Multihoming and Multi-path Routing

CS 7260
Nick Feamster
January 29. 2007

Today's Topic

- **IP-Based Multihoming**
  - What is it?
  - What problem is it solving? (Why multihome?)
  - How is it implemented today (in IP)?
  - Traffic Engineering
  - How many upstream ISPs are enough?

- Problems with IP-based multihoming
  - Inbound route control
  - Routing table growth

- Another approach: **host-based multihoming**

What is Multihoming?

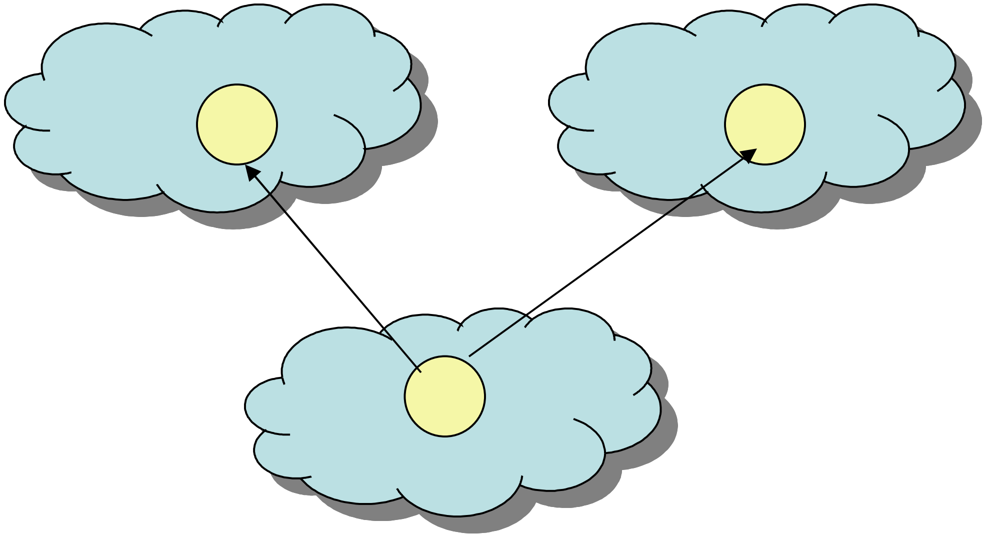
- The use of redundant network links for the purposes of *external* connectivity

- Can be achieved at many layers of the protocol stack and many places in the network
  - Multiple network interfaces in a PC
  - An ISP with multiple upstream interfaces

- Can refer to having multiple connections to
  - The same ISP
  - Multiple ISPs

Why Multihome?

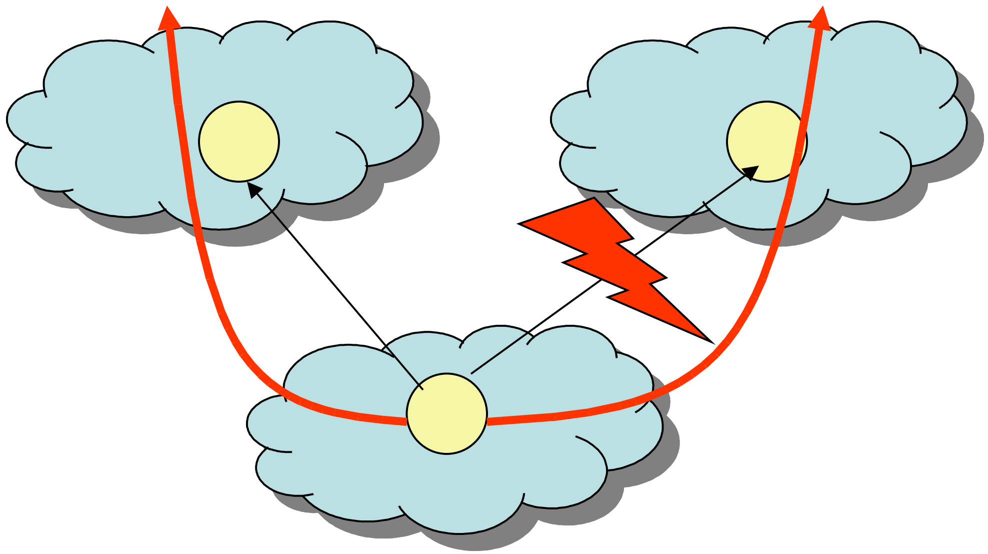
- Redundancy
- Availability
- Performance
- Cost



Interdomain traffic engineering: the process by which a multihomed network configures its network to achieve these goals

