Implementation

Major Internet routing protocols:

- RIP (v1 and v2): intra-domain, Bellman-Ford
  → also called “distance vector”
  → metric: hop count
  → UDP
  → nearest neighbor advertisement
  → popular in small intra-domain networks

- OSPF (v1 and v2): intra-domain, Dijkstra
  → also called “link state”
  → metric: average delay
  → directly over IP: protocol number 89
  → broadcasting via flooding
  → popular in larger intra-domain networks
• IS-IS: intra-domain, Dijkstra
  → “link state”
  → directly over link layer (e.g., Ethernet)
  → more recently: also available over IP
  → flooding
  → popular in larger intra-domain networks

• Source routing: packet specifies path
  → implemented in various link layer protocols
  → ATM call set-up: circuit-switching
  → IPv4/v6: option field
  → mostly disabled
  → large ISPs: sometimes used internally for diagnosis
BGP (Border Gateway Protocol):

- Inter-domain routing
  - border routers vs. backbone routers

→ “peering” between two AS’s
→ includes customer-provider relationship
→ exchanges: peering between multiple AS’s
• CIDR addressing
  \[ \text{i.e., } a.b.c.d/x \]
  \[ \rightarrow \text{Purdue: } 128.10.0.0/16, 128.210.0.0/16, 204.52.32.0/20 \]
  \[ \rightarrow \text{check at www.iana.org (e.g., ARIN for US)} \]
• Route table look-up: maximum prefix matching
  \[ \rightarrow \text{e.g., entries: } 128.10.0.0/16 \text{ and } 128.10.27.0/24 \]
  \[ \rightarrow \text{destination address } 128.10.27.20 \text{ matches } 128.10.27.0/24 \text{ best} \]
• Metric: policy
  \[ \rightarrow \text{e.g., shortest-path, trust, pricing} \]
  \[ \rightarrow \text{meaning of “shortest”: delay, router hop, AS hop} \]
  \[ \rightarrow \text{route amplification: shortest AS path } \neq \text{ shortest router path} \]
  \[ \rightarrow \text{mechanism: path vector routing} \]
  \[ \rightarrow \text{BPG update message} \]
BGP route update:

\[ \rightarrow \text{BGP update message propagation} \]

BGP update message:

\[ ASN_{A_k} \rightarrow \cdots \rightarrow ASN_{A_2} \rightarrow ASN_{A_1}; a.b.c.d/x \]

Meaning: ASN \( A_1 \) (with CIDR address a.b.c.d/x) can be reached through indicated path

\[ \rightarrow \text{“path vector”} \]

\[ \rightarrow \text{called AS-PATH} \]

Some AS numbers:

- Purdue: 17
- BBN: 1
- UUNET: 701
- Level3: 3356
- Abilene (aka “Internet2”): 11537
Purdue’s backbone network (Fall 2004): ITaP
Level3 backbone network: www.level3.com

→ 10 Gbps backbone (same as Purdue)
→ part of backbone: OC-48 (2.488 Gbps)
Abilene/Internet2 backbone: www.internet2.edu
Policy:

- if multiple AS-PATHs to target AS are known, choose one based on policy
  
  → e.g., shortest AS path length, cheapest, least worrisome

- advertise to neighbors target AS’s reachability
  
  → also subject to policy

  → no obligation to advertise

  → specifics depend on bilateral contract (SLA)

SLA (service level agreement):

  → bandwidth (e.g., 1 Gbps, OC-3, DS3)

  → delay (e.g., avrg. 25ms US), loss (e.g., 0.05%)

  → pricing (e.g., 1 Mbps: below $100)

  → availability (e.g., 99.999%)

  → etc.
Ex:

- AS F
- AS C
- AS B
- AS A
- AS H
- AS D
- AS G
- AS E

Diagram:

- AS F -> AS B -> AS A; a.b.c.d/x
- AS H -> AS D -> AS B -> AS A; a.b.c.d/x
- AS G -> AS D -> AS B -> AS A; a.b.c.d/x
- Purdue: ASN 17; 128.10.0.0/16
BGP-update procedure:

Upon receiving BGP update message from neighbor to target AS $A$

1. Store AS-PATH reachability info for target $A$
   $\rightarrow$ AdjIn table (one per neighbor)

2. Determine if new path to $A$ should be adopted
   $\rightarrow$ policy
   $\rightarrow$ path should be unique
   $\rightarrow$ BPG table (locRIB) & IP routing table update
   $\rightarrow$ inter-domain: IP table update from BGP

