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

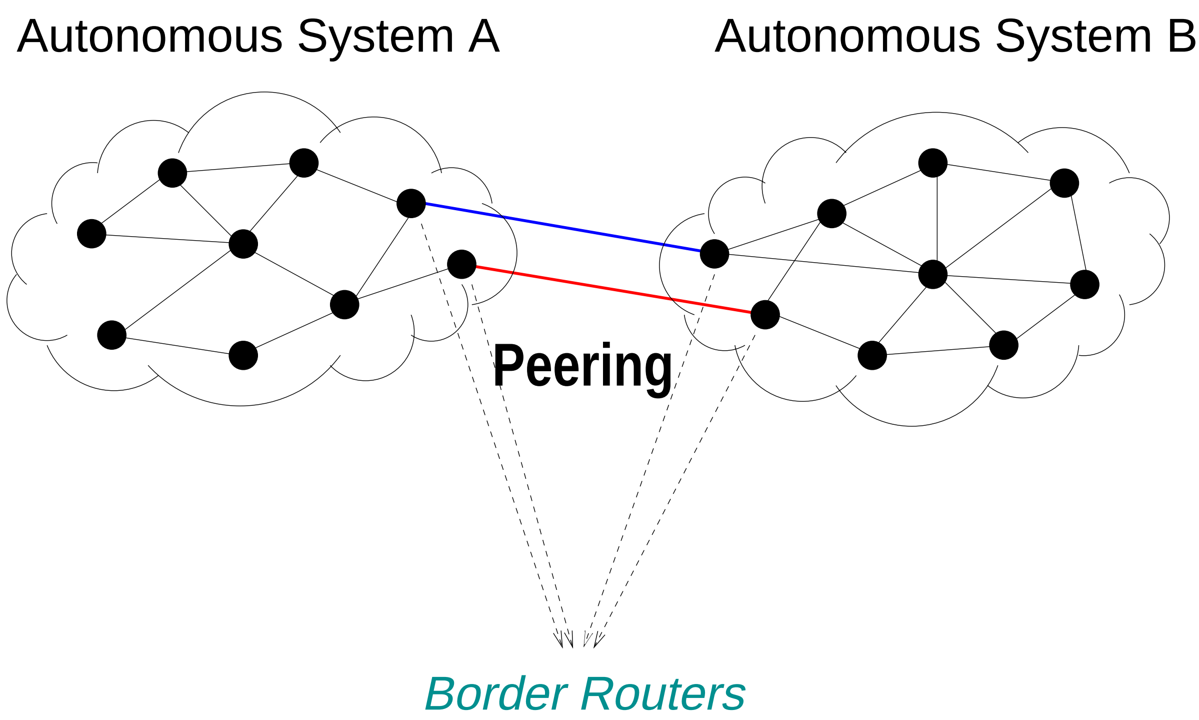
  → mostly disabled

  → large ISPs: sometimes used internally for diagnosis

BGP (Border Gateway Protocol):

- Inter-domain routing

  $\longrightarrow$ border routers vs. backbone routers



$\longrightarrow$ "peering" between two AS's

$\longrightarrow$ includes customer-provider relationship

$\longrightarrow$ exchanges: peering between multiple AS's

- CIDR addressing
  - → i.e., $a.b.c.d/x$
  - → Purdue: 128.10.0.0/16, 128.210.0.0/16, 204.52.32.0/20
  - → check at `www.iana.org` (e.g., ARIN for US)
- Route table look-up: maximum prefix matching
  - → e.g., entries: 128.10.0.0/16 and 128.10.27.0/24
  - → destination address 128.10.27.20 matches 128.10.27.0/24 best
- Metric: policy
  - → e.g., shortest-path, trust, pricing
  - → meaning of "shortest": delay, router hop, AS hop
  - → route amplification: shortest AS path $\neq$ shortest router path
  - → mechanism: path vector routing
  - → BPG update message

BGP route update:

$\longrightarrow$ BGP update message propagation

BGP update message:

$$\text{ASN}A_k \rightarrow \cdots \rightarrow \text{ASN}A_2 \rightarrow \text{ASN}A_1; \text{a.b.c.d/x}$$