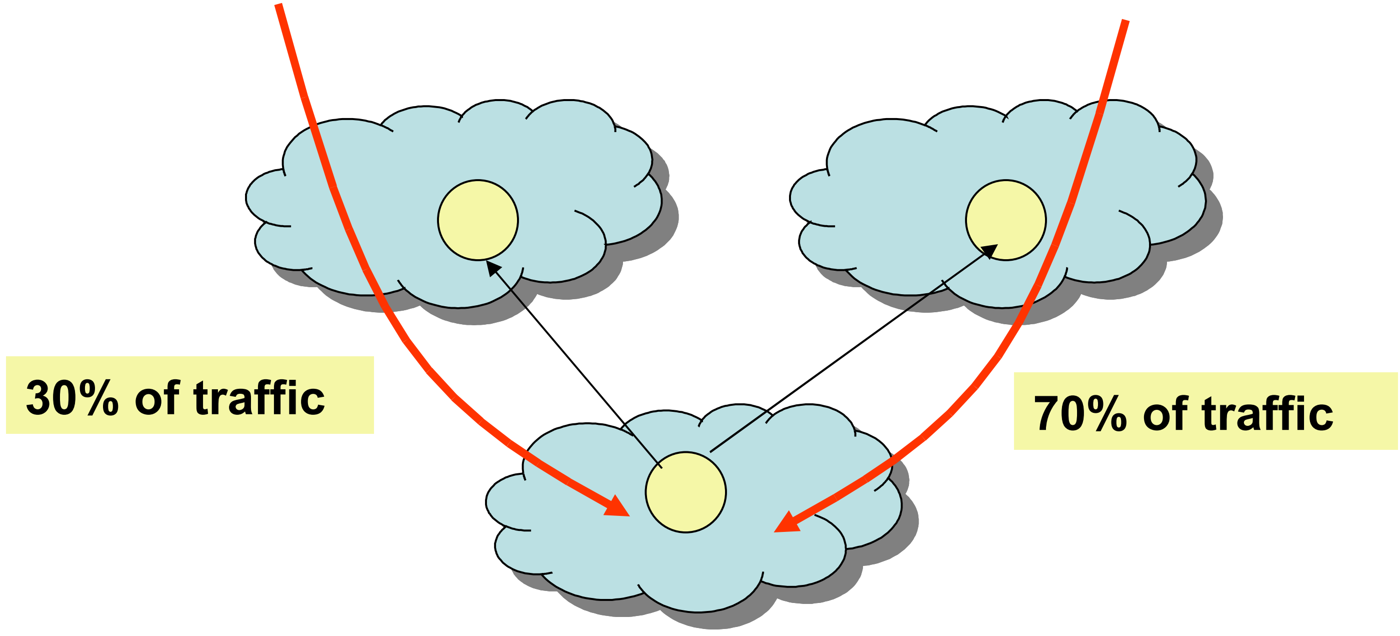
Redundancy

- Maintain connectivity in the face of:
  - Physical connectivity problems (fiber cut, device failures, etc.)
  - Failures in upstream ISP



Performance

- Use multiple network links at once to achieve **higher throughput** than just over a single link.
- Allows incoming traffic to be **load-balanced**.



Multihoming in IP Networks Today

- **Stub AS:** no transit service for other ASes
  - No need to use BGP
- **Multi-homed stub AS:** has connectivity to multiple immediate upstream ISPs
  - Need BGP
  - No need for a public AS number
  - No need for IP prefix allocation
- **Multi-homed transit AS:** connectivity to multiple ASes *and* transit service
  - Need BGP, public AS number, IP prefix allocation

BGP or no?

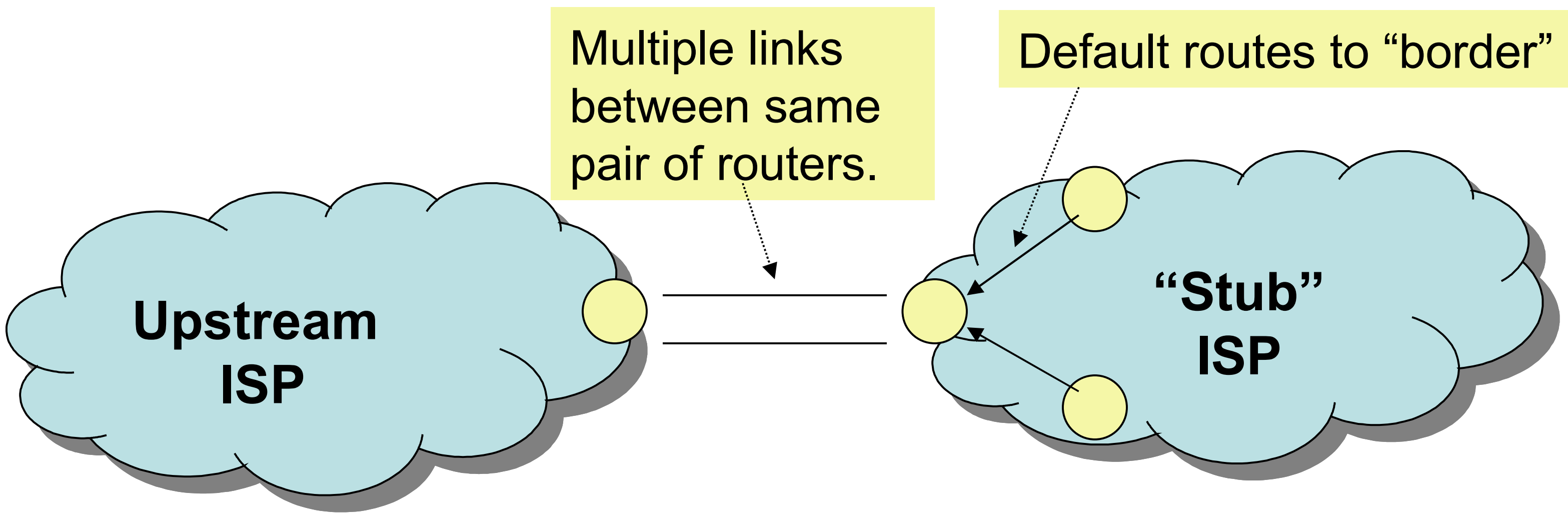
- Advantages of **static routing**
  - Cheaper/smaller routers (less true nowadays)
  - Simpler to configure
- Advantages of **BGP**
  - More control of your destiny (have providers stop announcing you)
  - Faster/more intelligent selection of where to send outbound packets.
  - Better debugging of net problems (you can see the Internet topology now)

Same Provider or Multiple?

