Routing Overview

EECS 228

Abhay Parekh

parekh@eecs.berkeley.edu

October 7, 2002

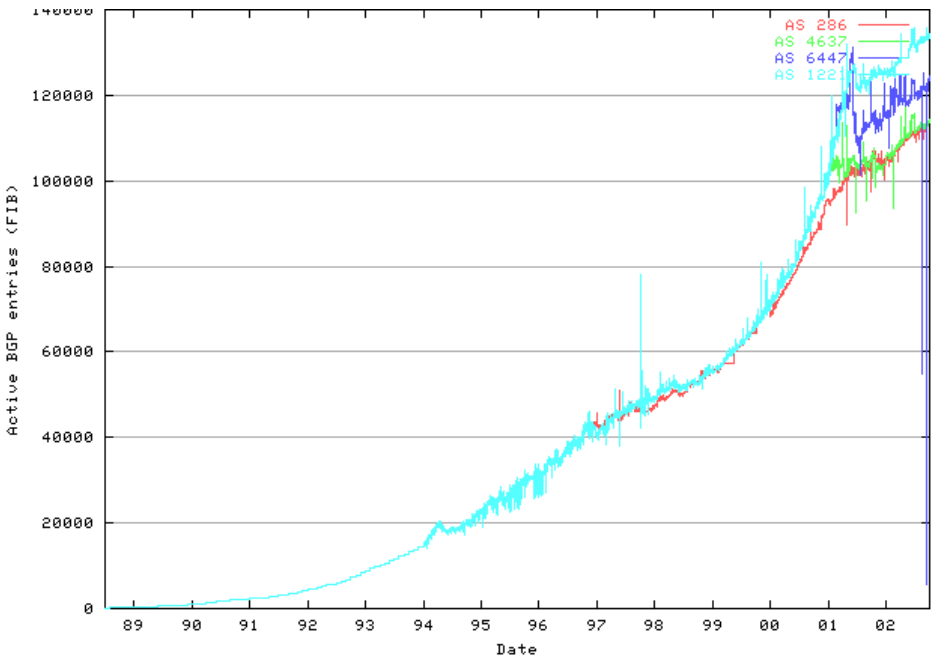
What is Routing?

Routing is the core function of a network

It ensures that

- information accepted for transfer
- at a source node
- is delivered to the correct
- set of destination nodes,
- at reasonable levels of performance.

Routing in the Internet is really about Scaling Range and Number!



Size of core internet router tables v/s time

- Hosts: Number, Mobile, Virtual, P2P
- Traffic: Growth in volume, rates and session types (e.g. unicast, multicast, anycast, voice, video, transactional, bulk)
- Networks: Peering, Information Hiding, Policies, Ad-hoc etc.
- All this and must be “backward compatible” as well!

Why study routing?

- Routing struggles to
  - operate correctly as users, networks and traffic rates increase in range and number while
    - maintaining the distributed, federated nature of the internet
    - providing “acceptable” performance
- Studying routing in the context of this struggle provides rich insights into how current networks work, and how to build better ones

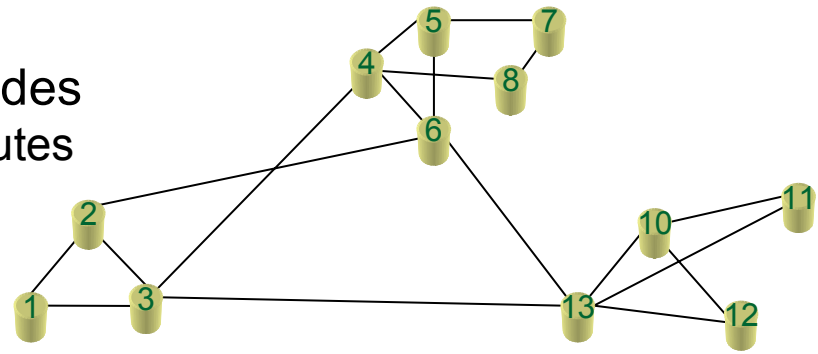
Our Approach

- Lay out the hard issues which are often architectural
- Try to gain insight into the issues via simplified models
- Shoot for insight into the general problem of routing and not spend too much time on the specifics of any one protocol or system

Today

- First part:
  - What are basic sub functions of routing?
  - How should we think about them?
- Second part:
  - How does routing relate to other network functions
  - Path computation - Centralized

Routing Sub-Functions



- **Addressing:** Uniquely identify the nodes
  - host IP address, group address, attributes
  - set is dynamic!
- **Topology Update:**
  - Discover topology
  - Measure “distance” metric(s)
  - Dynamically provision (on slower timescale)
- **Destination Discovery:** Find node identifiers of the destination set
- **Route Computation:** Pick the tree (path)
  - Kind of path: Multicast, Unicast
  - Centralized or Distributed Algorithm
  - Metrics
  - Hierarchy/Policy
- **Switching:** Forward the packets at each node

Routing Protocols

- **Addressing:** Uniquely identify the nodes
  - host IP address, group address, attributes
  - set is dynamic!
- **Topology Update:** Characterize and maintain connectivity
  - Discover topology
  - Measure “distance” (one or more metric)
  - Dynamically provision (on slower timescale)
- **Destination Discovery:** Find node identifiers of the destination set
- **Route Computation:** Pick the tree (path)
  - Kind of path: Multicast, Unicast
  - Centralized or Distributed Algorithm
  - Policy
  - Hierarchy
- **Switching:** Forward the packets at each node

Routing is a distributed function

- Topology changes can be detected by nearby nodes
- These changes must be reflected in the routes
- Need a mechanism and a protocol
- Changes must be disseminated: Three mechanisms
  - Link State: Communicate the names and costs of neighbors. Each node maintains the entire topology. E.g. used in OSPF
  - Distance Vector: Communicate current distance estimates of node to every other node. E.g. used in RIP
  - Path Vector: Communicate current estimates of preferred paths from node to every other node. E.g. used in BGP
- Dissemination protocol may react strangely to network conditions
  - Most routing problems are triggered by this…

Colloquial Routing

- **Addressing:** Uniquely identify the nodes
  - host IP address, group address, attributes
  - set is dynamic!
- **Topology Update:** Characterize and maintain connectivity
  - Discover topology
  - Measure “distance” metric(s)
  - Dynamically provision (on slower timescale)
- **Destination Discovery:** Find node identifiers of the destination set
- **Route Computation:** Pick the tree (path)
  - Kind of path: Multicast, Unicast
  - Global or Distributed Algorithm
  - Policy
  - Hierarchy
- **Switching:** Forward the packets at each node

Many Kinds of Routing

Driven by…

- Destination set
  - IP point-to-point
  - Multicast
- Physical Characteristics
  - Optical
  - Ad-hoc
  - Diffusion (sensor networks)
  - Interconnection network routing
  - Geographic (wireless)
- Network Function
  - P2P
  - Content Distribution Networks

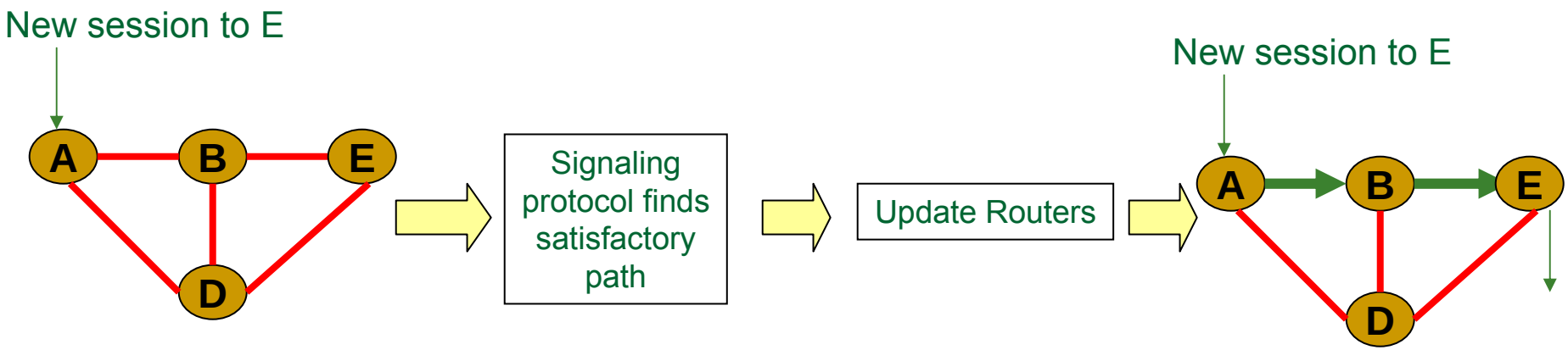
Kinds of Route Computation

- Centralized v/s Distributed
- Hierarchical v/s Flat
- Datagram v/s Virtual Circuit
- Single Path v/s Multipath
- Unicast v/s Multicast
- Shortest Path v/s Policy and Deflection

