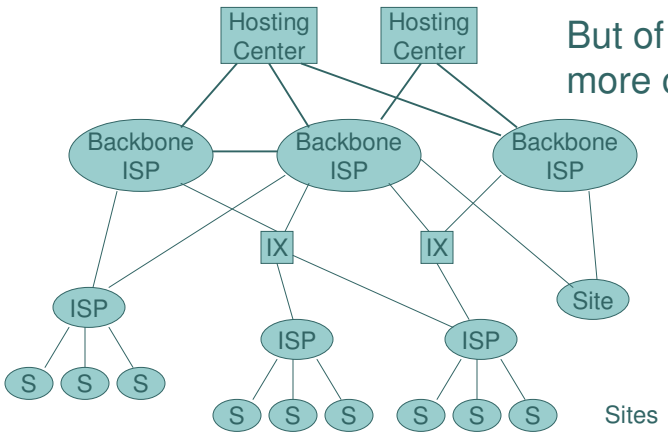
CS514: Intermediate Course in Computer Systems

Lecture 17: February 26, 2003
"Internet Routing"

Revisit Internet topology (from lecture 3)

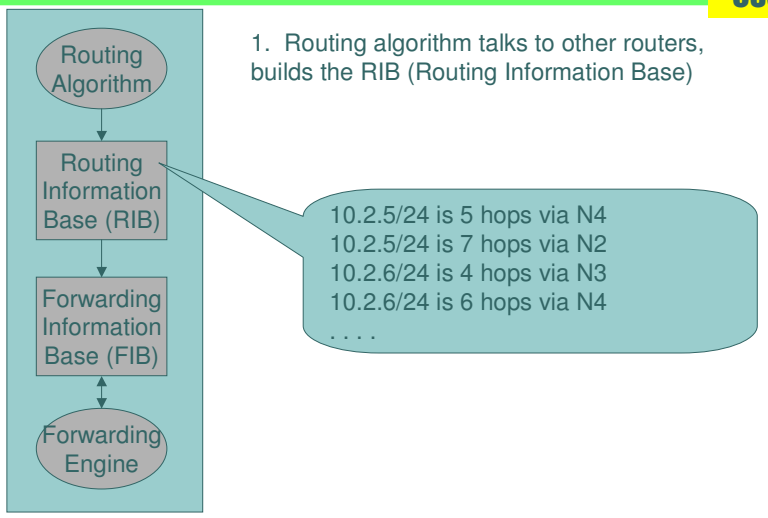
CS514



But of course way more complex

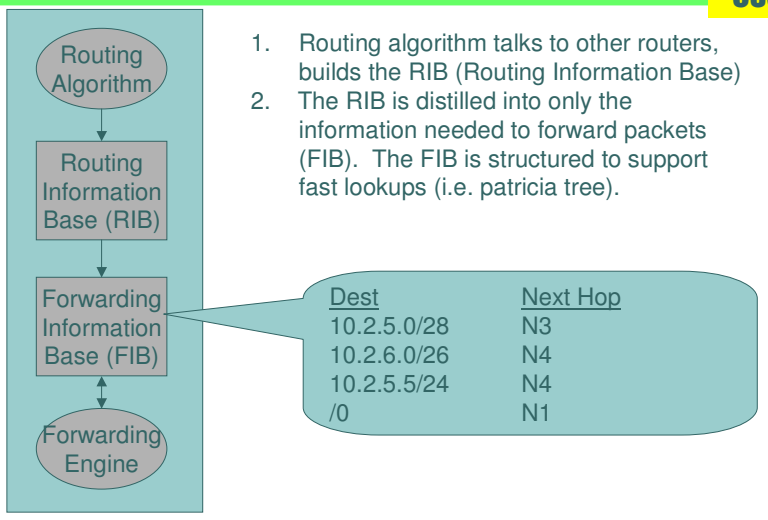
Basic Router Operation

CS514



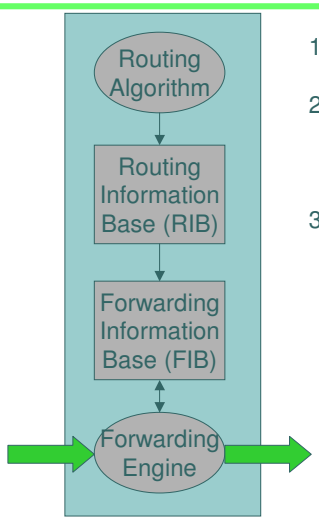
Basic Router Operation

CS514



Basic Router Operation

CS514



1. Routing algorithm talks to other routers, builds the RIB (Routing Information Base)
2. The RIB is distilled into only the information needed to forward packets (FIB). The FIB is structured to support fast lookups (i.e. patricia tree).
3. The Forwarding engine does a FIB lookup for every packet received.