- If your provider is reliable and fast, and affordably, and offers good tech-support, you may want to multi-home initially to them via some backup path (slow is better than dead).

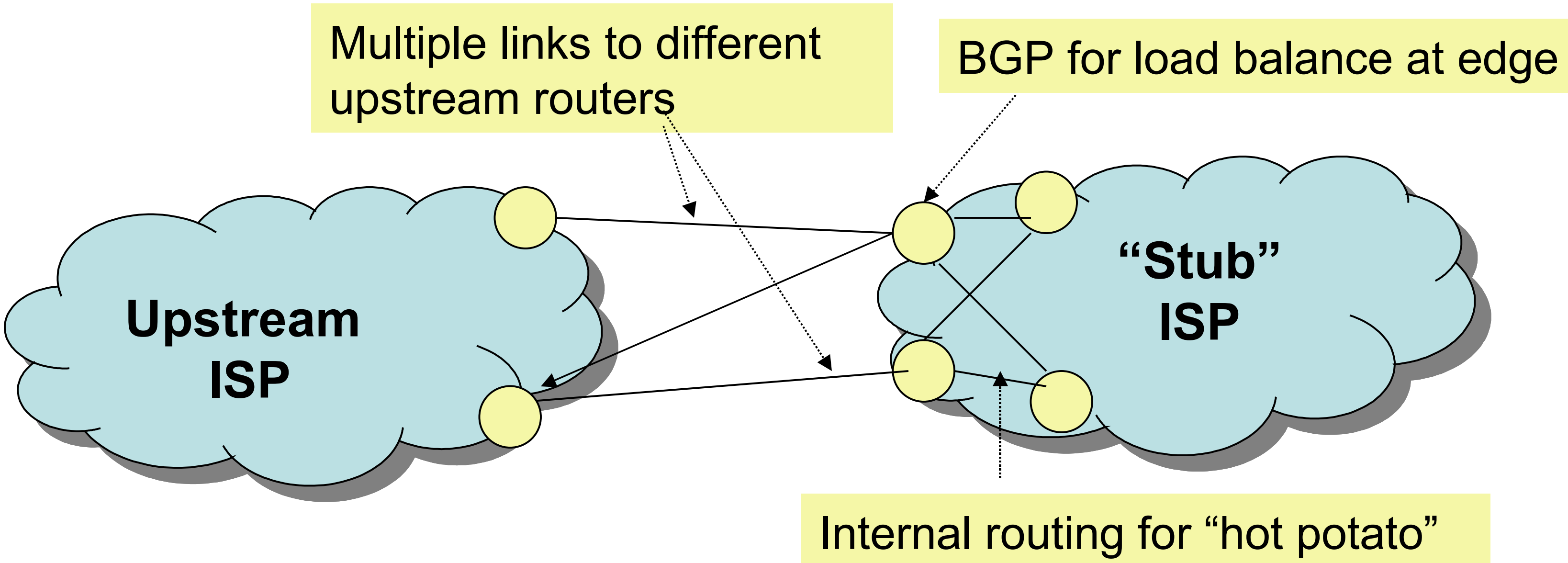
- Eventually you'll want to multi-home to different providers, to avoid failure modes due to one provider's architecture decisions.

Multihomed Stub: One Link



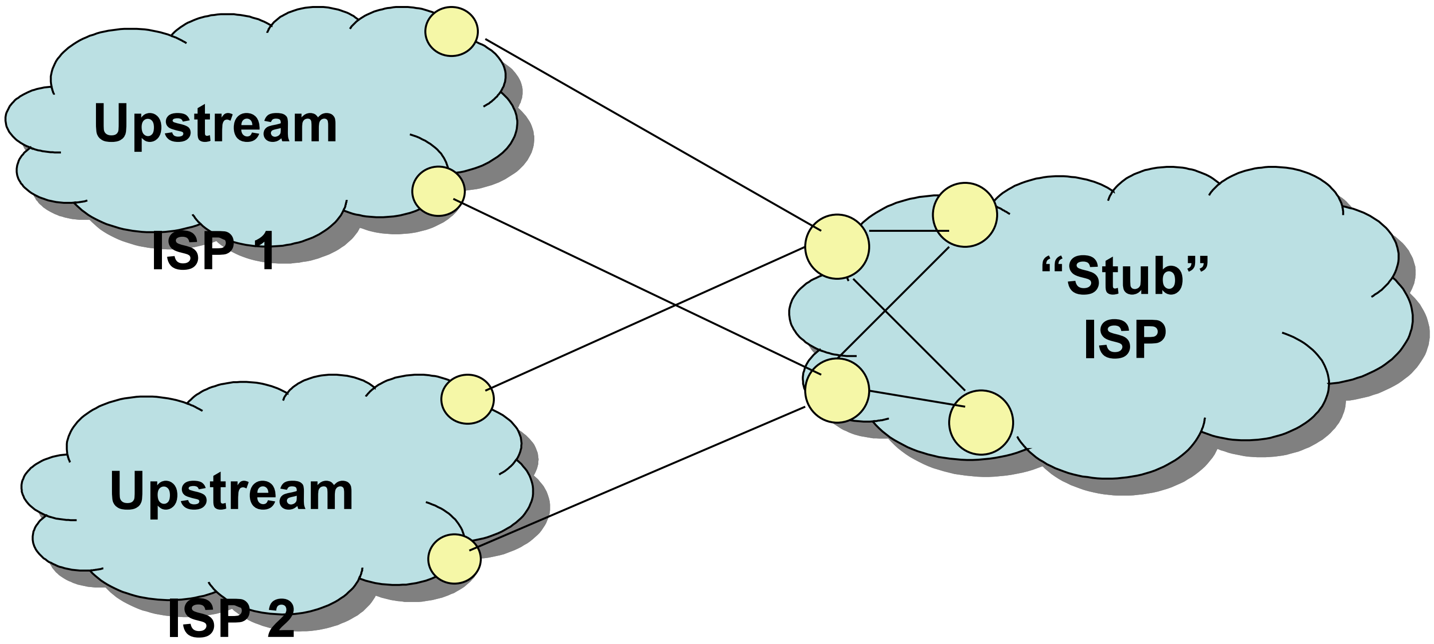
- Downstream ISP’s routers configure default (“static”) routes pointing to border router.
- Upstream ISP advertises reachability