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

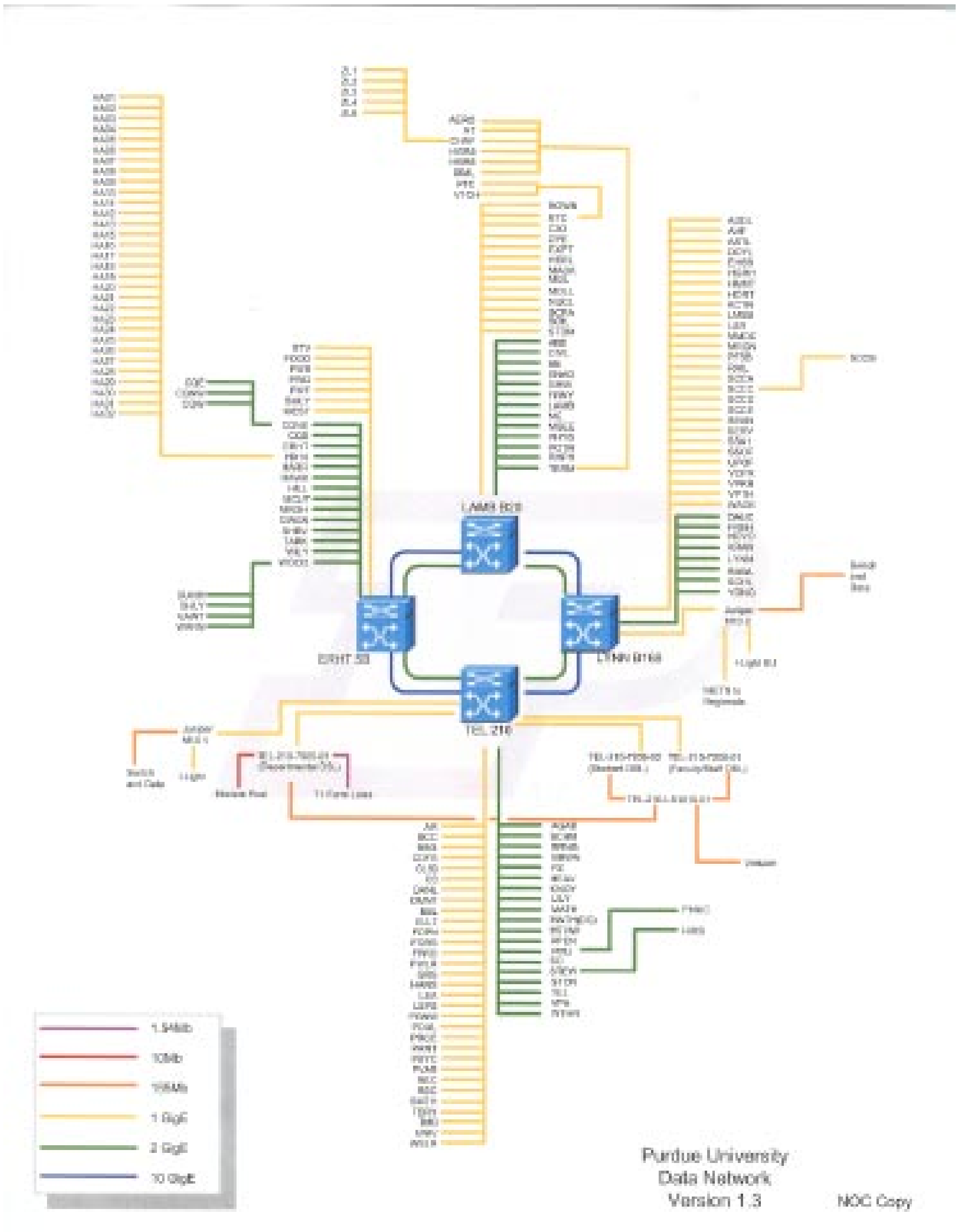
$\longrightarrow$ "path vector"