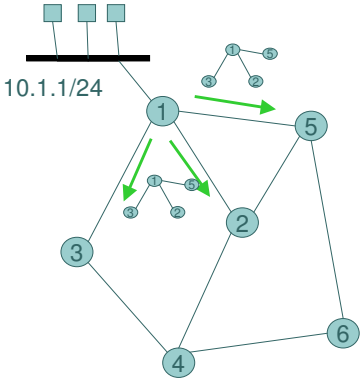
Types of Routing algorithm

CS514

- Two basic kinds
  - Link-state ("map-based")
    - OSPF, IS-IS
      - (Open Shortest Path First), (Intermediate-System to Intermediate-System)
  - Distance-vector ("rumor-based")
    - BGP, RIP
      - (Border Gateway Protocol), (Routing Information Protocol)

Link State Routing

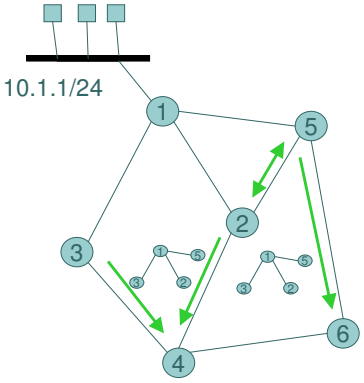
CS514



1. Each node tells its neighbors what networks it has, and who its neighbors are (link-state update).

Link State Routing

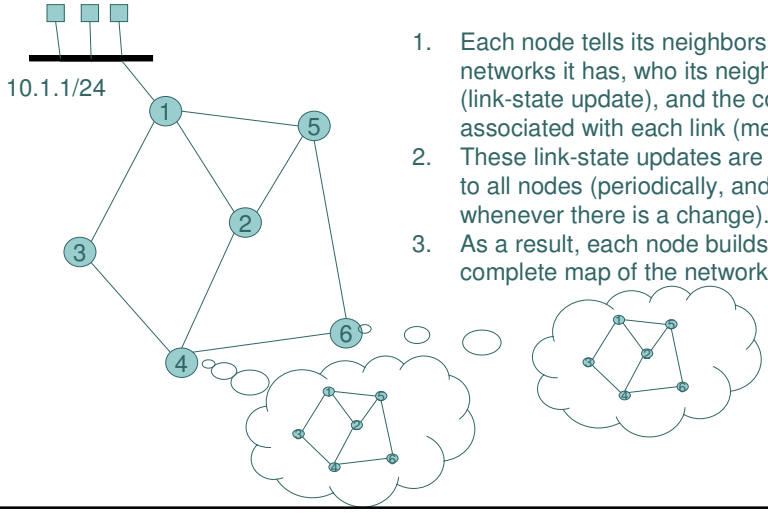
CS514



1. Each node tells its neighbors what networks it has, who its neighbors are (link-state update), and the cost associated with each link (metric).
2. These link-state updates are flooded to all nodes (periodically, and also whenever there is a change).

Link State Routing

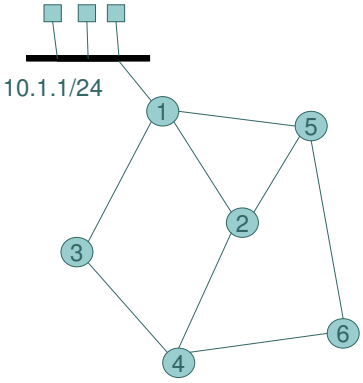
CS514



1. Each node tells its neighbors what networks it has, who its neighbors are (link-state update), and the cost associated with each link (metric).
2. These link-state updates are flooded to all nodes (periodically, and also whenever there is a change).
3. As a result, each node builds up a complete map of the network (RIB)

Link State Routing

CS514



1. Each node tells its neighbors what networks it has, who its neighbors are (link-state update), and the cost associated with each link (metric).
2. These link-state updates are flooded to all nodes (periodically, and also whenever there is a change).
3. As a result, each node builds up a complete map of the network (RIB)
4. A spanning-tree algorithm is run over the network to produce a set of shortest paths. From these the FIB is generated.