Multihomed Stub: Multiple Links



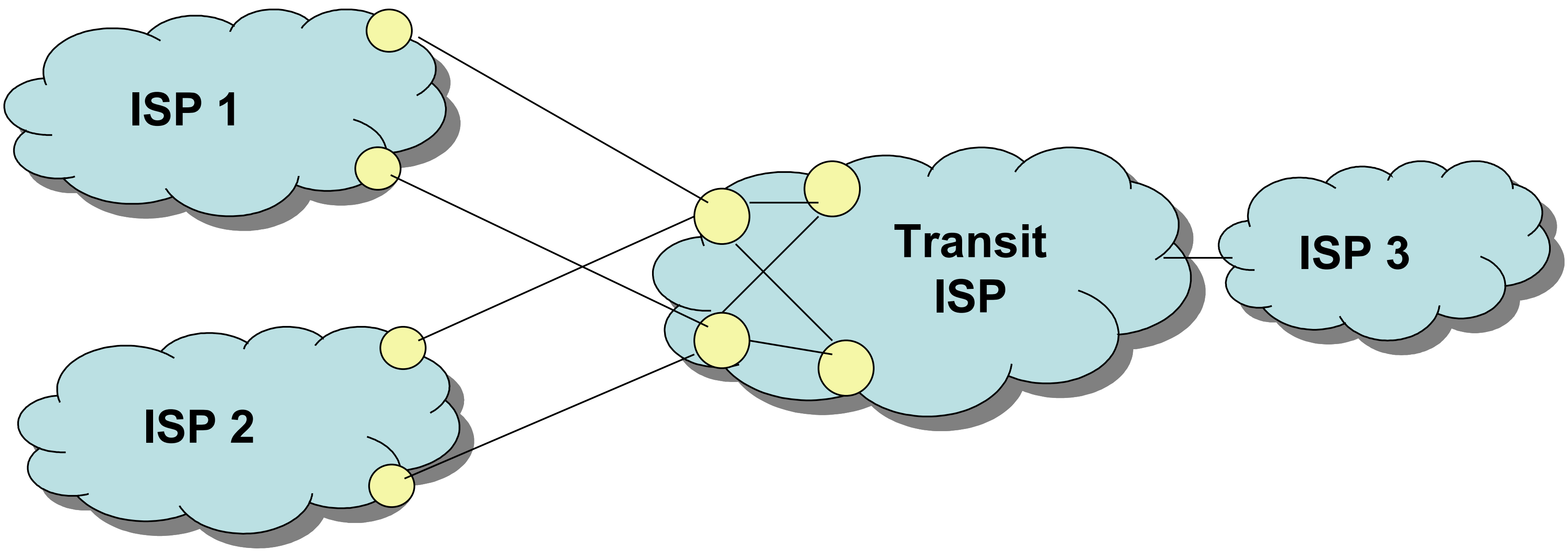
- Use BGP to share load
- Use **private AS number** *(why is this OK?)*
- As before, upstream ISP advertises prefix

Multihomed Stub: Multiple ISPs



- Many possibilities
  - Load sharing
  - Primary-backup
  - Selective use of different ISPs
- Requires BGP, public AS number, etc.

Multihomed Transit Network



- BGP everywhere
- Incoming and outcoming traffic
- **Challenge:** balancing load on intradomain *and* egress links, given an offered traffic load

