Link State Routing

Contrast with the previous approaches
- first flood topology information
- then compute the routes

- decentralized, distributed
- more computation than DV
- Used in Internet from 1979
- OSPF & IS-IS uses LS routing.

Setting

- similar to DV

(1) Nodes only know the costs to their neighbors, not the topology
(2) can talk only to neighbors
(3) can run algorithm concurrently
(4) can fail and messages may be lost.

The Algorithm

Two phases

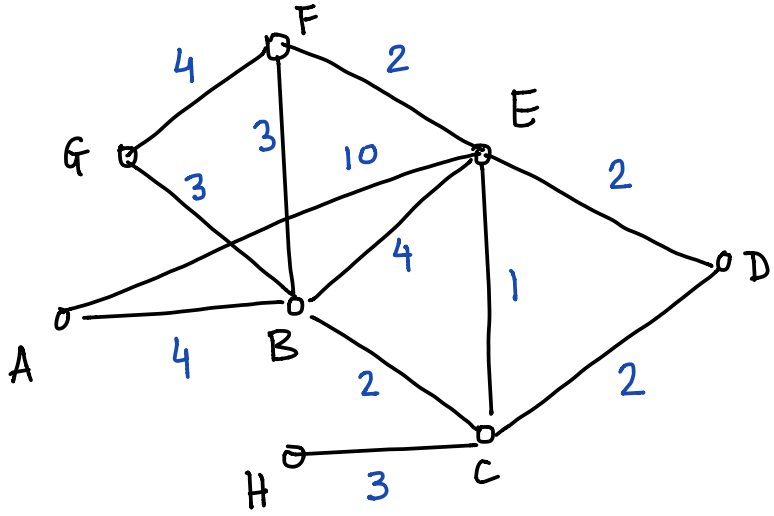
1. Nodes flood topology in the form of link state packets
   - every node learns the full topology
2. Each node computes its own forwarding table
   - by running Dijkstra (or equivalent)

Phase 1 : topology flooding

Each node floods LSP (link state packet)

E's LSP

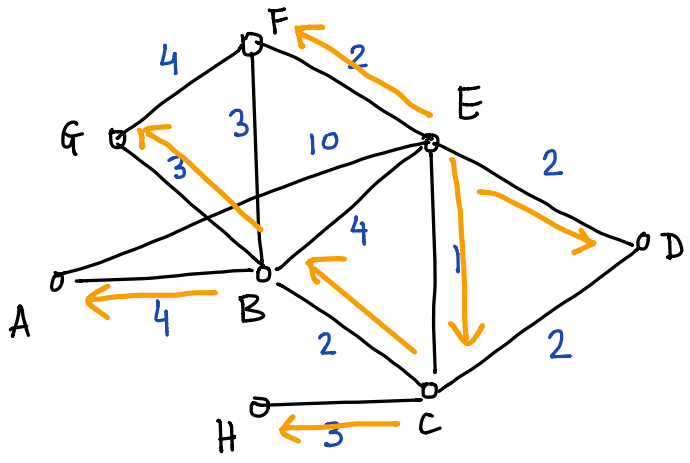
Seq #	
A	10
B	4
C	1
D	2
F	2



Phase 2: Route finding

Every node knows the complete topology
— by combining the LSPs

Each node runs Dijkstra
- possible since the whole network is known
- replicated computation
- compile forwarding tables directly [source tree from that node]



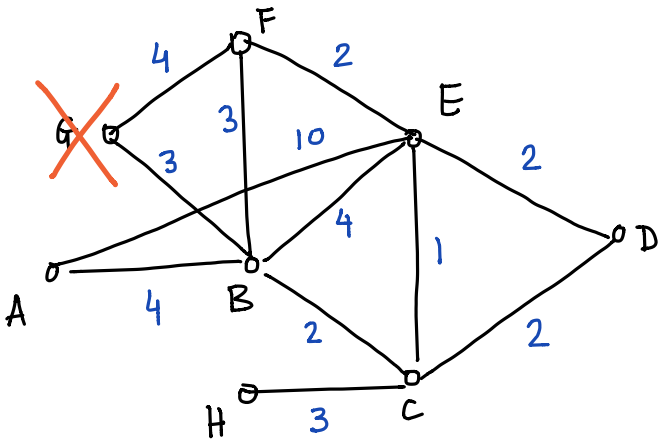
Forwarding table of E

To	Via
A	C
B	C
C	C
D	D
E	E
F	F
G	C

Assumes each node breaks ties in a consistent manner

Handling Changes

- After node/link failure, the other nodes send updated LSPs
- Every node recomputes their routes



B

A	4
C	2
E	4
F	3

F

B	3
E	2

G missing

Link failure

- Both nodes notice and issue updated LSPs
- link is removed

Node failure

- all neighbors notice and issue updated LSPs
- all edges to that node is removed.

Link/Node additions are also flooded via LSPs of the adjacent/neighboring nodes
- Easier case

There are certain complications of LS in practice, [flooding pkt corrupted, diff tie breaking]
- Real world implementations are engineered to handle and be robust against such corner cases.