Datagram v/s Virtual Circuit

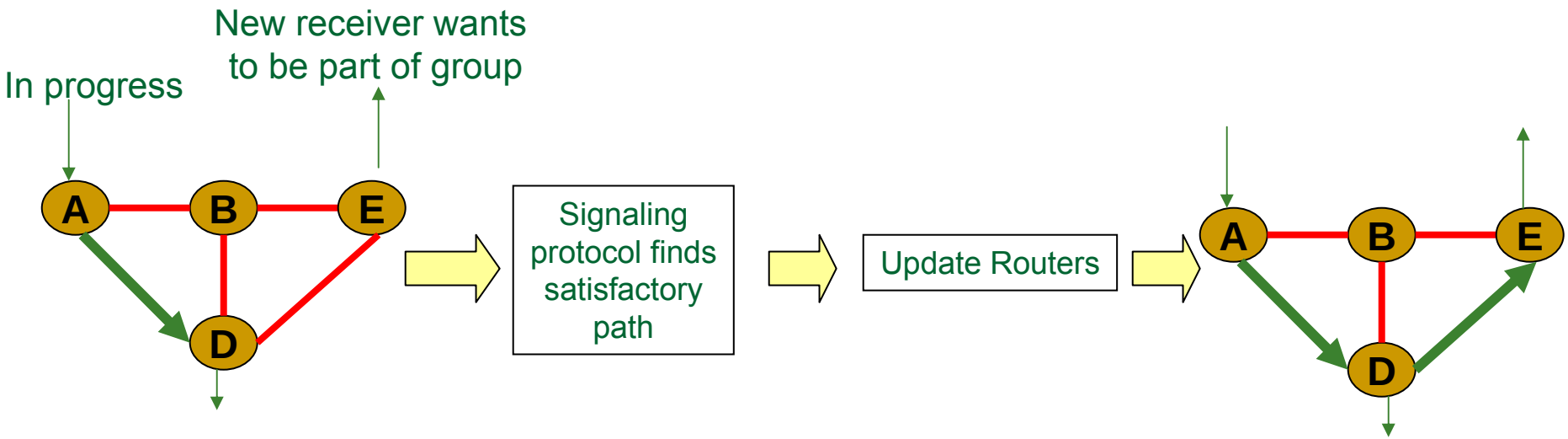
- Datagram routing
  - Each packet to be forwarded independently
  - More sensitive to current network conditions
- Virtual Circuit
  - Each packet from same o-d uses same route
  - More state (pick the “right” granularity)
  - Less sensitive to network conditions
- QoS sensitive networks use VC’s and signaling
  - Find a route that has the resources available for the connection.
  - “Reserve” the resources before sending data packets

The reservation process: Sender



- Ability to guarantee e2e delay helps in signaling and resource reservation
- Service disciplines and other network QoS mechanisms affect the routes that are chosen
- Signaling is hard in peered networks

The reservation process: Receiver



- Ability to guarantee e2e delay helps in signaling and resource reservation
- Service disciplines and other network QoS mechanisms affect the routes that are chosen
- Signaling is hard in peered networks

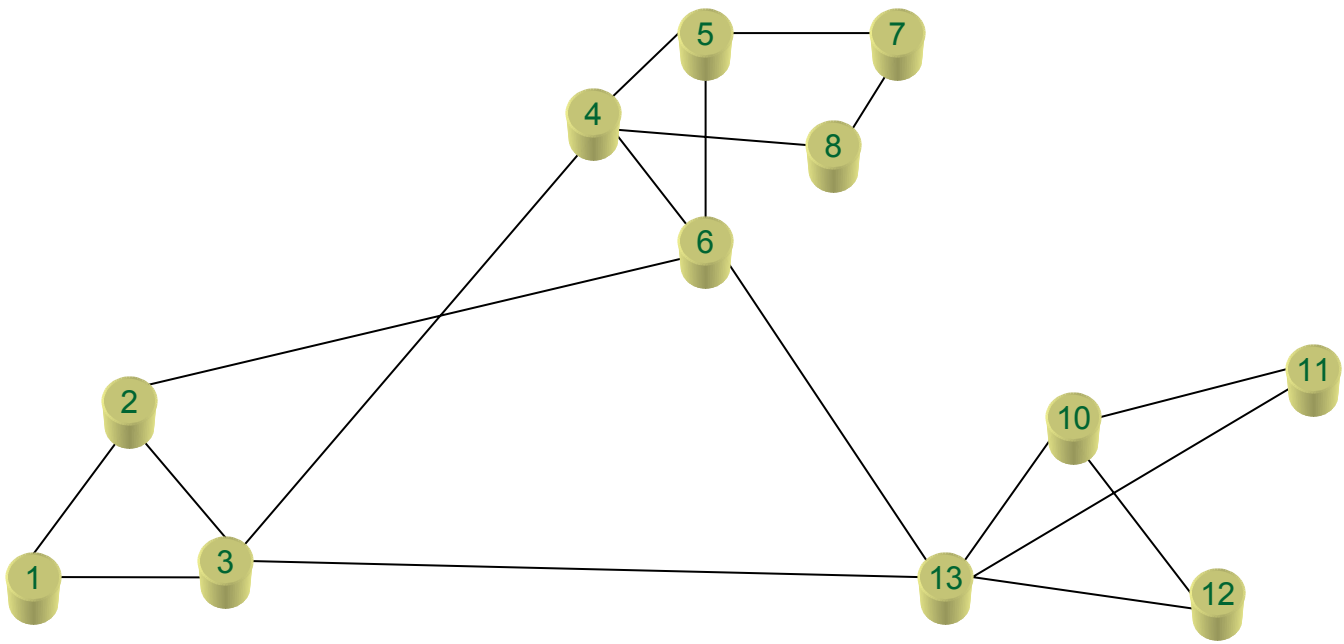
Mostly Ignored…

- **Addressing:** Uniquely identify the nodes
  - host IP address, group address, attributes
  - set is dynamic!
- **Topology Update:** Characterize and maintain connectivity
  - Discover topology
  - Measure “distance” metric(s)
  - Dynamically provision (on slower timescale)
- **Destination Discovery:** Find node identifiers of the destination set
- **Route Computation:** Pick the tree (path)
  - Kind of path: Multicast, Unicast
  - Global or Distributed Algorithm
  - Policy
  - Hierarchy
- **Switching:** Forward the packets at each node

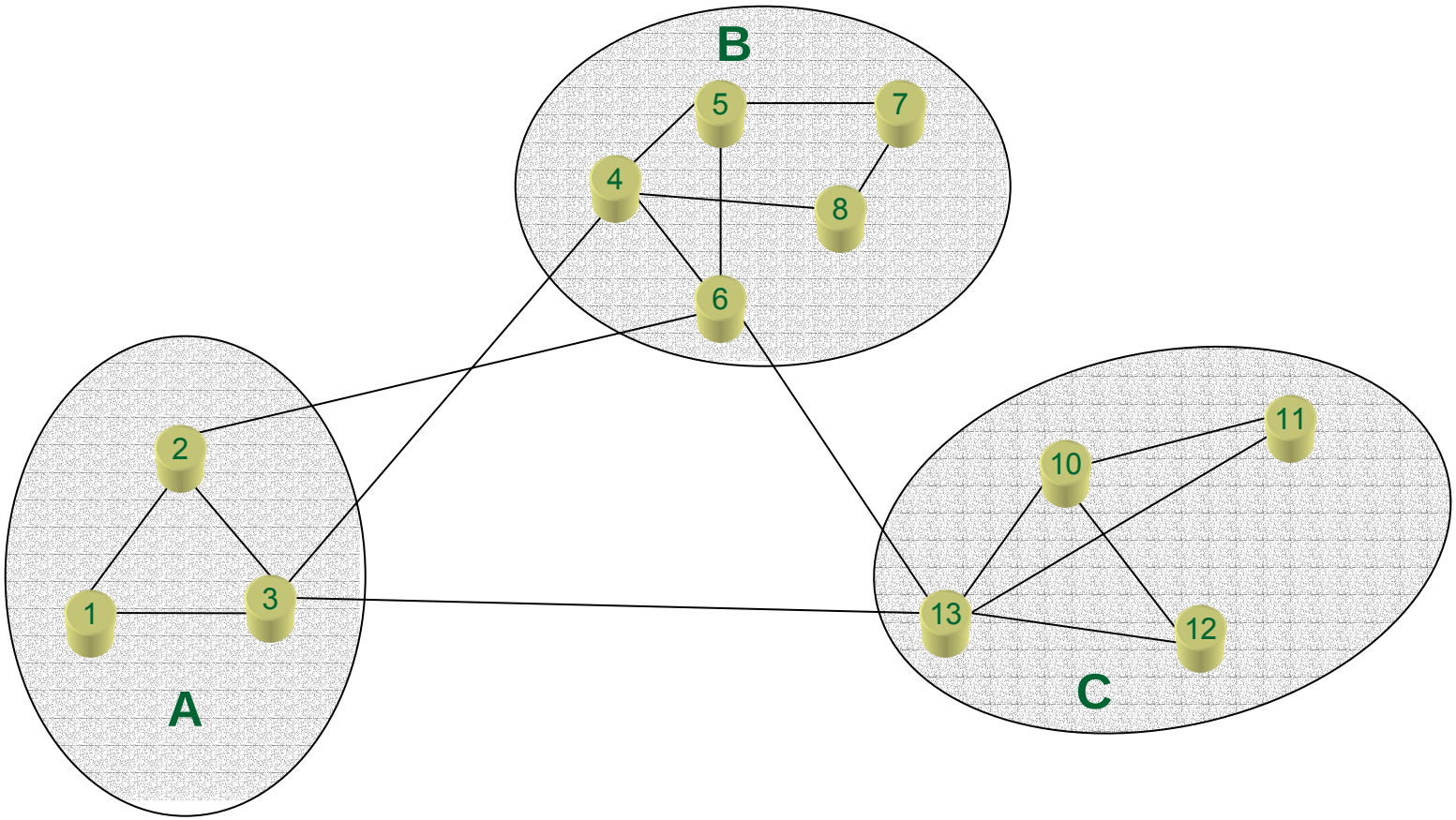
Addressing

- Why not a flat address space?
  - Just use node id's like MAC addresses
  - Can't aggregate – routing table is huge O(N)
- Addressing structure and allocation are crucial to scaling the internet
  - Allocation is not geographic (within a country) but topological
- Major shifts implemented to scale
  - Class-based to Classless to IPV6
- What if end-devices didn't have IP addresses?