Interdomain Traffic Engineering

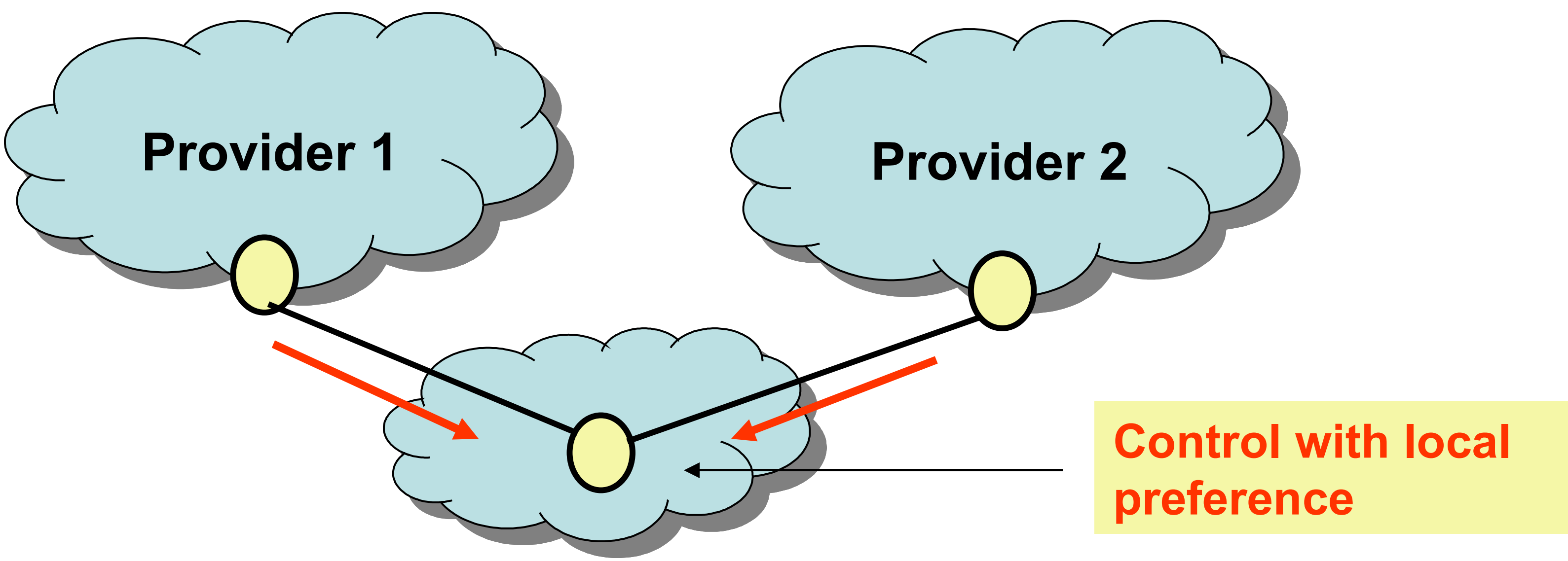
- The process by which a network operator configures the network to achieve
  - Traffic load balance
  - Redundancy (primary/backup), etc.

- Two tasks
  - Outbound traffic control
  - Inbound traffic control

- **Key Problems:** Predictability and Scalability

Outbound Traffic Control

- Easier to control than inbound traffic
  - *Destination-based routing:* sender determines where the packets go
- Control over **next-hop AS only**
  - Cannot control selection of the entire path



Outbound Traffic: Load Balancing

- Control routes to provider **per-prefix**
  - Assign local preference across destination prefixes
  - Change the local preference assignments over time
- Useful inputs to load balancing
  - End-to-end path performance data
  - Outbound traffic statistics per destination prefix
- **Challenge:** Getting from traffic volumes to groups of prefixes that should be assigned to each link

Premise of "intelligent route control" preoducts.

Traffic Engineering Goals

- Predictability
  - Ensure the BGP decision process is deterministic
  - Assume that BGP updates are (relatively) stable
- Limit overhead introduced by routing changes
  - Minimize frequency of changes to routing policies
  - Limit number of prefixes affected by changes
- Limit impact on how traffic enters the network
  - Avoid new routes that might change neighbor’s mind
  - Select route with same attributes, or at least path length