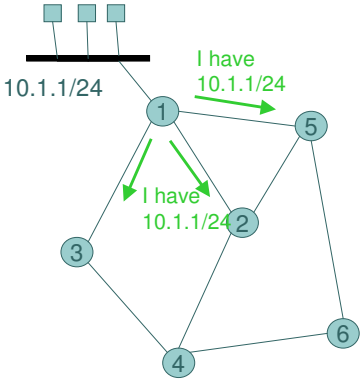
Tricky parts of link state

CS514

- Must flood updates quickly
  - Transient loops exist until all nodes synchronized
  - But flooding and spanning tree calculation are expensive---flapping link can cause havoc
- Knowing most recent update is trickier than you'd think
  - Circular sequence number space math (this brought down Arpanet years ago)
    - 0xff > 0, 0xffff > 0xff, 0 > 0xffff
  - If node reboots, initialized seq number can be mistaken for an old one and ignored
  - Decnet solution: very large seq number (years worth)
    - If max reached, node crashes itself

Distance Vector Routing

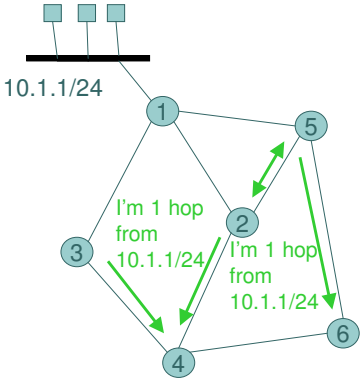
CS514



1. Each node tells its neighbors what networks it has.

Distance Vector Routing

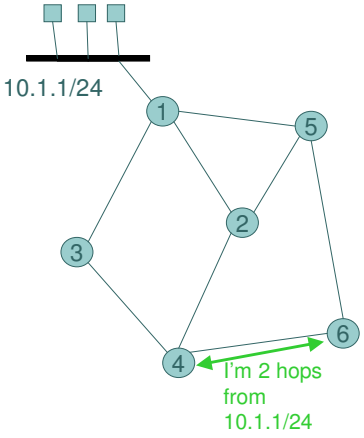
CS514



1. Each node tells its neighbors what networks it has.
2. Each node tells how many hops it is from each destination (periodically in RIP, periodically and when changes occur in BGP).

Distance Vector Routing

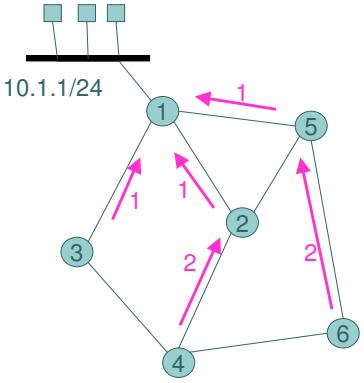
CS514



1. Each node tells its neighbors what networks it has.
2. Each node tells how many hops it is from each destination (periodically in RIP, periodically and when changes occur in BGP).

Distance Vector Routing

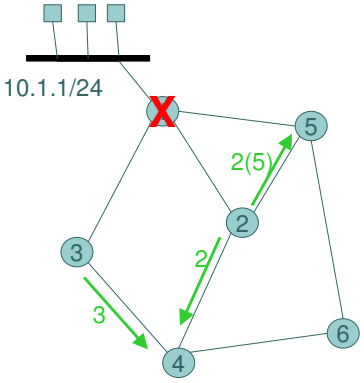
CS514



1. Each node tells its neighbors what networks it has.
2. Each node tells how many hops it is from each destination (periodically in RIP, periodically and when changes occur in BGP).
3. Each node selects the neighbor with the fewest hops as its next hop towards each destination. These are inserted into the FIB.

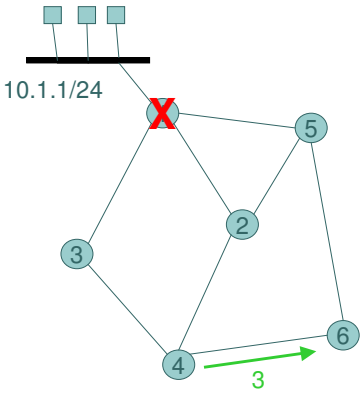
Distance Vector Problem: Counting to Infinity

CS514



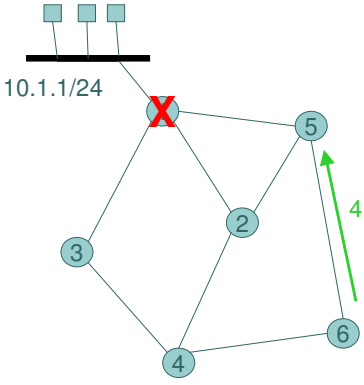
Distance Vector Problem: Counting to Infinity

CS514



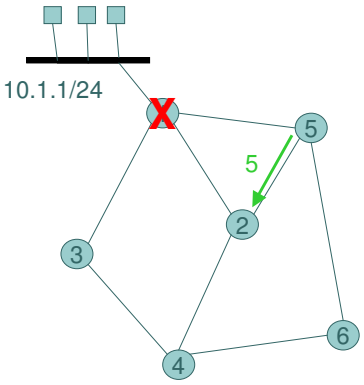
Distance Vector Problem: Counting to Infinity