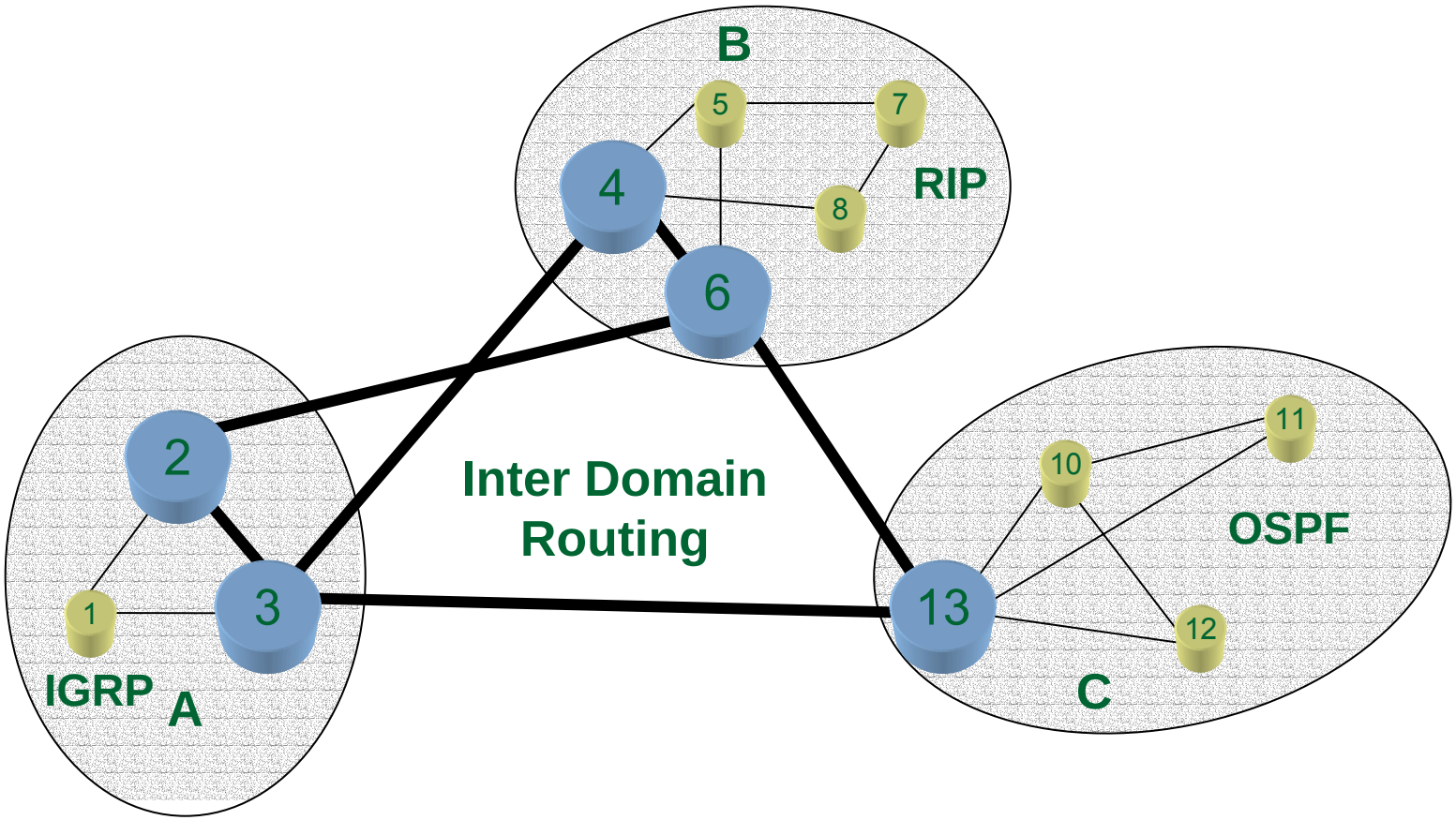
Flat Network



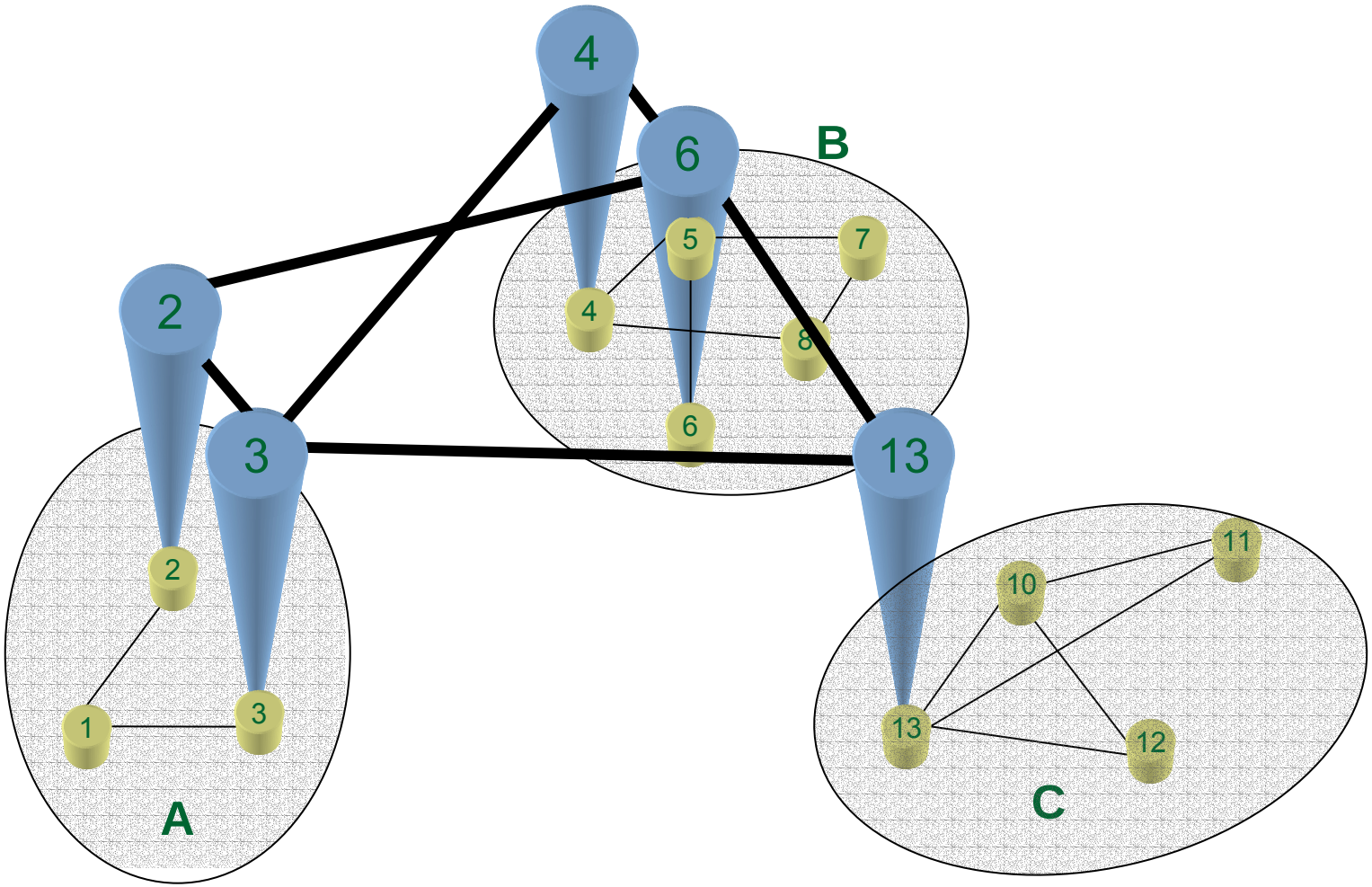
Routing Interconnects AS's



Border Routers



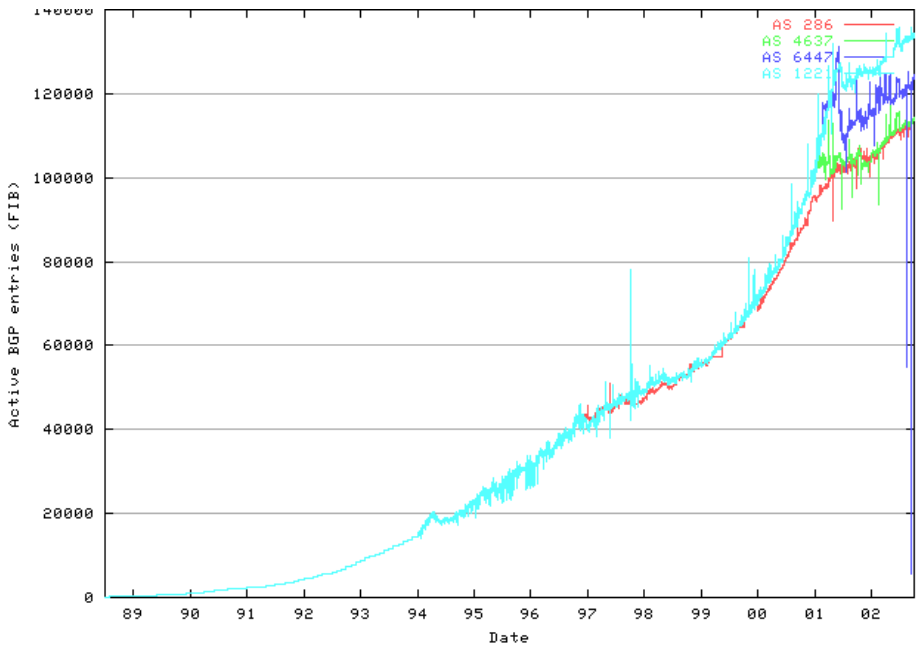
Hierarchical Routing

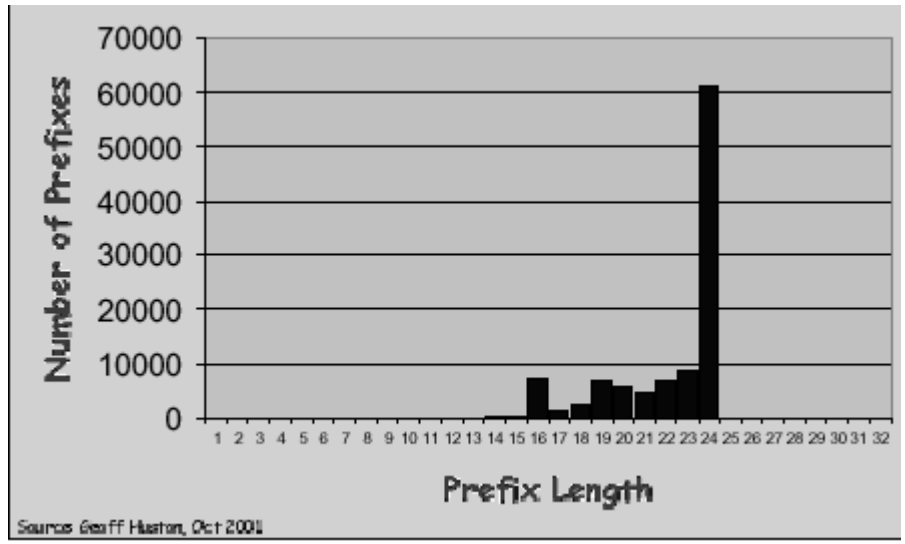


Classless Inter-domain Routing Addresses

- 32 bits in the address divided into 4 8-bit parts, A.B.C.D
  - Each part takes value 0,1,2,…,255
    - E.g. 128.23.9.0
- Specify a range of addresses by a prefix: X/Y
  - The prefix common to the entire range is the first Y bits of X.
  - X: The first address in the range has prefix X
  - Y: 2^{32-Y} addresses in the range
- Example 128.5.10/23
  - Common prefix is 23 bits: 01000000 00000101 0000101
  - Number of addresses: $2^9 = 512$
- Prefix aggregation
  - 128.5.10/24 and 128.5.11/24 gives 128.5.10/23
- Addresses allocated by central authority: IANA
- Routers match to longest prefix

BGP Routing Table Scaling





- Many small networks
- Aggregation hides a lot…

Destination address discovery

- DNS converts a name to an IP-address
  - Geographically distributed database

But….very often sender has no idea where the packets will end up!

- Middleboxes: Firewalls, Load Balancers, NAT
- Multicast
  - Receiver joins a group (IP address) and sender sends to the group.
  - Recipient addresses resolved at the router
- Content Distribution Networks and P2P