Comparison between DV and LS

Goals	DV	LS
Correctness	distributed Bellman Ford	Replicated Dijkstra
Efficient path	shortest path	shortest path
Fast recovery [after failure]	Slow - many exchanges	Fast - flood and compute
Scalability	Excellent	Moderate

Use of LS Routing

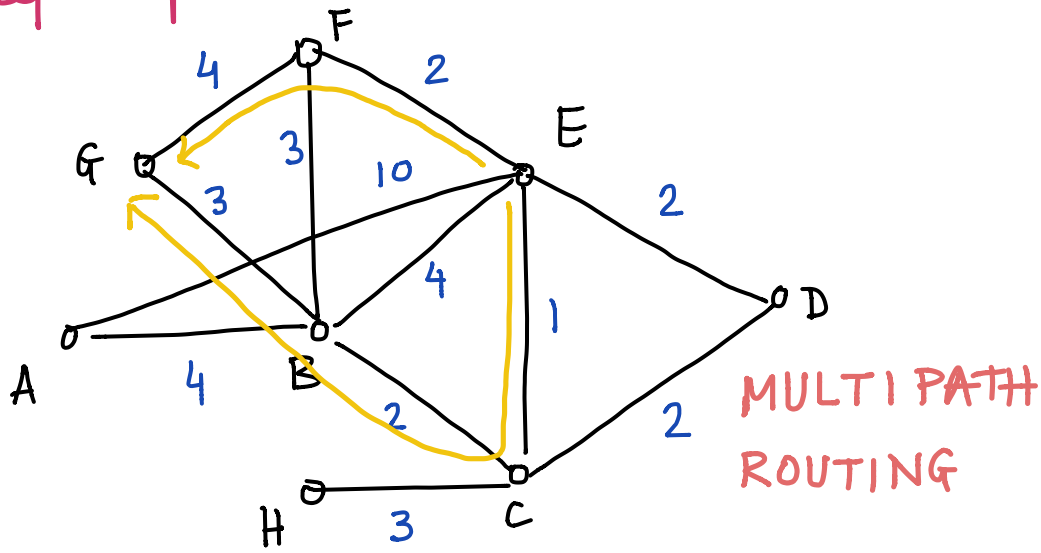
Widely used in large enterprise and ISP networks

- IS-IS = Intermediate System to Intermediate System
- OSPF = Open Shortest Path First.

Equal Cost Multipath Routing

A simple extension of the shortest path algorithm

"keep all paths to a destination"



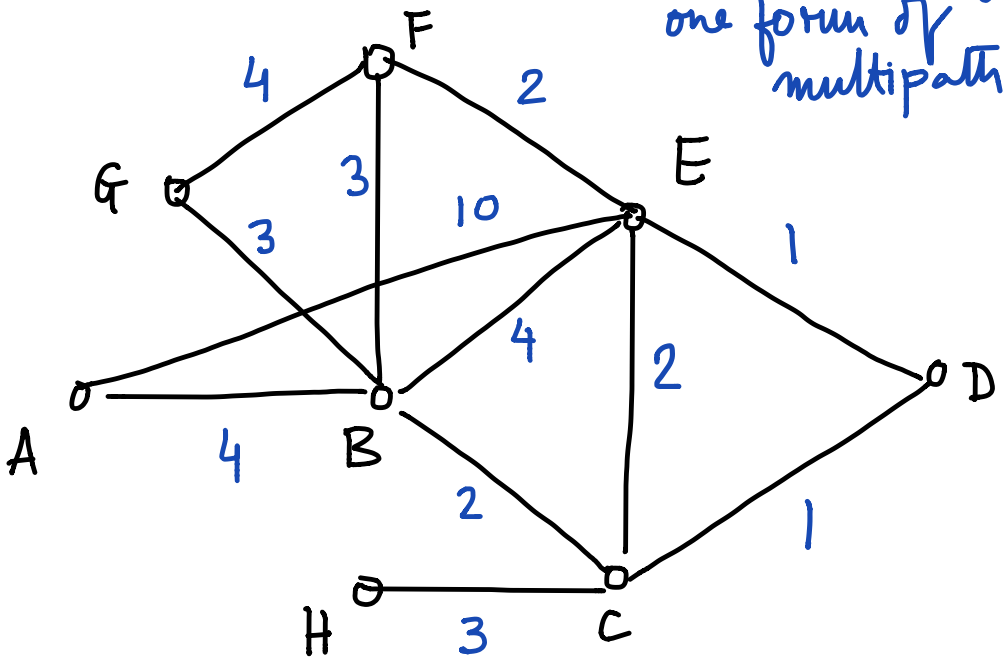
Multipath Routing

Allow multiple routing paths from node to destination be used at once

Why?
- Redundancy
- Improve performance [simultaneously multiple traffic flow]

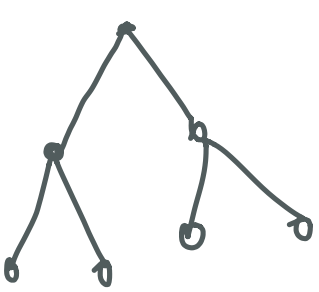
Q:
- How to find multiple paths?
- How to send traffic through them?

