitrarily interconnected set of AS's
- Define local traffic as traffic that originates at or terminates on nodes within an AS, and transit traffic as traffic that passes through an AS
- AS's classified into three types:
  - Stub AS: an AS that has only a single connection to one other AS; such an AS will only carry local traffic
  - Multihomed AS: an AS that has connections to more than one other AS, but refuses to carry transit traffic
  - Transit AS: an AS that has connections to more than one other AS, and is designed to carry both transit and local traffic

Matta @ BUCS - Routing 1-50

50

BGP basics

- Pairs of routers (BGP peers) exchange routing info over TCP connections: BGP sessions
- When AS2 advertises a prefix to AS1, AS2 is *promising* it will forward any datagrams destined to that prefix towards the prefix

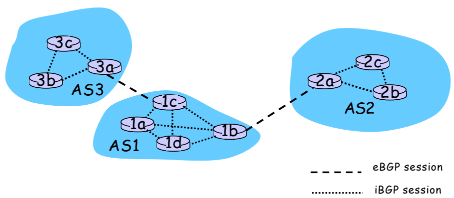


Matta @ BUCS - Routing 1-51

51

Distributing reachability info

- With eBGP session between 3a and 1c, AS3 sends prefix reachability info to AS1
- 1c can then use iBGP to distribute this new prefix reach info to all routers in AS1
- 1b can then re-advertise the new reachability info to AS2 over the 1b-to-2a eBGP session
- When router learns about a new prefix, it creates an entry for the prefix in its forwarding table



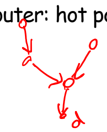
Path attributes & BGP routes

- When advertising a prefix, advertisement includes BGP attributes
  - prefix + attributes = "route"
- Two important attributes:
  - AS-PATH: contains the ASs through which the advertisement 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 advertisement, uses import policy to accept/decline and attach a LOC_PREF (local preference) value to accepted paths

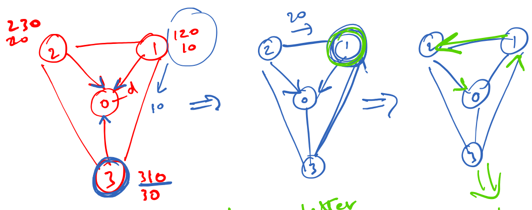
BGP route selection

- 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

If everyone does SP routing => stable tree of SPs



Convergence Not Guaranteed!

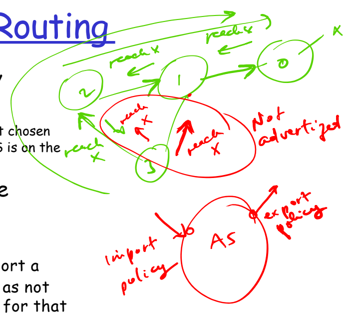


Someone always has a better path than the current one to switch to

55

Path Vector Routing

- Loops can be completely avoided
  - Export policy: Do not export chosen path to neighbor if neighbor AS is on the path
- Policy constraints can be imposed
  - Based on ASes on paths
  - Export policy: Do not export a path to a neighbor AS so as not to carry (transit) traffic for that neighbor
  - Import policy: accept (and assign LOC_PREF) or reject



56

BGP messages

- BGP messages exchanged over TCP
- BGP messages:
  - OPEN: opens BGP peering session with 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 (e.g. unsupported option in OPEN msg); also used to close BGP session

57