  - Content replicated based on demand – must be discovered
  - Proxmity-based redirection
- **Currently, this part of the internet is a mess!**
  - This is can be a “search” problem or an “allocation” problem
  - …

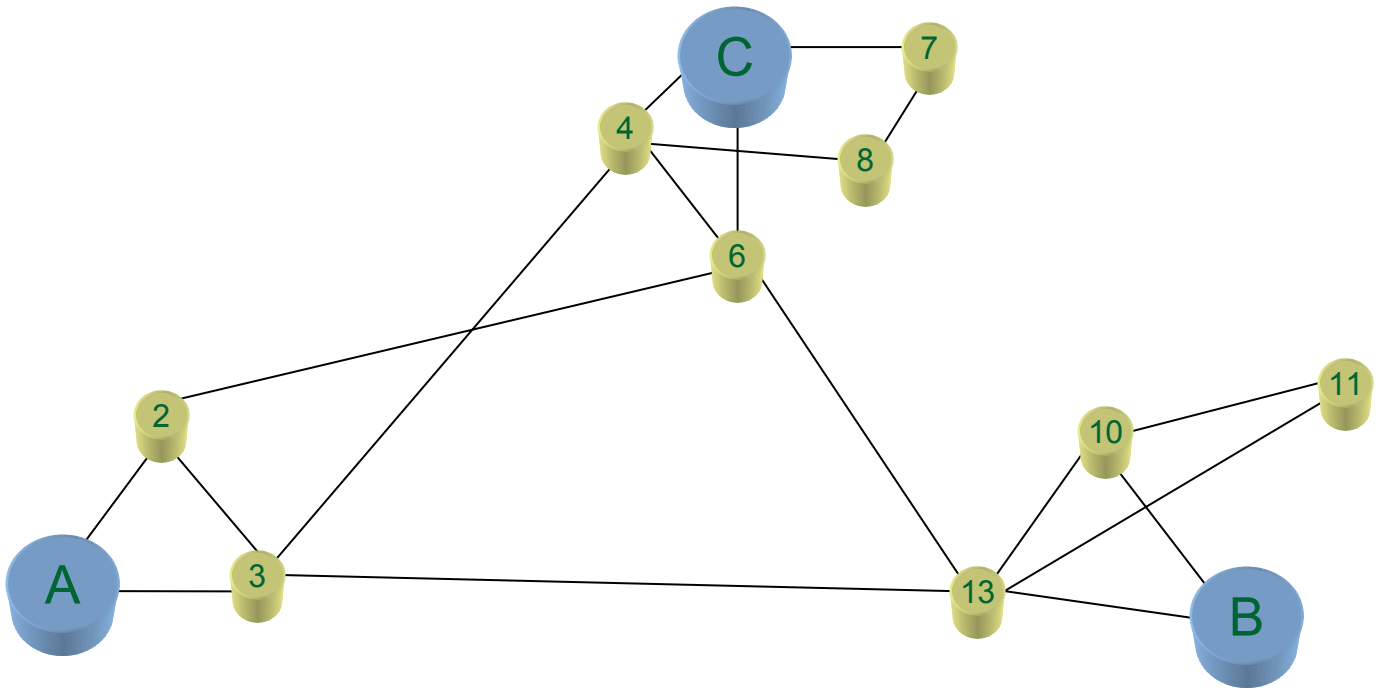
Router Design

- **Addressing:** Uniquely identify the nodes
  - host IP address, group address, attributes
  - set is dynamic!
- **Topology Update:** Characterize and maintain connectivity
  - Discover topology
  - Measure “distance” metric(s)
  - Dynamically provision (on slower timescale)
- **Destination Discovery:** Find node identifiers of the destination set
- **Route Computation:** Pick the tree (path)
  - Kind of path: Multicast, Unicast
  - Global or Distributed Algorithm
  - Policy
  - Hierarchy
- **Switching:** Forward the packets at each node

Packet Forwarding

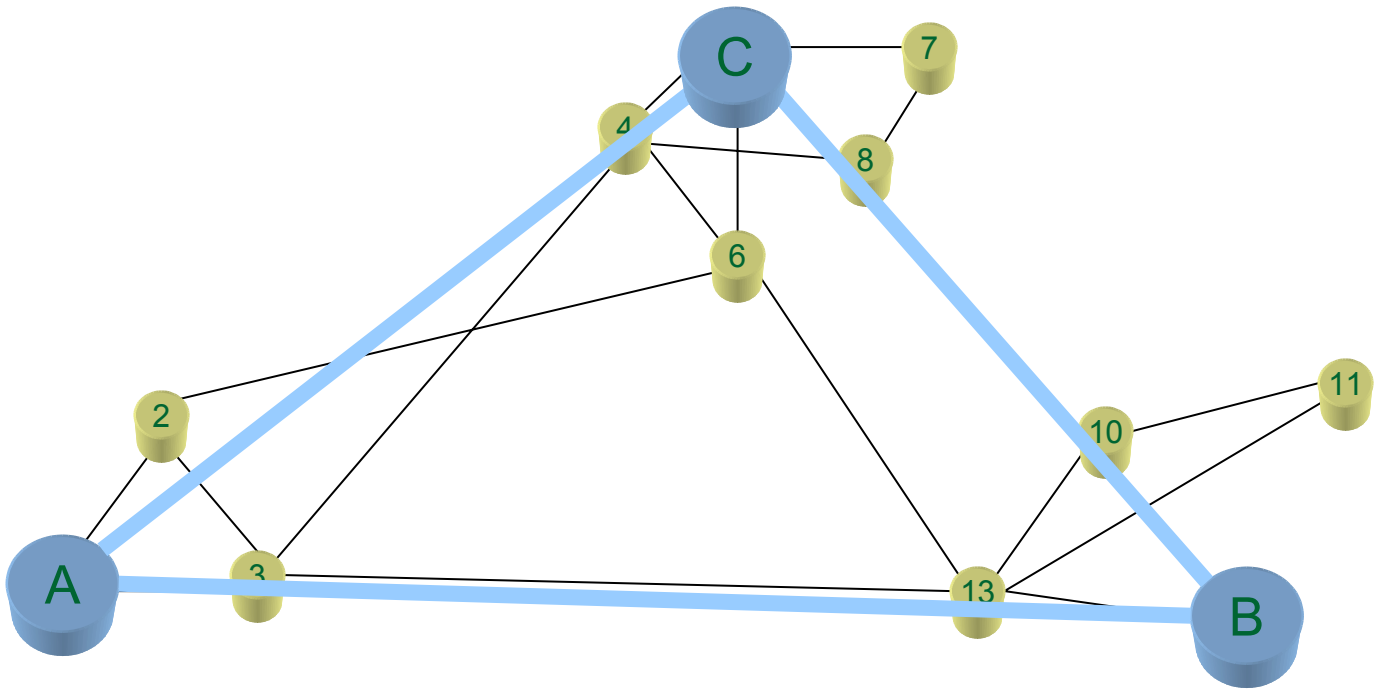
- For a datagram network
  - Routing table has prefixes and port numbers
  - Given a packet with destination, D, find the entry with the longest prefix match and send the packet to the corresponding port
  - Many algorithms for fast prefix matching!
- For a virtual circuit network
  - State includes connection identifiers which map to port numbers
  - Design somewhat complicated based on what kinds of QoS mechanisms exist.
- Qos/Class-based networks
  - Input v/s Output Queueing
  - Service Disciplines
    - Per-connection
    - Per-class