ECMP routing is one form of multipath routing

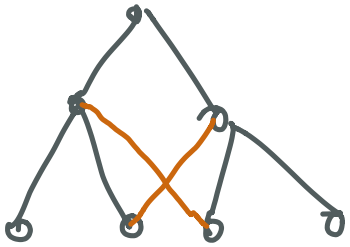


Keep all paths if there is a tie

Obs: with ECMP routing, source trees become DAGs



Tree

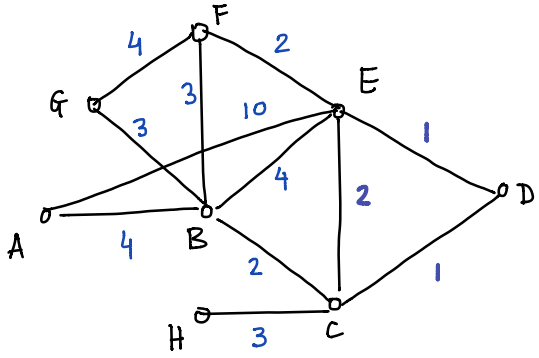


DAG.

Modified algorithms to find source "DAG"

Dijkstra

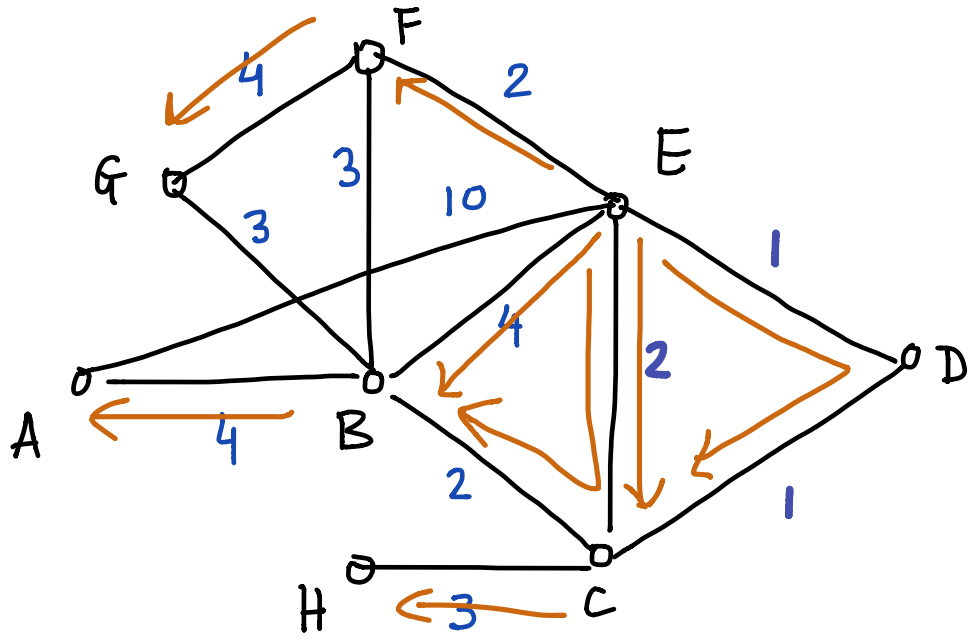
instead of picking an arbitrary next hop in case of a tie, add all



Distance Vector

Update the DV with set of all neighbors

(distance, set of via nodes)



E's forwarding table

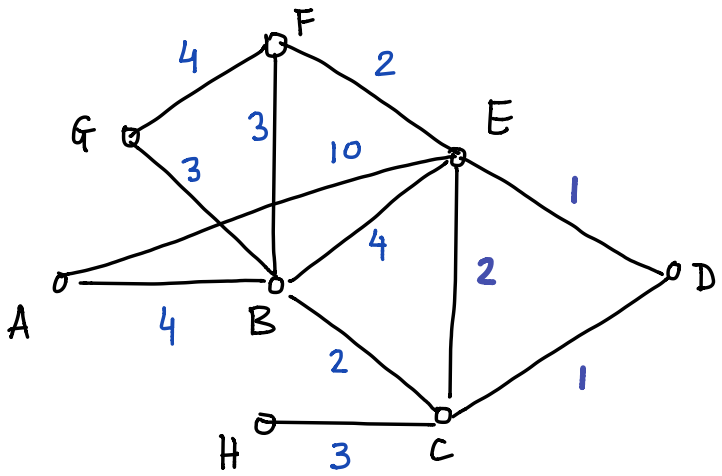
A	B, C, D
B	B, C, D
C	C, D
D	D
E	–
F	F
G	F
H	C, D

- Find multiple paths
- Route through them

Forwarding with ECMP

① Pick a node to forward uniformly at random
- Balanced load
- Same traffic, via different path may encounter different delay / quality "jitter" in the traffic

② Forward packets for a given source/destination pair to a fixed node
- a source-dest pair is called a flow
- same next hop for a flow
- less jitter, but more unbalanced.