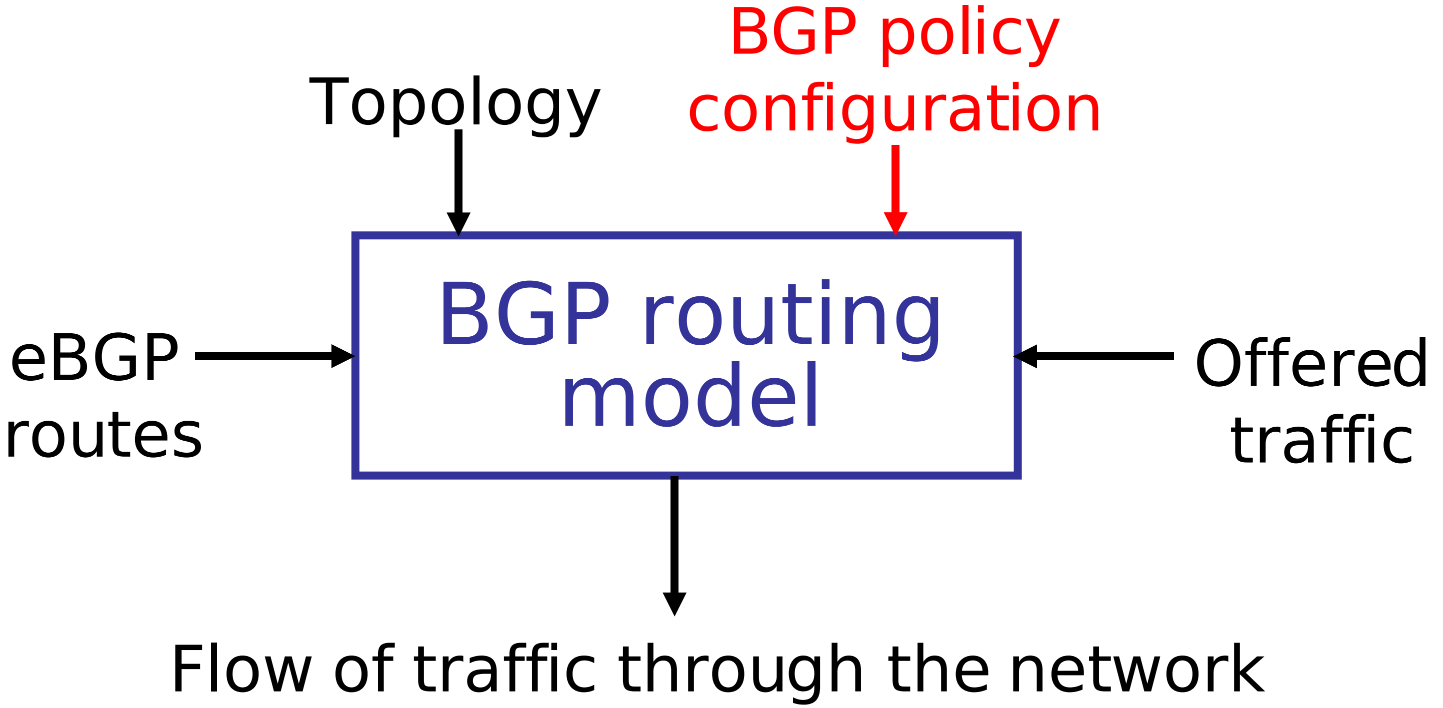
Managing Scale

- Destination prefixes
  - More than 90,000 destination prefixes
    - Don’t want to have per-prefix routing policies
  - Small fraction of prefixes contribute most of the traffic
    - Focus on the small number of heavy hitters
  - Define routing policies for selected prefixes

- Routing choices
  - About 27,000 unique “routing choices”
    - Help in reducing the scale of the problem
  - Small fraction of “routing choices” contribute most traffic
    - Focus on the very small number of “routing choices”
  - Define routing policies on common attributes

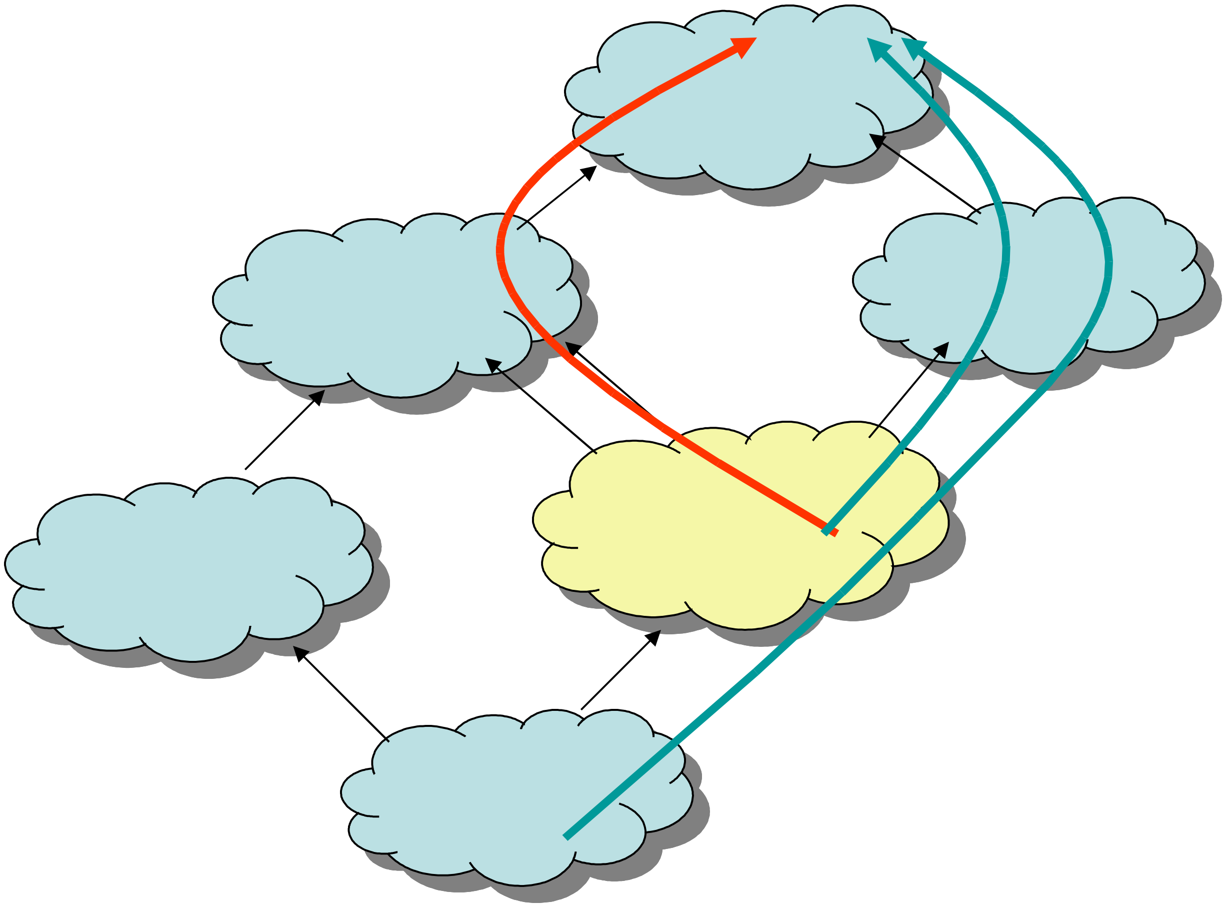
Achieving Predictability

- Route prediction with static analysis
  - Helpful to know effects *before* deployment
  - Static analysis can help