Layers of Routing



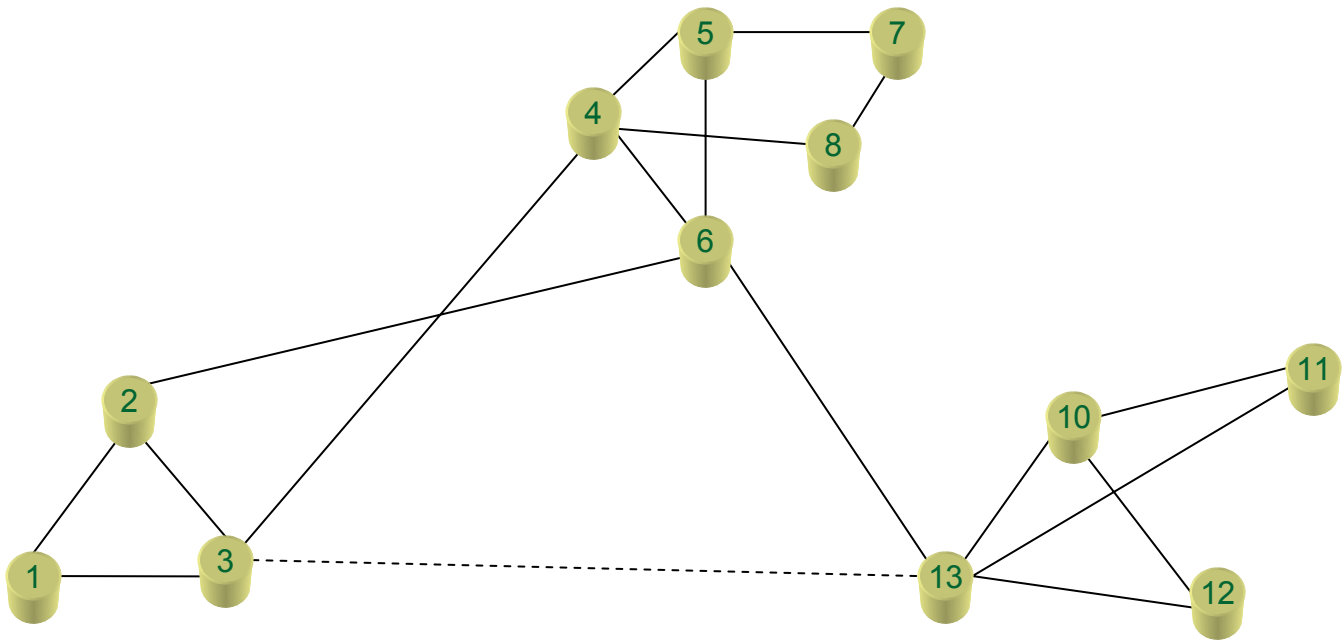
Overlay Network Nodes

Layers of Routing



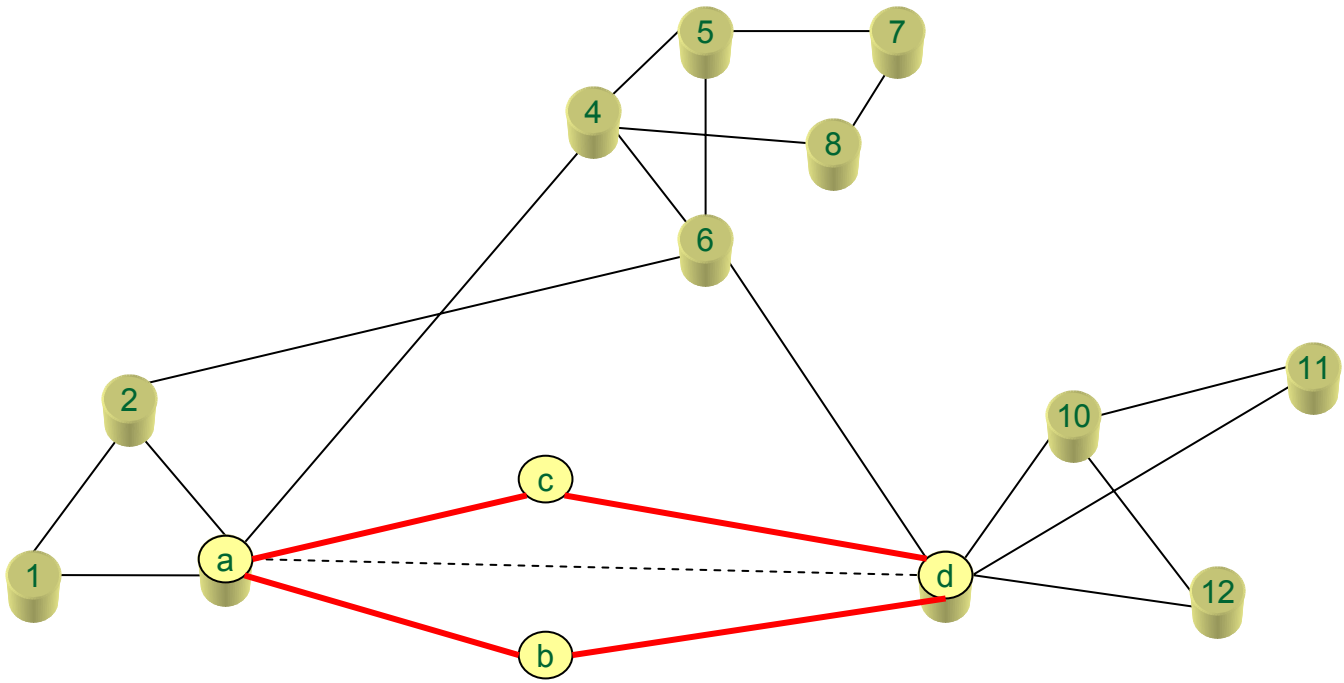
- Overlay Networks are extremely popular
- MBONE, Akamai, Virtual Private Networks
- Overlay Networks may even peer!

Layers of Routing



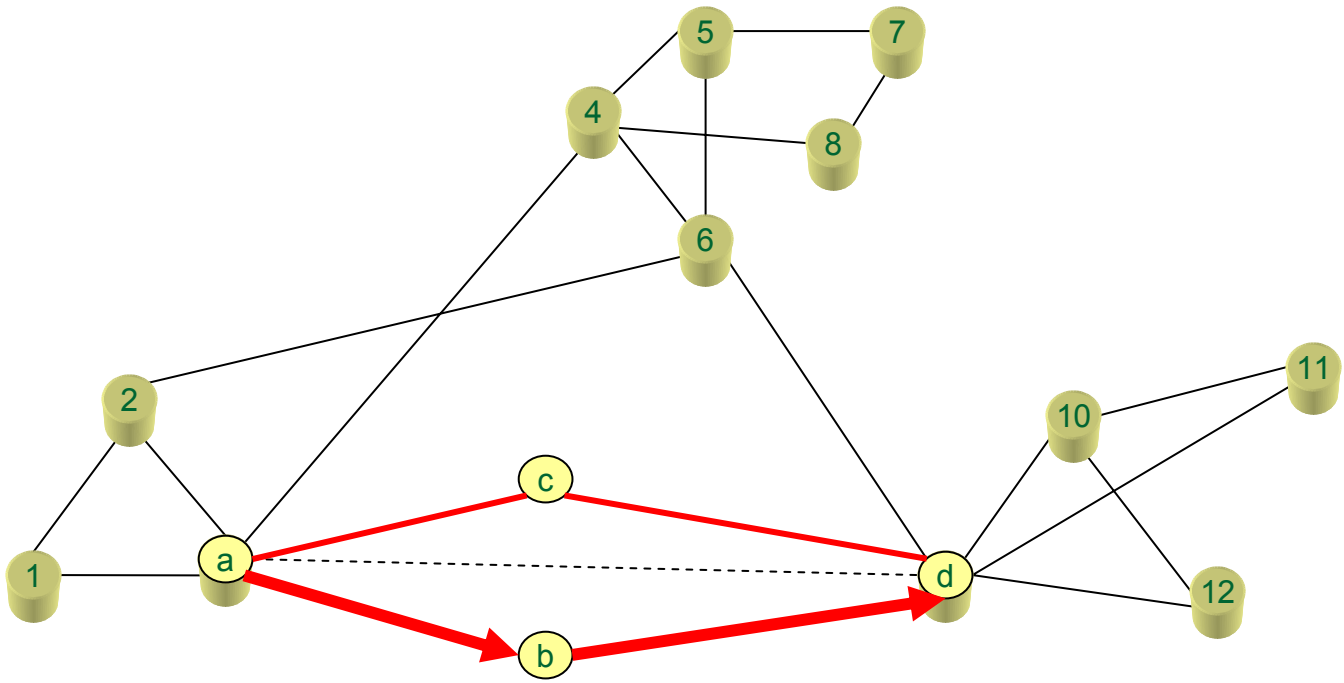
IP network can be an overlay itself!