E's forwarding table

flow	
F → C	C
E → C	C
F → H	D
E → H	D

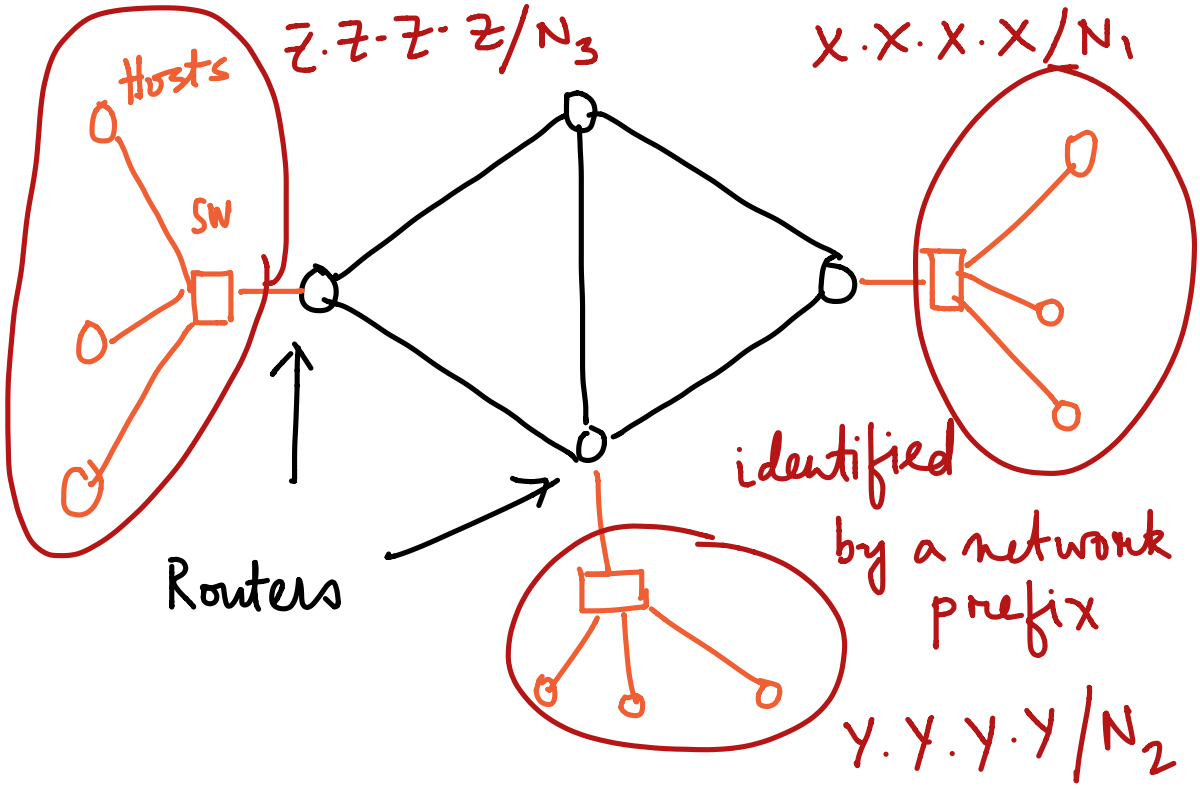
How Routers and Hosts interact?

forwarding tables would become unmanageable if all nodes were part of the routing process.

Routers route, Hosts don't

Recap:

- Hosts on the same network have same network prefix
- Hosts send off-network traffic to the nearest router
- Routers discover routes
- Do longest prefix matching to send packets to next hop



Routers are identified by the network prefix they represent.

Scalable routing : Hierarchies

Routers often route to regions and not to specific routers

- scale, keep forwarding tables compact
- routing messages grow
- computation grows

Approaches

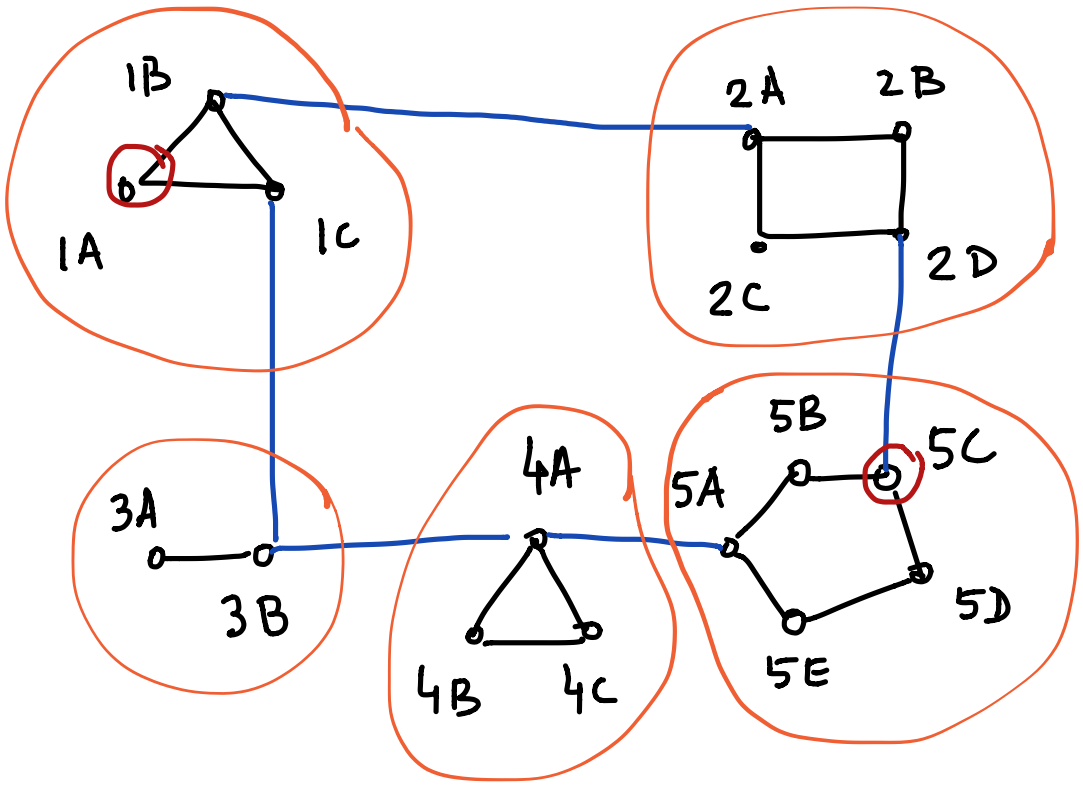
- IP prefixes -
- Hierarchical routing (HR)