CS514



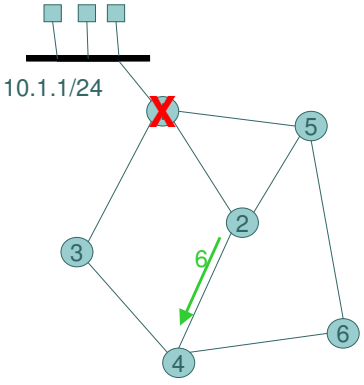
Distance Vector Problem: Counting to Infinity

CS514



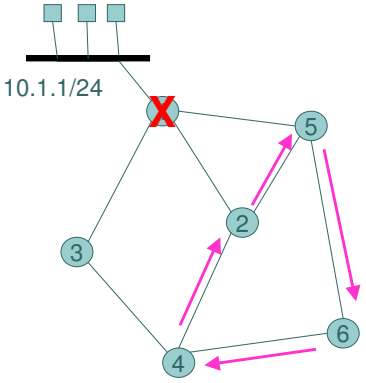
Distance Vector Problem: Counting to Infinity

CS514



Distance Vector Problem: Counting to Infinity

CS514



Packets are forwarded in a loop

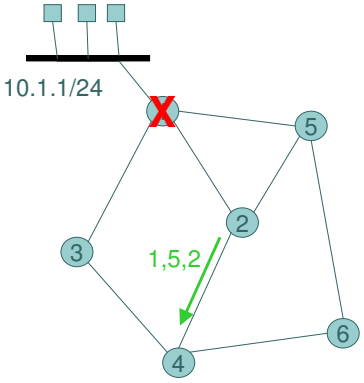
Count-to-Infinity solution: Path Vector

CS514

- Include entire path to destination instead of hop count
  - Add yourself to any path you advertise
- Simple rule: Don't import any path that you are already in

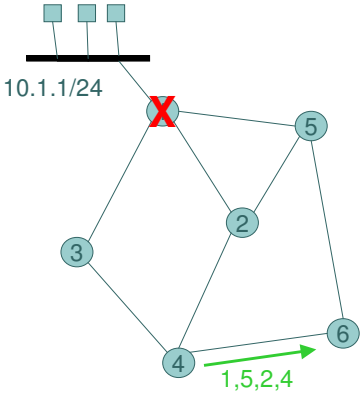
Path Vector

CS514



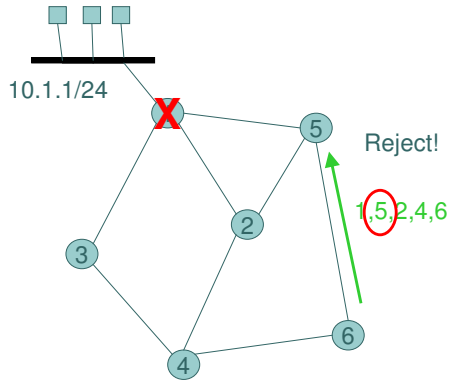
Distance Vector Problem: Counting to Infinity

CS514



Distance Vector Problem: Counting to Infinity

CS514



Inter-domain and Intra-domain Routing

CS514

- Internet architecture separates intra-domain routing (within a site or ISP) from inter-domain routing (between sites and ISPs)
  - Intra-domain (IGP): OSPF (LS), IS-IS (LS), RIP (DV)
  - Inter-domain (EGP): BGP (PV)
- Obvious reasons:
  - Different requirements between intra- and inter-
  - Autonomy: within a site you should be able to run what you want.

OSPF (Open shortest path first)

CS514

- Link-state, Intra-domain
  - Run within ISPs as well as sites
- Two-level hierarchy
  - Backbone plus stubs
  - Hides details of stub topologies from rest of network
- Can import routes from external networks
  - BGP, or even other IGPs

OSPF (Open shortest path first)

CS514

- Elects “designated router” on LANs
  - Avoid N^2 updates for a single LAN
- Has a multicast extension
  - Flood group membership everywhere (expensive)
- Has an IPv6 extension

BGP (Border Gateway Protocol)

CS514

- The inter-domain routing protocol
- Path vector
  - Elements of path are ISPs and sites, not individual routers
  - Each ISP and some sites have an Autonomous System (AS) number (16 bits)
  - These AS numbers constitute the path

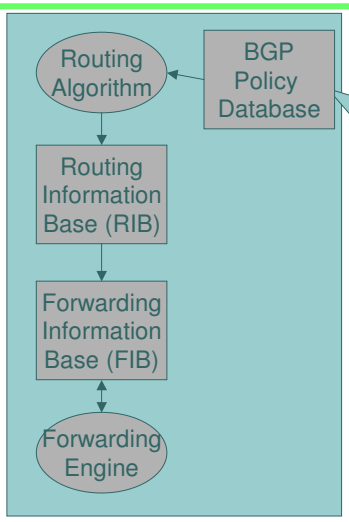
Why path vector and not link state?