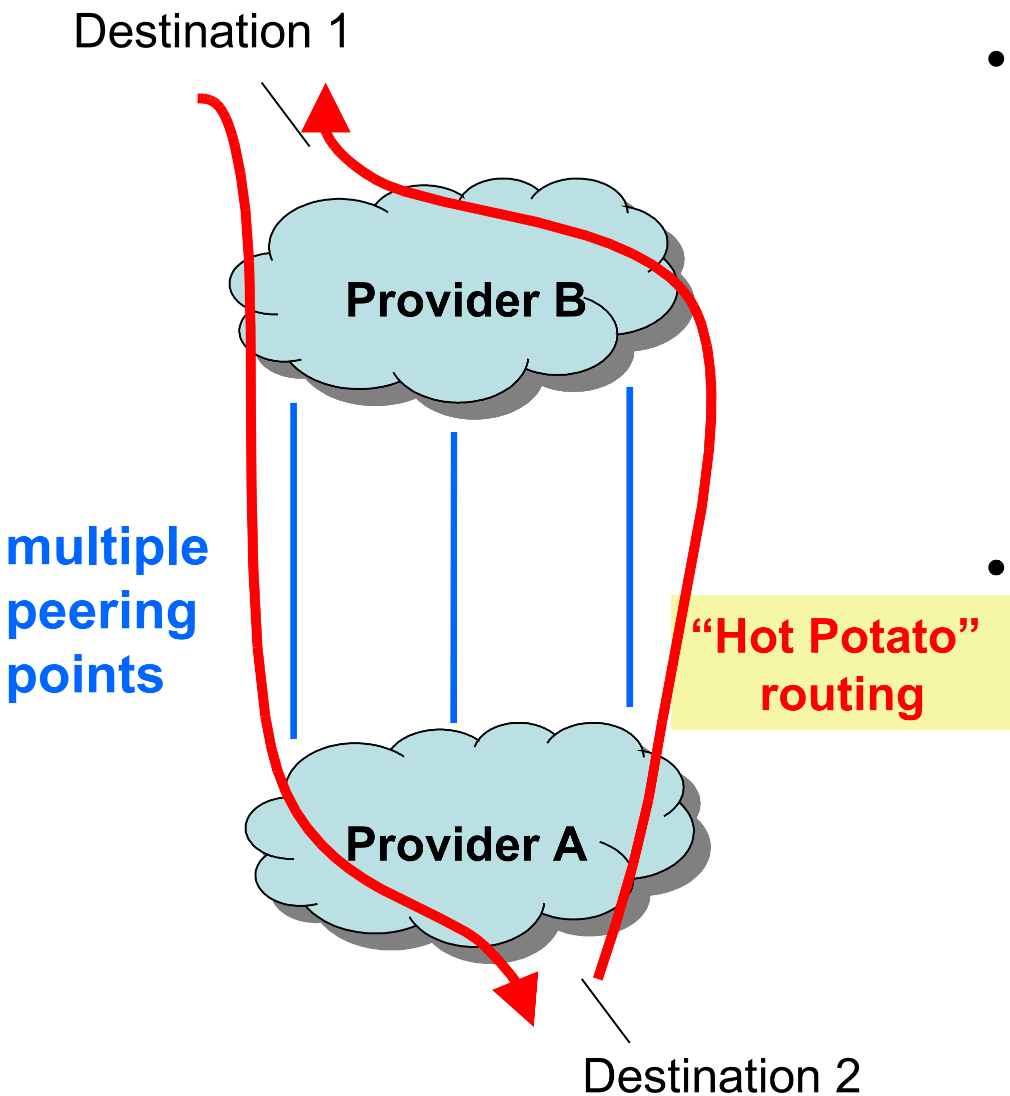


Challenges to Predictability

- For transit ISPs: effects on incoming traffic
  - Lack of coordination strikes again!



Inter-AS Negotiation



- Coordination aids predictability
  - Negotiate where to send
  - Inbound and outbound
  - Mutual benefits

- How to implement?
  - What info to exchange?
  - Protecting privacy?
  - How to prioritize choices?
  - How to prevent cheating?

Outbound: Multihoming Goals

- **Redundancy**
  - Dynamic routing will failover to backup link
- **Performance**
  - Select provider with best performance per prefix
  - Requires active probing
- **Cost**
  - Select provider per prefix over time to minimize the total financial cost