HR introduces a larger routing unit

- IP prefix - already in use
- Region with multiple ip prefixes

Routing will happen in stages

1. Send to region
2. Send to the IP prefix within that region.



Full table 1A

1A		
1B	1B	1
1C	1C	1
2A	⋮	
2B		
⋮		
5C	1B	5
⋮		

Hierarchical table 1A

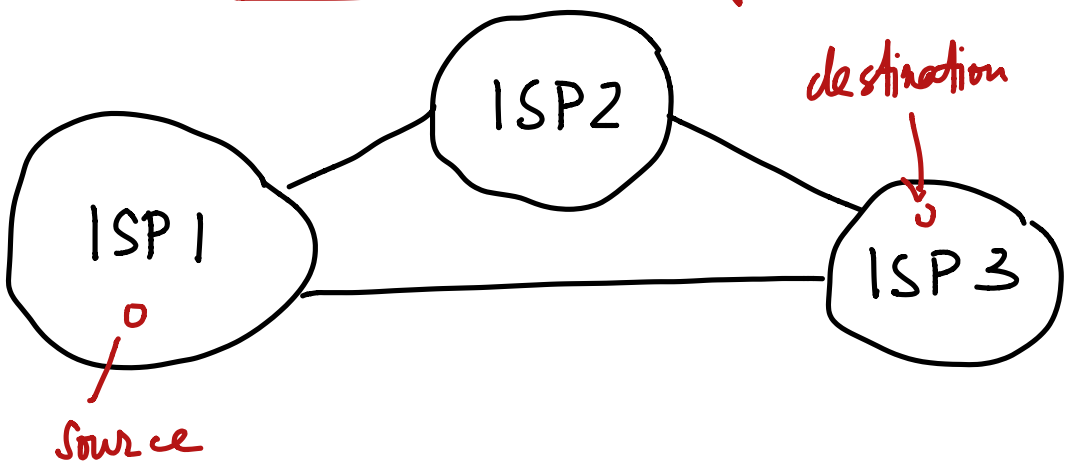
1A		
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

Simplicity – Optimality trade off.

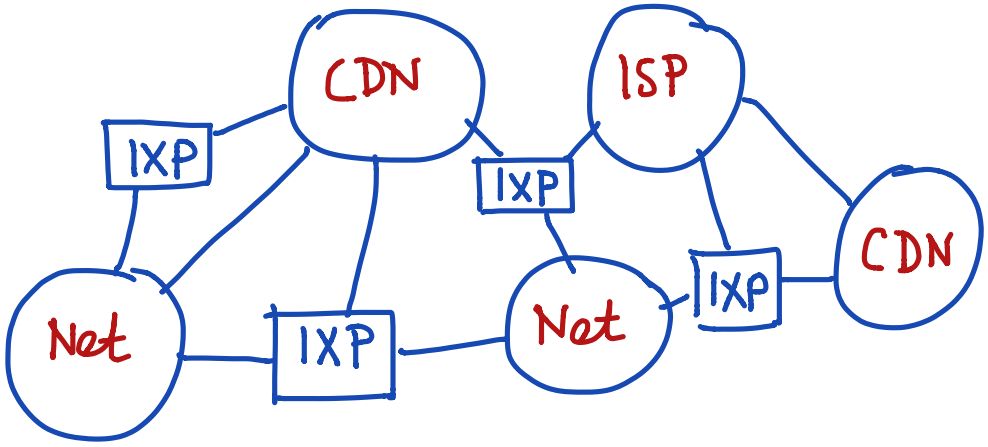
HR endnotes

- Outside the region, nodes have one route to all hosts in the region
  - saves the forwarding table space
- In the region, nodes may have different routes to outside region
  - routing decision is still at node level.