3. Determine who to advertise reachability for target $A$
   $\rightarrow$ selective advertisement

Note: if shortest-path then same as Dijkstra in-reverse
BGP-withdrawal:
1. Use BGP keep-alive message to sense neighbor
   → timeout
2. If keep-alive does not arrive within timeout, assume node is down
3. Send BGP withdraw message for neighbor who is deemed down if no alternative path exists; else send BGP update message
   → may trigger further updates

Other BGP features:
- BGP runs over TCP
  → port number 179
  → i.e., “application layer” protocol
- BPG-4 (1995); secure BGP
  → S-BGP: not implemented yet (“BBN vs. Cisco”)
Performance

Route update frequency:

→ routing table stability vs. responsiveness
→ rule: not too frequently
→ 30 seconds
→ stability wins
→ hard lesson learned from the past (sub-second)
→ legacy: TTL

Other factors for route instability:

→ selfishness (e.g., fluttering)
→ BGP’s vector path routing: inherently unstable
→ more common: slow convergence
→ target of denial-of-service (DoS) attack
Route amplification:

→ shortest AS path \( \neq \) shortest router path

→ e.g., may be several router hops longer

→ AS graph vs. router graph

→ inter- vs. intra-domain routing: separate subsystems

→ policy: company in Denmark

Route asymmetry:

→ routes are not symmetric

→ estimate: > 50%

→ mainly artifact of inter-domain policy routing

→ various performance implications

→ source traceback
Black holes:

$\rightarrow$ persistent unreachable destination prefixes
$\rightarrow$ BGP routing problems
$\rightarrow$ further aggrevated by DNS
$\rightarrow$ purely application layer: end system problem
Topology:

→ who is connected to whom

→ Internet AS graph (segment of Jan. 2002)
Contrast with random graph: same number of nodes and edges

\[ \rightarrow \text{random graph: choose each link with prob. } p \]

\[ \rightarrow \text{independently: prob. of } k \text{ neighbors is } p^k \]
Phenomenon:

\[ \Pr\{u \text{ has } k \text{ neighbors}\} \propto k^{-\alpha} \quad (2 < \alpha < 3) \]

\[ \text{called power-law graph} \]

In contrast to random graph:

\[ \Pr\{u \text{ has } k \text{ neighbors}\} \propto p^k \]

\[ \text{probability is exponentially small in } k \]

\[ \text{UUNET (AS 701) has } > 2500 \text{ neighbors!} \]

\[ > 12500 \text{ domains in 2002} \]

\[ \text{probabilistically UUNET should not exist} \]

\[ \text{so things are not random} \]

What’s going on . . .

\[ \text{connection to airlines?} \]
Ex.: Delta Airlines route map

→ by design: hub and backbone architecture
→ mixture of centralized/decentralized design
→ small system: centralized is good
→ large system: decentralization necessary
Small system with centralized design:

→ star topology

→ e.g., Southwest Airlines

→ essentially two conjoined star topologies

→ a matter of load balancing

→ backbone topology: trivial
Simple backbone topologies comprised of stars:

- ring of stars
- mesh of stars
- random/planar backbone of stars
- tree (hierarchy) of stars

\[ \Pr\{\deg(u) = k\} \propto k^{-\alpha} \]

- different star sizes
- cliques: peering at exchange points
- tier’ed hierarchy
- sparse backbone: random-like
View as “molecular stew” of lego-like building blocks:

\[ n \Pr(\deg(u)=4) \]

→ “stir” stew of ingredients until graph is formed

→ no dangling links

The aforementioned: structural design point-of-view

“A few are connected to many, many are connected to a few.”

Dynamic point-of-view:

→ “The rich get richer, the poor get poorer.”

→ growth process: preferential attachment

→ attach to \( u \) with probability \( \propto \deg(u) \)

→ makes sense up to a point
Performance implications:

- bad: single point of “failure”
  - note domains don’t fail like routers
- bad: severe load imbalance
  - perform similar calculation as ad hoc
- good: “Checkpoint Charlie”
  - can detect and act on bad traffic efficiently
  - small deployment but large impact
  - e.g., worm and DDoS attack traffic filtering
- good: caching put content close to demand: efficiency

Power-law connectivity: not restricted to domain graphs

  - e.g., WWW, call, router, metabolic networks
  - social sciences: 1950s and earlier
  - Milgram’s “small world” (six degrees of separation)