CS514

- Link state would never work across multiple autonomous administrations
  - Link state requires that every router agree on value of every link metric
  - But different domains will disagree on the "cost" of crossing a given AS
    - AS may give itself a large cost (avoid having to be a transit), whereas other ASs may rather it a low cost
- Path vector allows each AS to make its own policy decisions
  - (with limitations)

BGP Policy Database

CS514



Set of peers
What to originate
What to import (per peer)
What to export (per peer)
Route filters
Aggregation policies

Routing policy is not part of BPG per se, but rather a set of locally configured rules

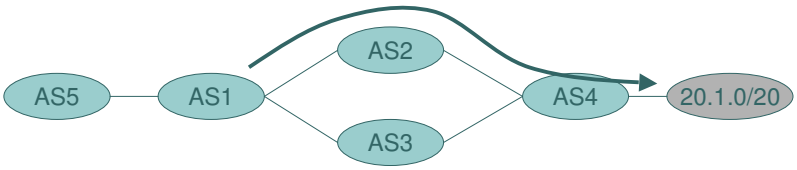
What kinds of policy decisions?

CS514

- Who to peer with (which ASs)
- What routes to originate
- What routes to import (prevent bogus advertisements)
- What routes to export (and how to aggregate them)
- What paths to prefer
  - Shorter AS paths
  - Some ASs preferred over others
    - The big ASs (UUnet, AT&T, etc.)
    - Primary versus backup transit AS

BGP policy limitation (hop by hop policy decisions)

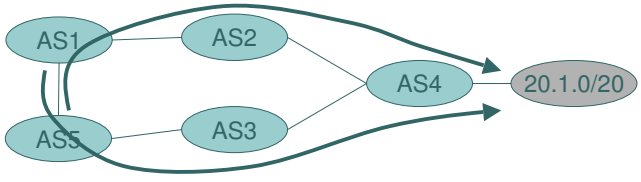
CS514



AS1 chooses AS2 as the path to 20.1.0/20.
AS5 is forced to accept the choice of AS1
(If AS5 really doesn't like it, it should find a new peer)

BGP policy conflict

CS514

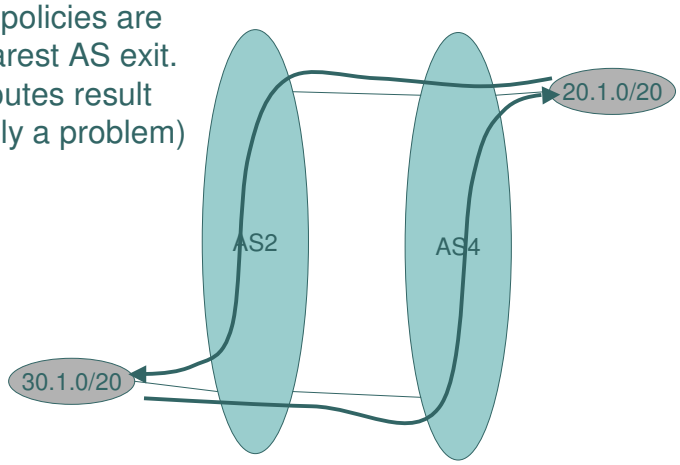


AS5 policy is to prefer route to AS4 via AS2
AS1 policy is to prefer route to AS4 via AS3
Both policies cannot be satisfied

Hot potato routing

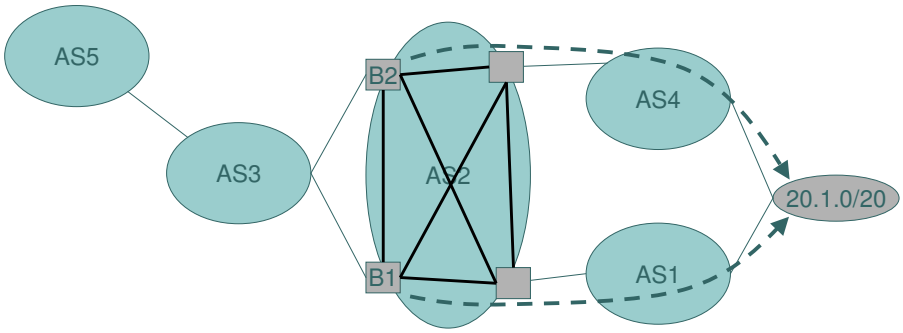
CS514

AS2 and AS4 policies are to route to nearest AS exit. Asymmetric routes result (not necessarily a problem)