Multi Party Routing



Structure of the Internet



IXP: Internet Exchange Point

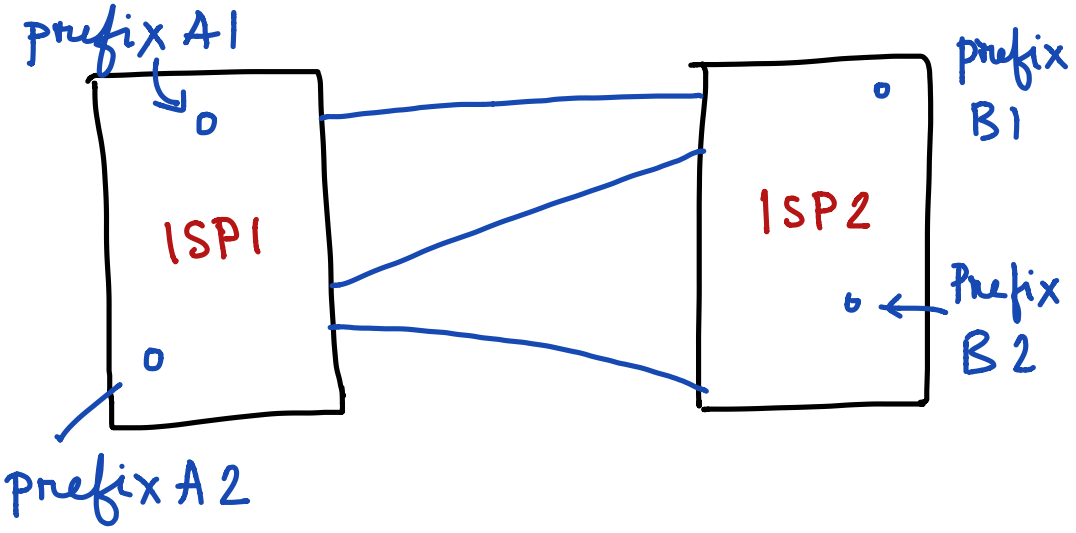
Routing Issues with multiple networks

(1) Scaling to very large networks
- IP prefix, hierarchical routing

(2) Routing policies within a network
- network is managed by independent entities, they have their own policies.

[Focus here]

Ex.



Policy : Shortest exit point within the network (both)

Route : A2 → B1 and B1 → A2

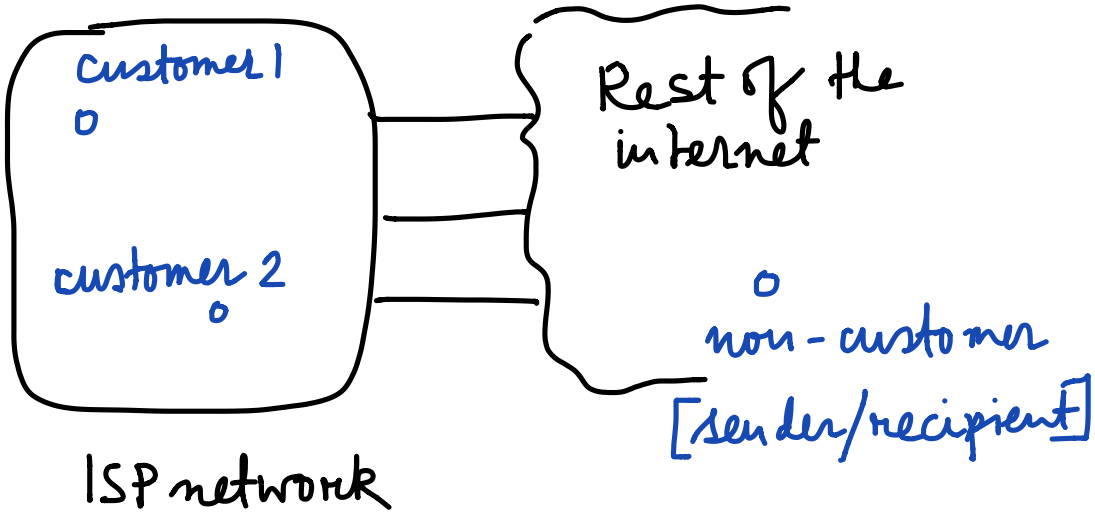
Routing Policies

- Set by the owner of the network
  - e.g. educational networks don't carry commercial traffic [identified by the destination prefix]

Common policies

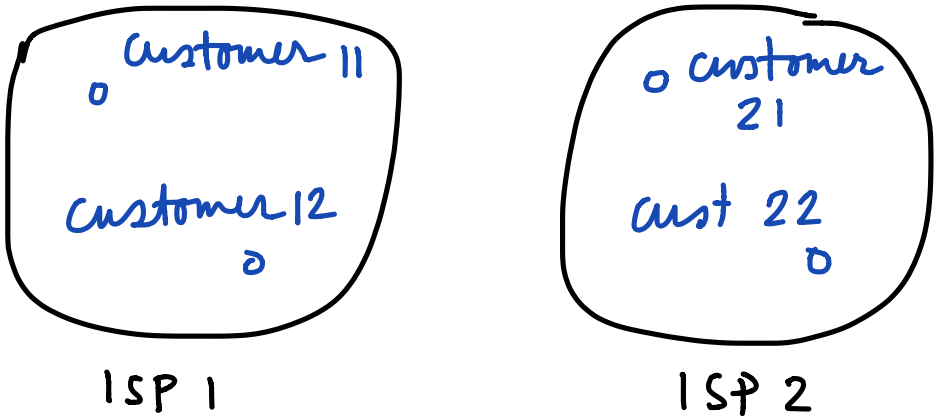
- TRANSIT - ISPs to customers
- PEER - ISPs to each other