Layers of Routing



ATM links can be the “physical layer” for IP

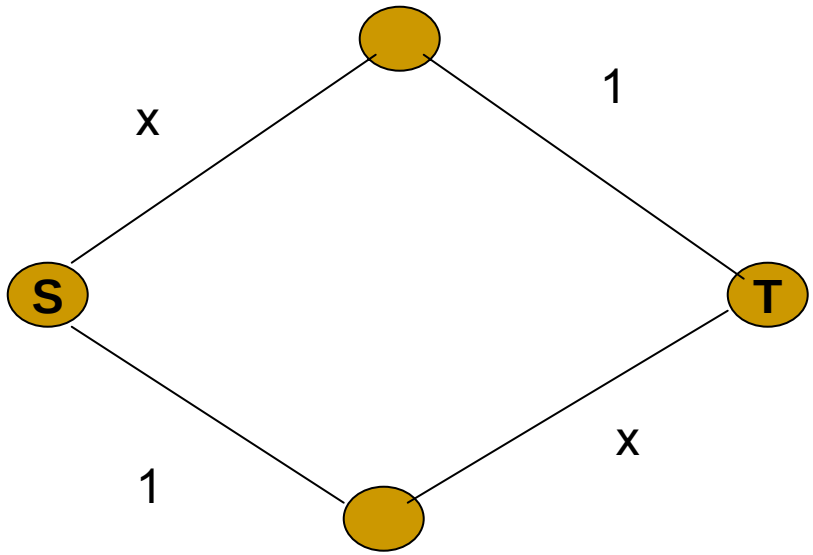
Layers of Routing



Virtual Circuit under Datagram!

Let’s look at simple models…

Why isn't route computation easy?



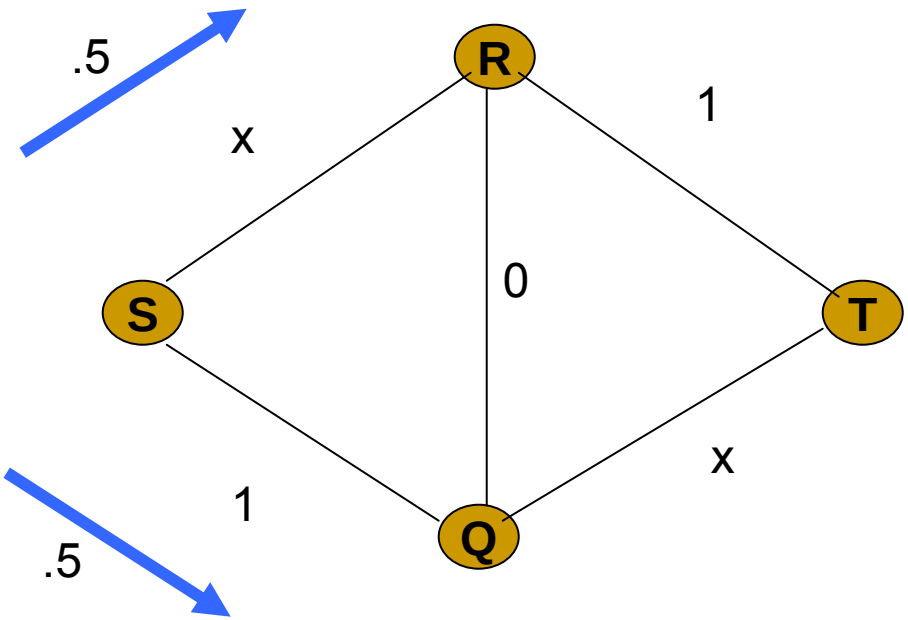
Weights are delays in hours

1 unit of traffic from s to t

If u is the amount of traffic on the upper route
Total delay = u(u+1) + (1-u)(2-u) = 2(u^2 – u +1)
Delay minimized at u=.5
Each bit is delayed 1.5hrs

Routing is Counter-Intuitive

Nash Equilibrium



Weights are delays in hours

1 unit of traffic from s to t

BRAESS'S PARADOX

D(S,Q)=x <= 1 via new link, and
R diverts all bits on to the new link

*Now each bit is delayed by **2** hours!*
This is the only stable operating point!

What is going on?

- Distributed Scheme acts selfishly to maximize a local objective function
  - This can result in globally suboptimal outcomes
  - Local knowledge can be outdated/incorrect
- Forwarding decisions affects link costs, which affect forwarding decisions. This can result in instability
- The distributed and dynamic nature of the underlying network makes “high performance” routing hard

Interesting Results (Roughgarden and Tardos)

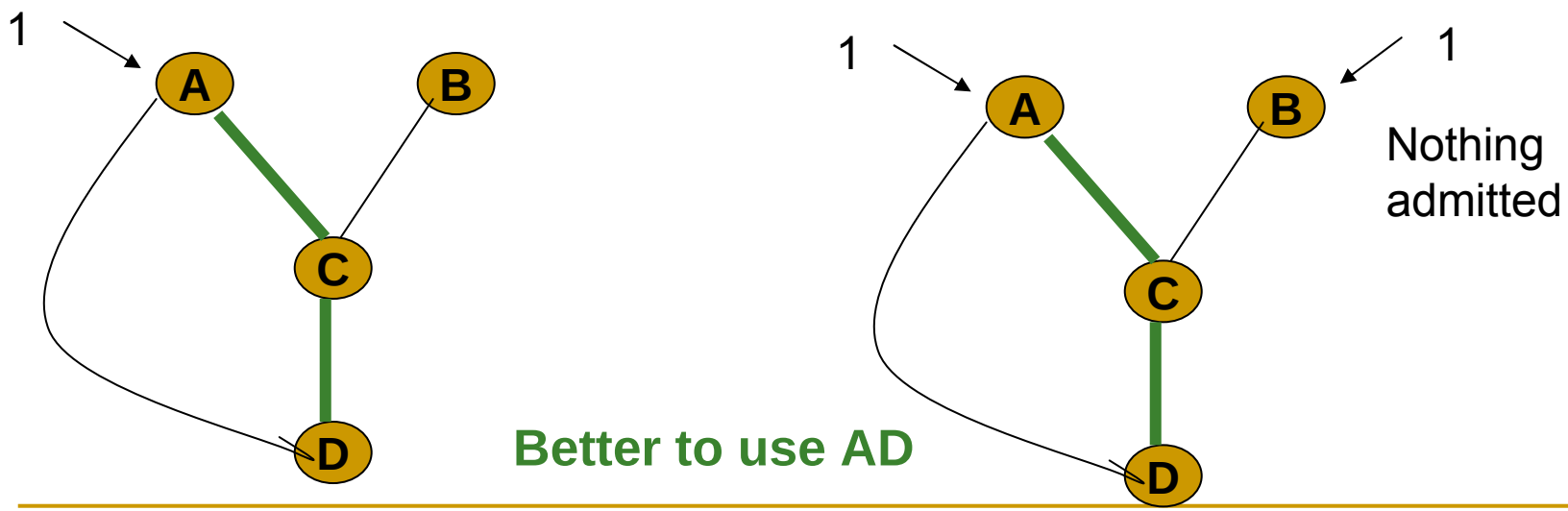
- Adding edges to N vertex graph increases common delay by at most an N/2 factor
- For any network with linear latency functions of the form $\ell_e(x)=a_e x+b_e$

  Cost of Nash flow ≤ 4/3 Cost of opt. Flow
- For all cont, non-decreasing latency functions

  C(Nash at rate r) ≤ C(Optimal at rate 2r)

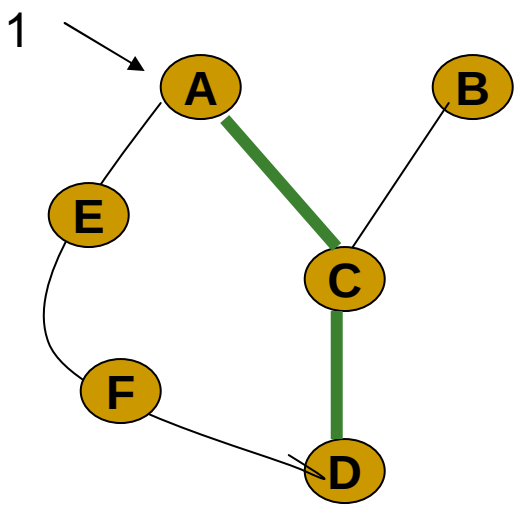
Flow Control and Routing

- Any function that "paces" the flow of bits into or within the network is flow control
  - Not just end-to-end!
- Example: All links have capacity 1