Misconfigured policies may lead to oscillation

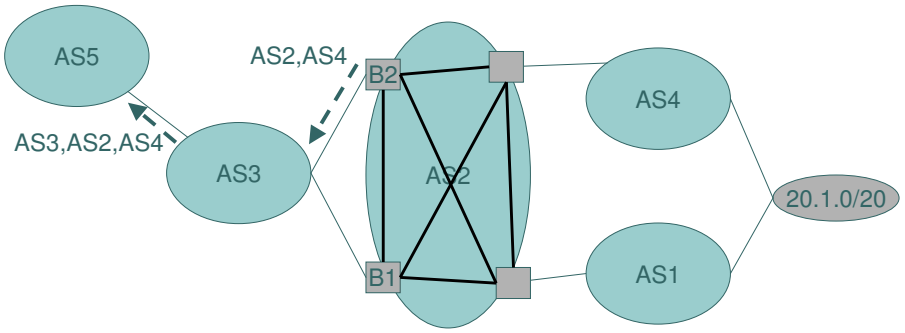
CS514



B2 configured to prefer AS4
B1 configured to prefer AS1

Misconfigured policies may lead to oscillation

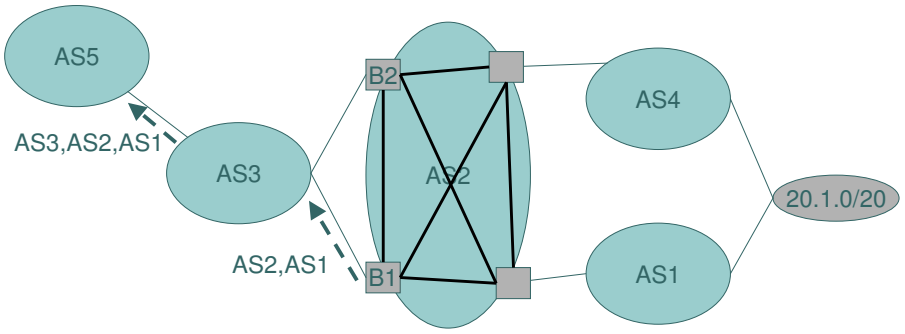
CS514



B2 (periodically) updates AS3 with path AS2,AS4

Misconfigured policies may lead to oscillation

CS514



B1 (periodically) updates AS3 with path AS2,AS1
With each period AS3 advertises a different route

Other route flapping

CS514

- A link continuously goes up and down
  - The update for this is propagated throughout the internet
- Mid-90's these kinds of problems were severe
  - 1996: 45,000 prefixes, 1,500 unique AS paths, 1,300 ASs, 3-6 million BGP update messages/day
    - 6 updates per prefix per hour!
    - (Labovitz et. al.)

Today much improved

CS514

- Better policy tools
- Better software
- Lots of damping
- But still, advances in BGP lead to new policy bugs
  - Route reflectors published in 2000 (RFC2796)
  - Inconsistent route reflectors problem published in 2002 (RFC3345)

Policy Tools

CS514

- Routing Policy Specification Language (RPSL) (RFC 2280)
  - Earlier policy languages exist
- Language to define BGP policies
  - Peers, import, export, route preference, aggregation
- Posted at Routing Registries (RIPE, RADB, etc.)
- Tools created to look for policy inconsistencies (within AS and across ASs)
- Tools created to match measured reality (BGP tables, traceroute) with policy expectations
  - RAToolSet, USC/ISI

Lots of Damping

CS514

- Stop advertising certain prefixes if they go up and down a lot
  - Improve stability
  - Lower overhead
- RIPE guidelines:
  - Don't dampen until after 4th flap in a row (in 50 minutes)
  - /24: dampen 60 minutes
  - /22,/23, dampen 30-45 minutes
  - </22, dampen 10-30 minutes
- Helps the internet, but means that you can go away for a long time
  - Because of some problem in the middle!