TRANSIT Policy



- ISP will send the packet to a non-customer and receive incoming packet and deliver to the customer
- customer will pay the ISP for this service

PEER policy



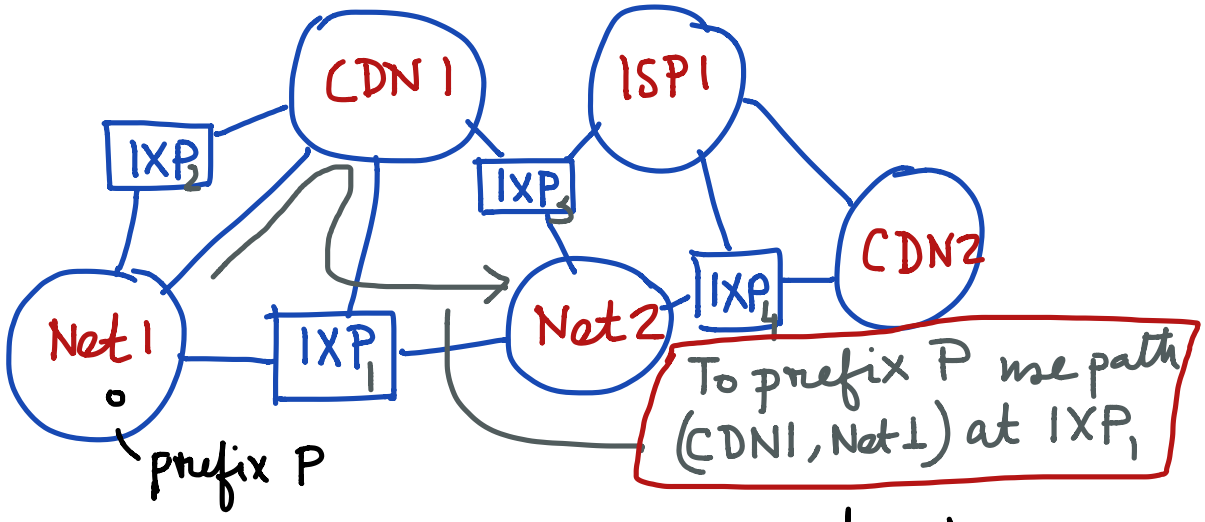
- policy ensures delivery only if the destination is in its network
- ISPs do not carry traffic which has a destination outside ISP's network [under this policy]
- ISPs don't pay each other

Border Gateway Protocol

Q: How to route packets through multiple parties, each having its own routing policies?

A: BGP computes internet-wide routes

BGP computes interdomain routes in the internet.