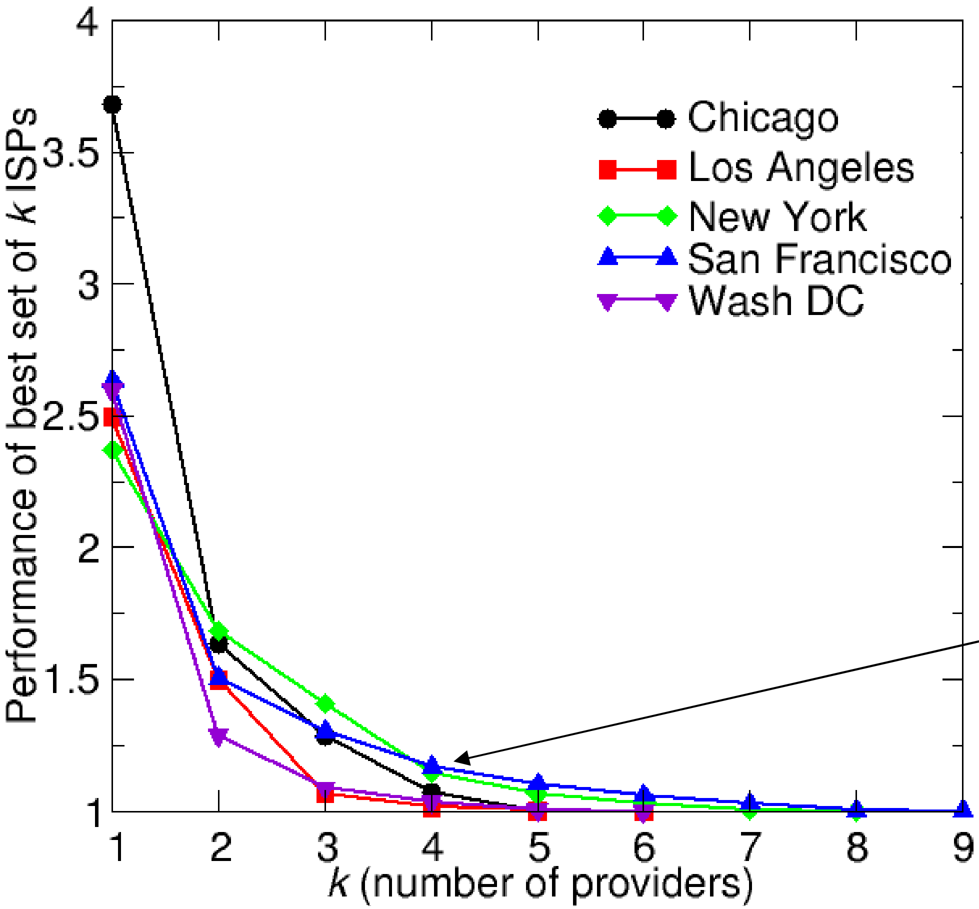
Inbound Traffic Control

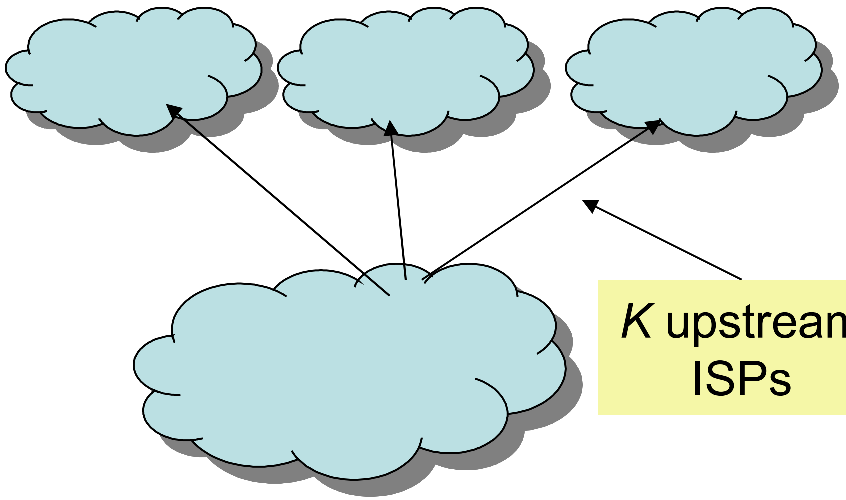
- **More difficult:** no control over neighbors' decisions.

- Three common techniques (previously discussed)
  - AS path prepending
  - Communities and local preference
  - Prefix splitting

How does today's paper (MONET) control inbound traffic?

How many links are enough?





K upstream ISPs

Not much benefit beyond 4 ISPs

Akella *et al.*, "Performance Benefits of Multihoming", *SIGCOMM 2003*

Problems with Multihoming in IPv4

- Routing table growth
  - Provider-based addressing
  - Advertising prefix out multiple ISPs – can’t aggregate

- Poor control over inbound traffic
  - Existing mechanisms do not allow hosts to control inbound traffic

Today's Reading

- *Source Selectable Path Diversity via Routing Deflections*, Yang *et al.*

- **Main idea:** Sources can detect and react to failures more quickly than the routing protocols often can.

- Source routing is appealing, but…
  – Scaling problems
  – Routers designed to forward on destination address

Benefits

- No need for coordination across ISPs
- No need for additional machinery (simple tweaks to shortest path routing work well)

Two Key Components

- Deflection Rules
  - Needed to prevent loops when packets are deflected
  - **Simple idea:** deflect packets only to hopes that are closer to the destination
  - **Complication:** may not expose enough path diversity
    - Deflections may come straight back

Enhancement #1: Two Hops Down

- **Rule:** Packet can be forwarded to any intermediate node for which the length of the path decreases along a two-hop sequence

- **Question:** Why will this not cause loops?
- **Answer:** 2-hop sequence always decreases cost.

- **Additional cost:** Forwarding decisions also depend on incoming link

Enhancement #2: Two Hops Forward

- Same as previous rule, but remove the incoming link used to reach the node in question

- Can cause more roundabout paths

Discussion Questions

- How does it work with BGP?
- Who’s responsible for tagging packets?
- Is this enough diversity?
- Is it too much? (i.e., is latency too high?)
- Overload?
  - Opposite: Better balancing/QoS?
- Stability problems?
- Selfish behavior?
- How good is random?