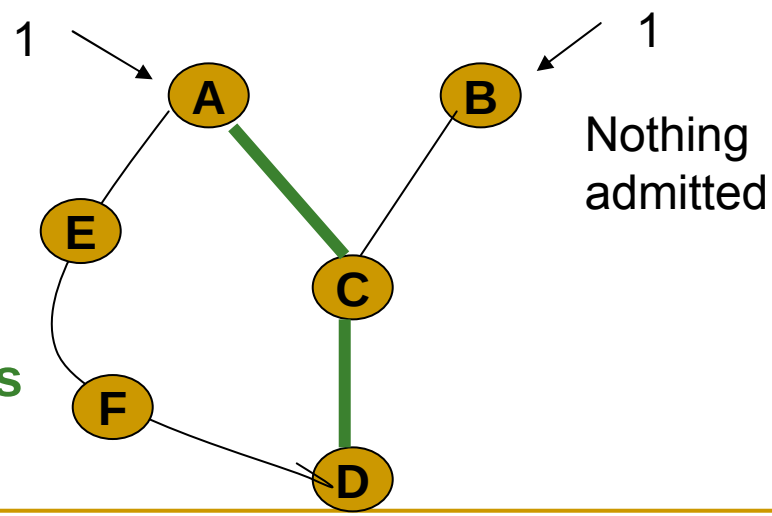


Flow Control and Routing

- Any function that "paces" the flow of bits into or within the network is flow control
  - Not just end-to-end!
- Example: All links have capacity 1



Min-hop does not help



What is going on?

- Routing and flow intimately related
  - Link congestion metrics for routing depends on flow control
  - Flow control feedback delay depends on routing
- Optimizing them jointly is nice in theory but intractable in practice.
- Separating flow control and routing makes both extremely difficult to implement with high performance
  - Routing metrics change unpredictably
  - Overall Network utilization remains low
- What if the problem is “static” or “one shot”?

"Optimal" Routing

- **x_p is the amount of flow on path p.**
- **F_{ij} is the abound of flow on link (i,j).**

r_w: o-d flow
Multipath allowed

$$\min \sum_{(i,j)} D_{ij}(F_{ij})$$

subject to

$$\sum_{p \in P_w} x_p = r_w, \qquad \forall w \in W$$

$$x_p \geq 0, \qquad \forall p \in P_w, w \in W$$

Can we characterize the optimal solution?

“Optimal” Routing

- Suppose we are given an optimal flow path flow vector, x*.
- Given paths p, p’ that connect the same o-d pair, if Δ units of flow were shifted from p to p’ the cost should not go down, i.e.

$$\frac{\partial D(x^*)}{\partial D(x_{p'})} - \frac{\partial D(x^*)}{\partial D(x_p)} \geq 0$$

so that

$$x_p^* > 0 \Rightarrow \frac{\partial D(x^*)}{\partial D(x_{p'})} \geq \frac{\partial D(x^*)}{\partial D(x_p)}$$

- This condition is sufficient as well when D_{ij} is convex!
- Under optimal routing flow is positive only on paths with min first derivative length!

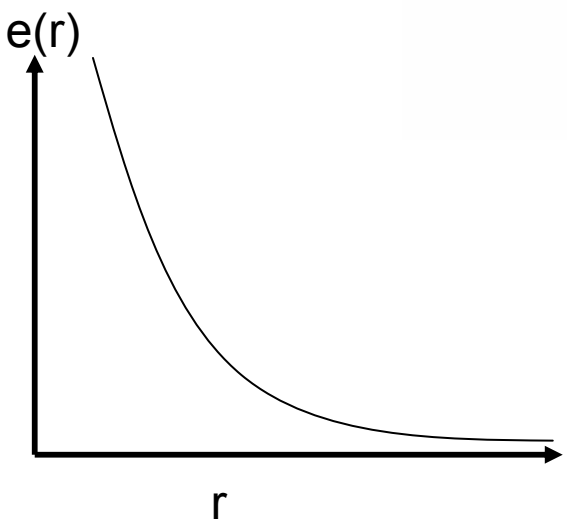
Joint Routing and Flow Control

$$\min \sum_{(i,j)} D_{ij}(F_{ij}) + \sum_{w \in W} e_w(r_w)$$

subject to

$$\begin{aligned} \sum_{p \in P_w} x_p &= r_w, && \forall w \in W \\ x_p &\geq 0, && \forall p \in P_w, w \in W \\ 0 &\leq r_w \leq \bar{r}_w, && \forall w \in W \end{aligned}$$

- r variable
- penalty fxn

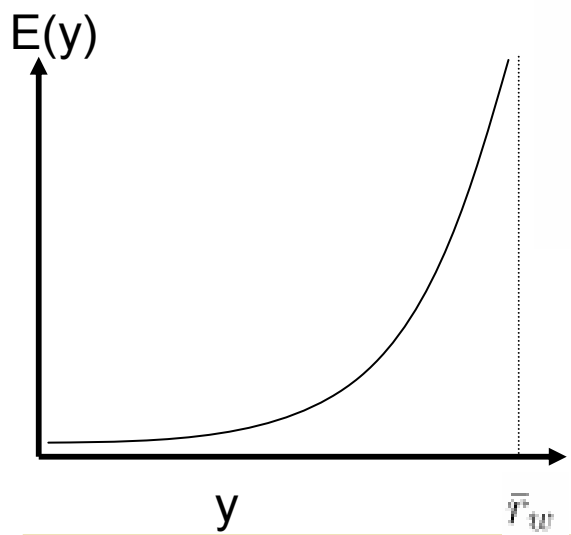


"Overflow" is $y_w = \bar{r}_w - r_w$

Recast as a Routing Problem

- y flow on overflow link
- penalty fxn flipped to be link cost

$$E_w(y_w) = e_w(\bar{r}_w - y_w)$$



$$\min \sum_{(i,j)} D_{ij}(F_{ij}) + \sum_{w \in W} E_w(y_w)$$

subject to

$$\begin{aligned} \sum_{p \in P_w} x_p + y_w &= \bar{r}_w, && \forall w \in W \\ x_p &\geq 0, && \forall p \in P_w, w \in W \\ y_w &\geq 0, && \forall w \in W \end{aligned}$$

Solve routing problem as before

Golestani 1980

This isn't used in practice

- Static solution
- Too many paths!
- Hard to compute – even analytically
- Quite useful to get insight
  - When flow control is employed there one should think of trading off the penalty against the increase in network performance for the rest of the traffic
  - If routing performance depends on congestion, then think about the sensitivity to changes in link flow rather than just absolute amounts of flow

Connectivity

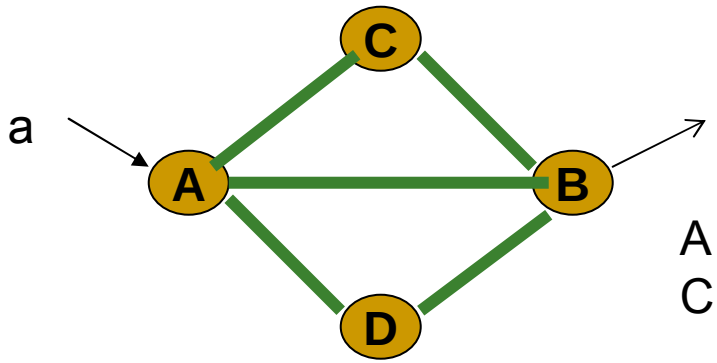
- What's an edge?
  - A physical link, a part of a link or several links?
- What's a link cost?
  - How many metrics?
- Topology broadcast
  - Very tricky since it can't rely on routing tables and has to respond to changes quickly

Edge Costs

- Traffic sensitive
  - $$D_{ij}(F_{ij}) = \frac{F_{ij}}{C_{ij} - F_{ij}}$$

  M/M/1 Queue delay
- Number of Hops
- How to represent multiple objectives?
  - E.g. delay **and** cost
  - Single Metrics: e.g. widest path routing
  - Mutliple Metrics: e.g. delay and buffer space
  - What if no combination of objectives can be agreed to…Policy Routing

Policy Routing



P1 = AB
P2 = ACB
P3 = ADB

A and B rank path preferences
C and D don't care

Routing algorithm tries the paths according to some order

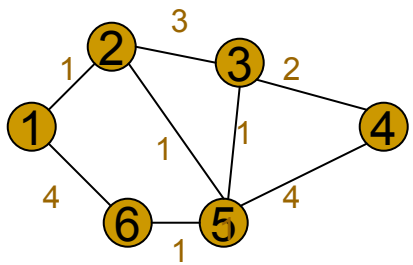
There is no routing algorithm that can rank the paths based on the preferences of A and B that respects unanimity, pair-wise independence and that is non-dictatorial

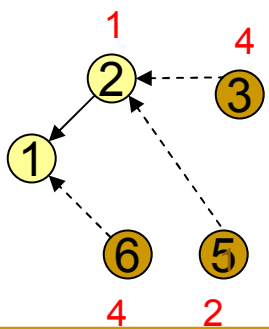
Arrow's impossibility theorem

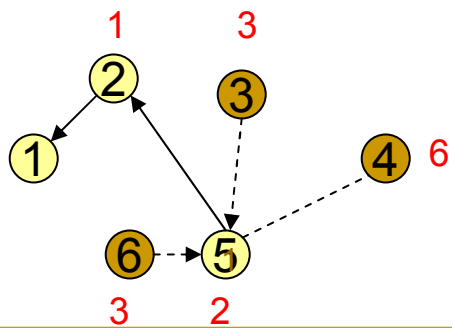
What if routing algorithm just has to pick one path?