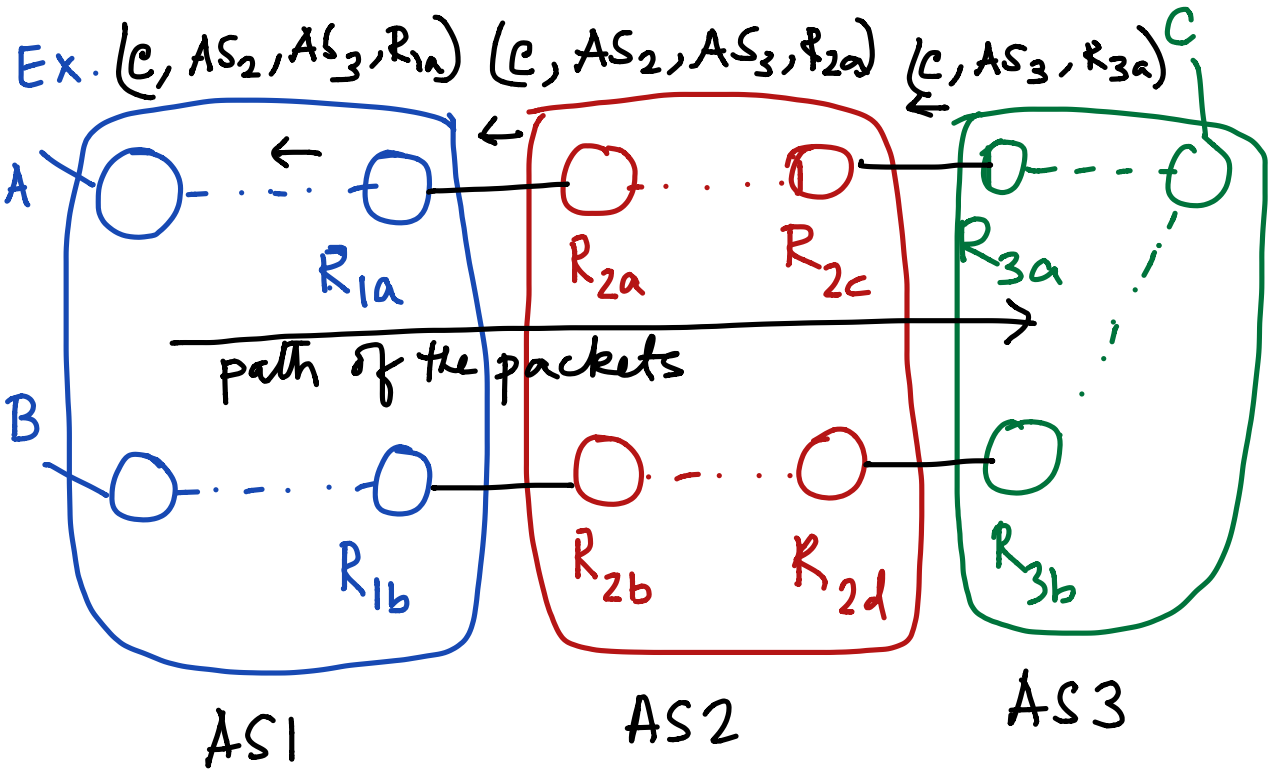


- BGP uses a Path Vector, closely related to a distance vector
- Path vector is an announcement of how that network can be reached.

More terminology

- Different parties are called Autonomous Systems (AS)
- Border routers of ASes announce BGP routes to each other

- Route announcements contain
  (IP prefix, path vector, next hop)
  - path vector is the list of ASes on the way to the prefix (policy, loops)
- Route announcements move in the opposite direction of traffic

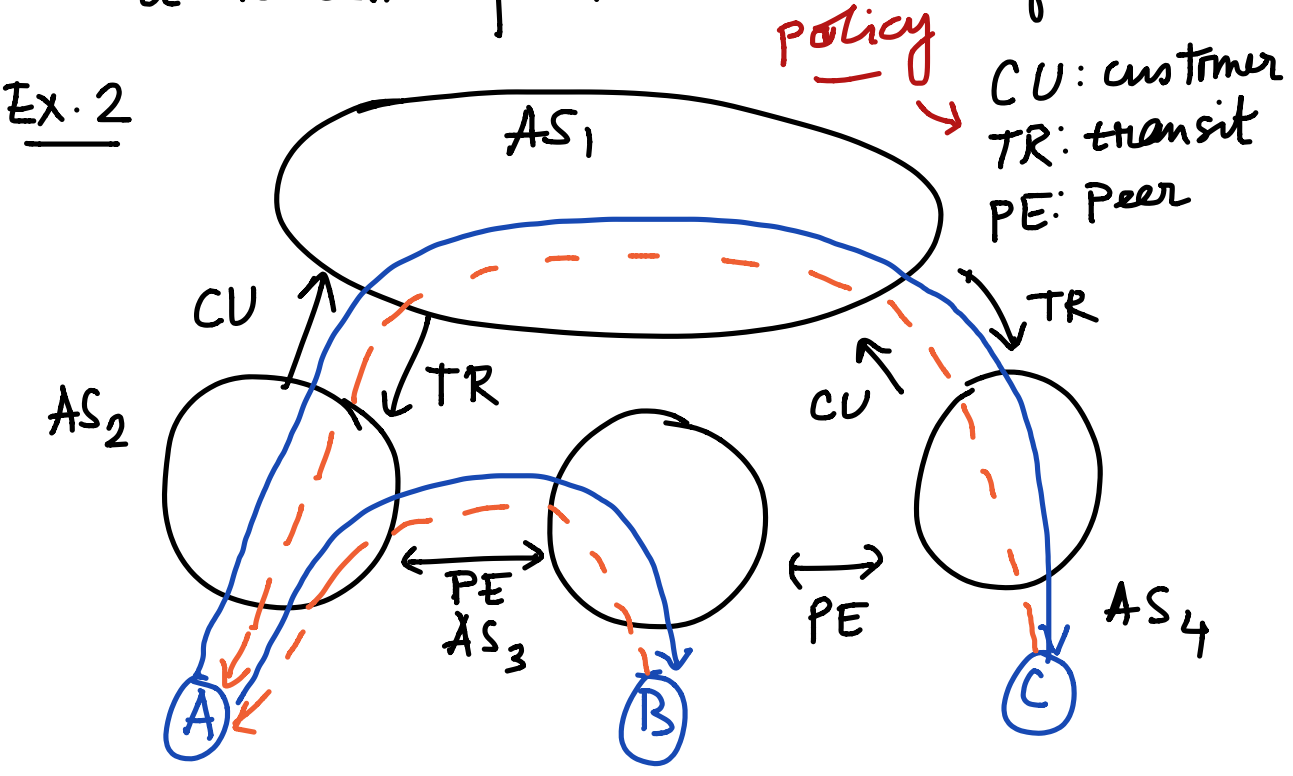


Policy implementation through BGP

(1) Border routers announce paths to only those ASes that may use that path

(2) Border routers of ISP select the best path among the ones it receives path announcement from

Ex. 2



AS_2 buys TR service from AS_1, AS_2 & AS_3 ⇒ PE

AS_2 sends $(A, (AS_2))$ to AS_1 — CU

AS_1 sends $(B, (AS_1, AS_3))$ to AS_2
$(C, (AS_1, AS_4))$ — TR

AS_2 sends $(A, (AS_2))$ to AS_3
AS_3 sends $(B, (AS_3))$ to AS_2 — PE

AS_2 heard one route to C (TR)
two routes to B (TR & PE)

Endnotes:

1. BGP is more detailed - this is a first example
2. Policy is important
3. Convergence, scalability, and more.