Effect of BGP policies on path quality

CS514

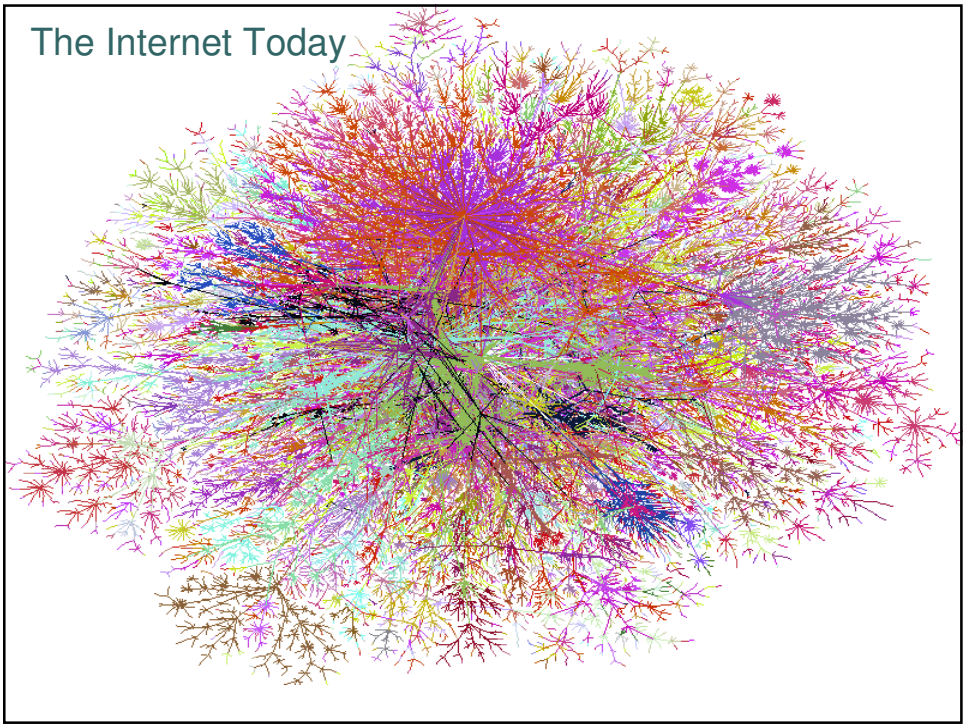
- Ramesh Govindan study (USC)
- Methodology:
  - Learn real physical topology with traceroutes, deduce actual AS connectivity
    - Imperfect, but not bad
  - Examine used “policy topology” from BGP tables, RADB (routing registry) database
  - Compare the two

Effect of BGP policies on path quality

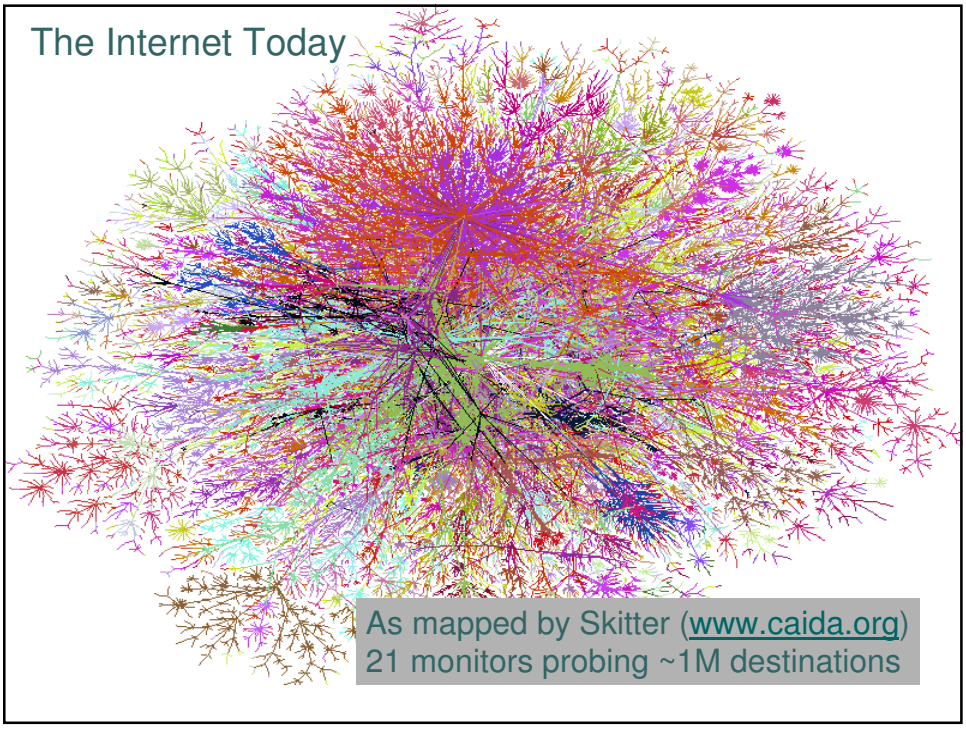
CS514

- Results:
  - About ½ of the paths a longer than shortest path
  - 20% of policy paths are 50% or more longer
  - 20% of policy paths are 5 hops or more longer
  - Policy tends to push paths through major backbones rather than possibly shorter routes
    - (But shorter routes may not be better routes!)