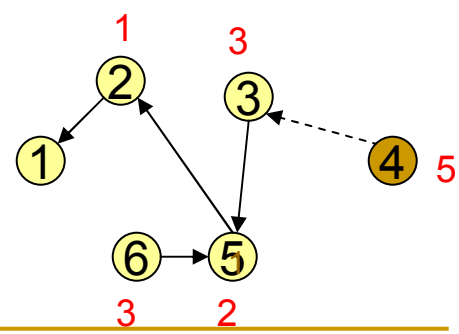
Dijkstra: Shortest Path

- All link costs are >= 0
- Find the distance from 1 to all the other nodes in order of increasing path length:
  - N iterations: Start at node 1 and label one node in each iteration. Set of labeled nodes is P. Know the shortest paths from 1 to every node in P.
  - In iteration k find a node, α, one hop away from P and that is closest to 1. This node must the closest distance away from 1 of all nodes not in P.







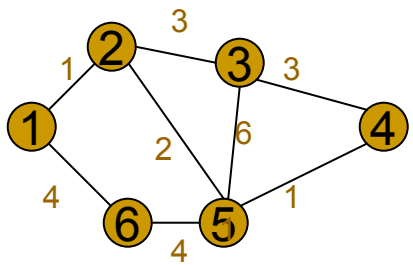


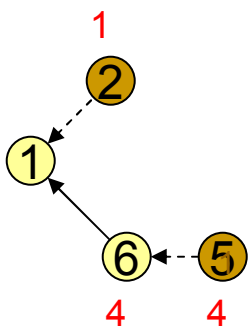
Generalizing Shortest Path

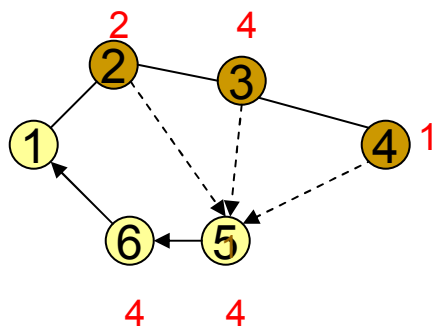
- Two operations
  - Find the weight of a path i.e. + over the edges
  - Rank the paths, i.e. <
- Example: Find the Widest Shortest Path
  - Edge weights: (distance, capacity) or (d,b)
  - Path weight: (sum distance, min capacity)
  - Path Ranking: d1 < d2 or d1=d2, b1>= b2
- How much can Dijkstra be generalized?

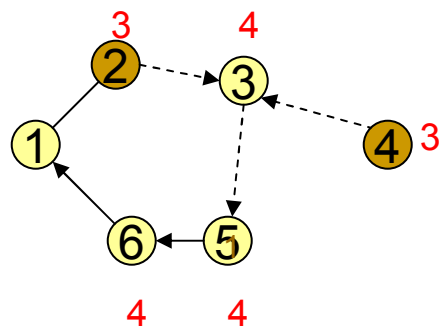
Dijkstra: Widest Path

- Path weight is min(edge wts); best path has largest weight
- Find the distance from 1 to all the other nodes in order of increasing path length:
  - N iterations: Start at node 1 and label one node in each iteration. Set of labeled nodes is P. Know the widest paths from 1 to every node in P.
  - In iteration k find a node, α, one hop away from P and that is widest to 1. This node must the widest distance away from 1 of all nodes not in P.





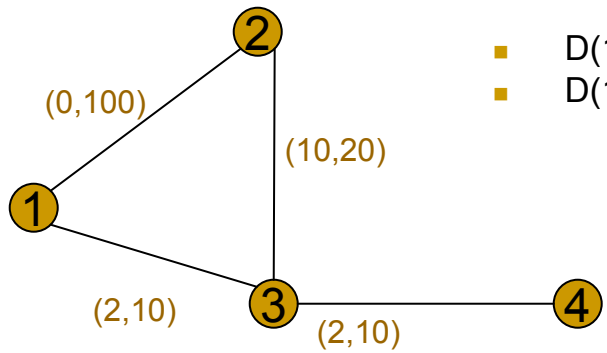




How General is Gen Dijkstra?

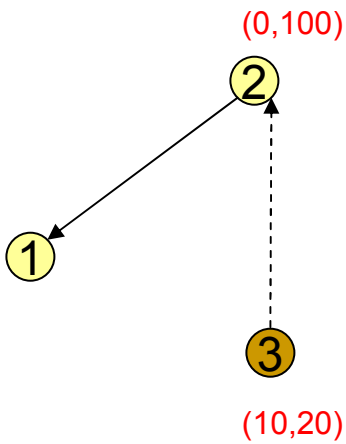
- S-lightest path from s to v: Every sub-path beginning with s is a lightest path
- Generalized Dijkstra works iff for every connected s-v pair there is a lightest path from s to v.
- Property stems from an underlying algebra:
  - Isotonicity: for all possible edge weights a,b,c
    - a <= b implies
      - a+c <= b+c
      - c+a <= c+b
- Sobrinho (IEEE Trans Networks 9/2002)

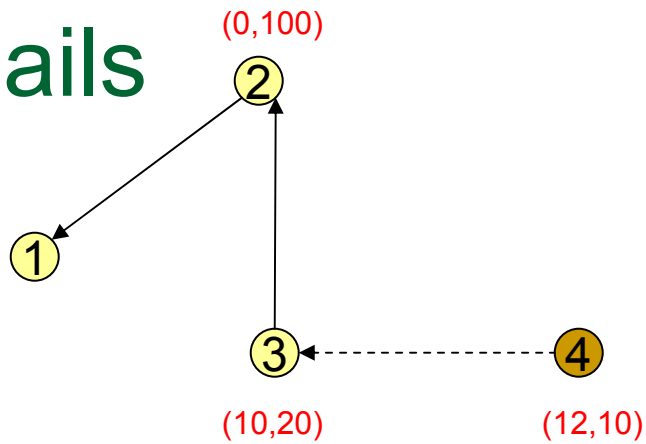
Shortest Widest Path



- D(1,3,4) = (2,10)+(2,10) = (4,10)
- D(1,2,3,4) = (0,100)+(10,20)+(2,10) = (12,10)

Dijkstra Fails





t

What if metrics can't be combined

- Example: Cost and Delay
- 2-constrained routing is NP complete
- Many heuristics suggested

- Currently, poorly understood.

Summary

- Routing is a multilevel, multilayered function
- Ignoring the dependence of routing performance on network design, flow control and link utilization allows us to make progress algorithmically, but may not result in good routing
- Multiple metrics are hard to accommodate
- Conclusion
  - Very difficult to use the routing function to squeeze better performance out of a network
  - That's why it is fun to try!

Moving ahead…

Week 7 Routing	October 7	Overview	Routing Reading
	October 9	Distributed Route Computation	
Week 8 Routing	October 14	Presentations on Readings 1,2,3	
	October 16	Hierarchical Routing	
Week 9 Routing	October 21	Guest Lecture: NetVMG	
	October 23	Presentations on Readings 4,5,6	
Week 10 Routing	October 28	Overlay and P2P routing	
	October 30	Presentations on Readings 7,8,9	

Readings for Student Presentations

1. Tim Roughgarden and Eva Tardos, How Bad is Selfish Routing, IEEE Symposium on Foundations of Computer Science 2000.
2. Frank Kelly, Routing in Stochastic Networks, Stochastic Networks, The IMA Volumes in Mathematics and its Applications, 71.
   Springer-Verlag, New York. 1995. 169-186.
3. Jinyang Li, John Jannotti, Douglas S. J. De Couto, David R. Karger, Robert Morris, "A Scalable Location Service for Geographic Ad Hoc Routing"

4a. John Hershberger and Subhash Suri. "Vickrey Prices and Shortest Paths: What is an edge worth?" **FOCS-2001**: 42nd Annual Symposium on Foundations of Computer Science

4b. J. Feigenbaum, C. Papadimitriou, R. Sami, and S. Shenker, "A BGP-based Mechanism for Lowest-Cost Routing," in *Proceedings of the 21st Symposium on Principles of Distributed Computing, 2002.*

[Papers 4a and 4b are to be presented together.]

5. Labovitz et. al, "The Impact of Internet Policy and Topology on Delayed Routing Convergence"
6. Lixan Gao, Timothy Griffin and Jennifer Rexford, "Inherently Safe Backup Routing with BGP," Infocom 2001.
7. Stoica et al. "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications", To Appear in the IEEE Transactions on Networking
8. Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, **On the Constancy of Internet Path Properties**, Proc. ACM SIGCOMM Internet Measurement Workshop, November 2001.
9. Savage, Collins and Hoffman "The Effects of End-to-End Internet Path Selection", Sigcomm '99

How to present the papers

1. Make sure you state the problem clearly
2. Minimize notation (you can change it!)
3. State assumptions clearly
4. Explain results with “intuitive” proofs
5. State the best and worst aspects of the paper
6. Don’t shoot for completeness but clarity
7. If possible, describe the coolest ideas in the paper