The Internet Today



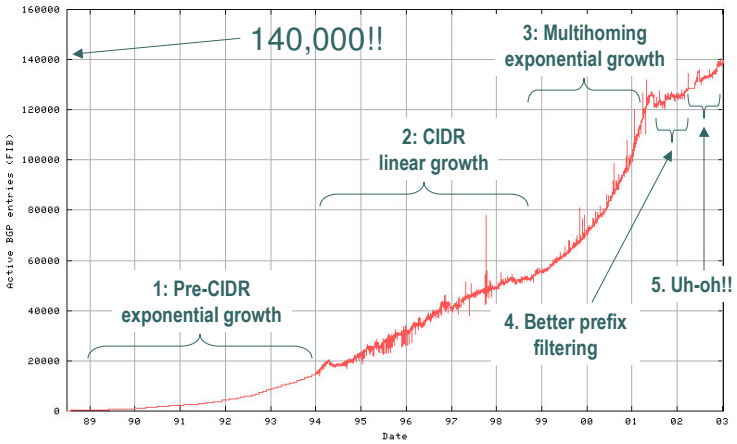
The Internet Today



As mapped by Skitter (www.caida.org)
21 monitors probing ~1M destinations

BGP Routing Table Growth

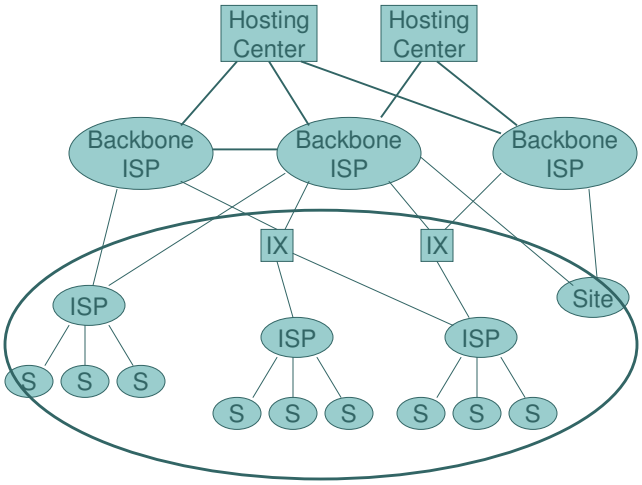
CS514



Source: The CIDR Report, www.cidr-report.org

Internet topology again

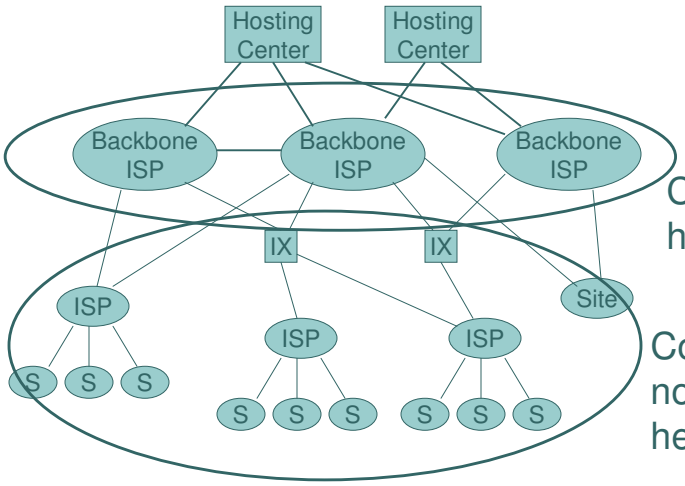
CS514



Complex and noisy down here

Internet topology again

CS514



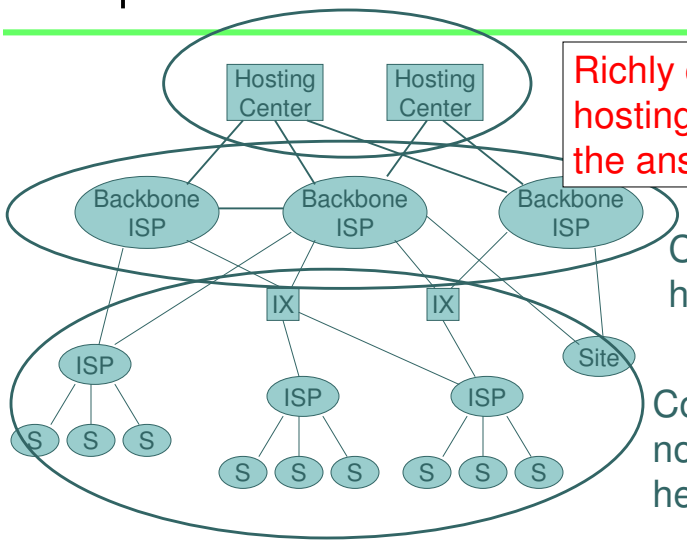
Cleaner up here

Complex and noisy down here

Internet topology again

CS514

Richly connected hosting centers still the answer!



Cleaner up here

Complex and noisy down here

Hosting Centers

CS514

- Connect to multiple major backbone ISPs
  - Avoid damp down
  - Avoid oscillations
  - Avoid thin peering points
  - Get good paths