$\longrightarrow$ called AS-PATH

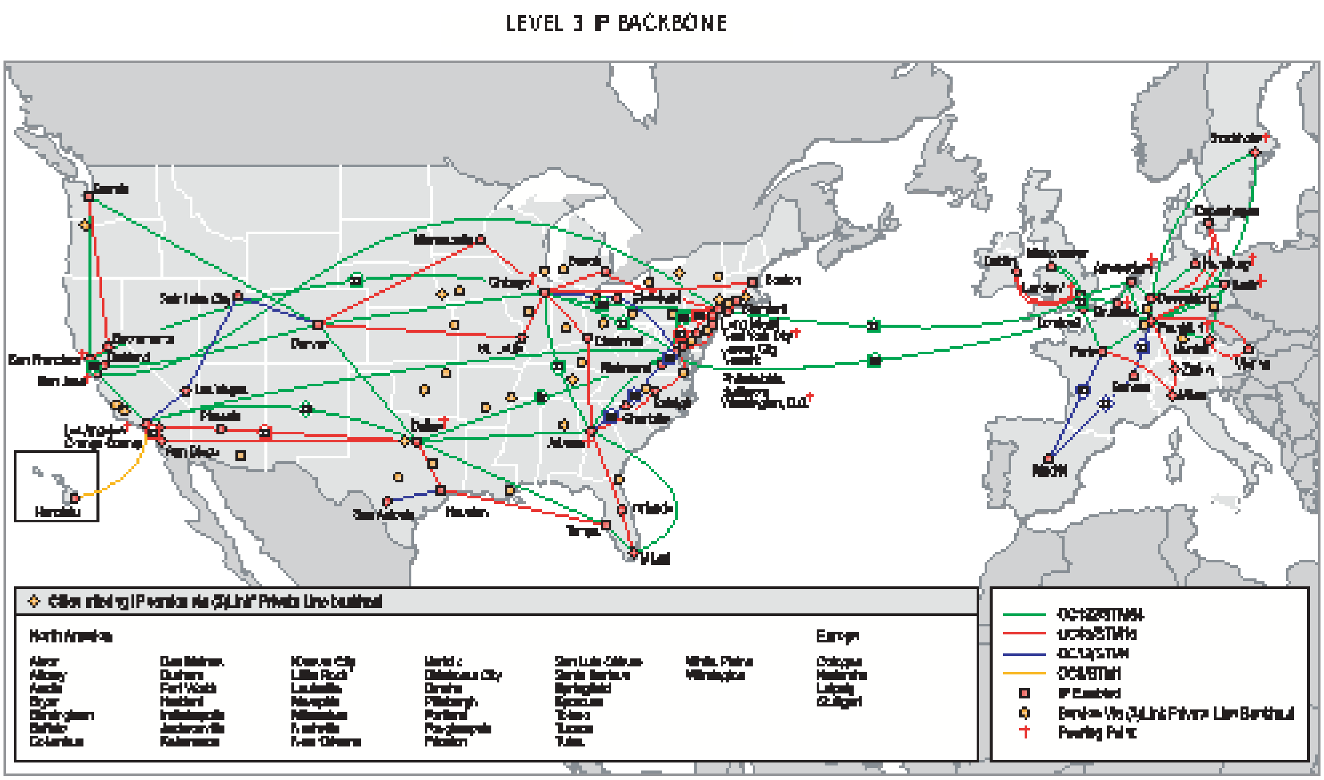
Some AS numbers:

- Purdue: 17
- BBN: 1
- UUNET: 701
- Level3: 3356
- Abilene (aka "Internet2"): 11537

Purdue's backbone network (Fall 2004): ITaP



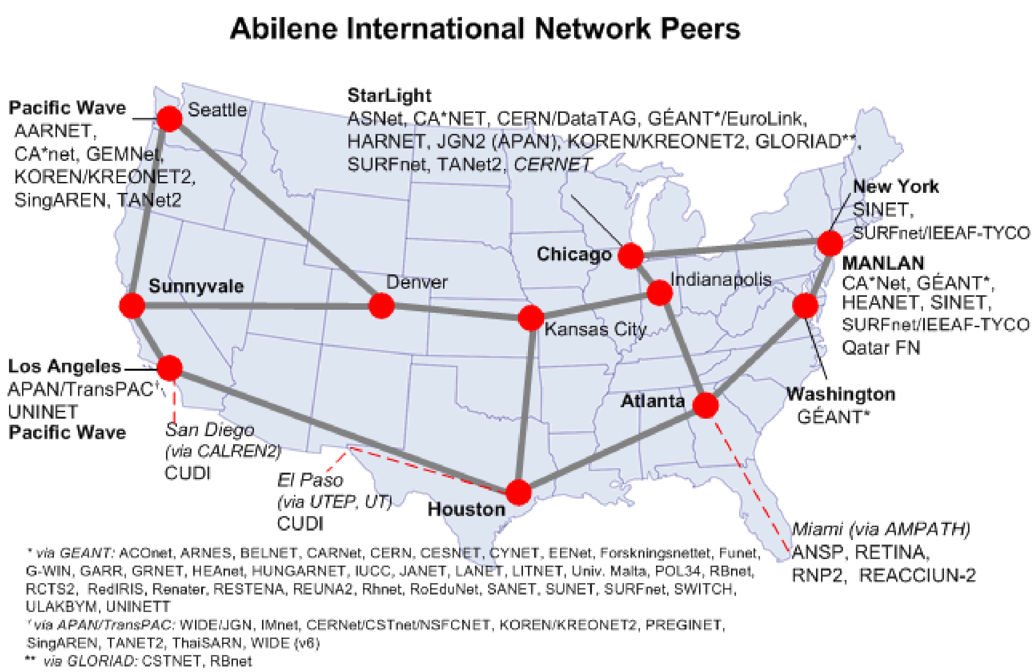
Level3 backbone network: www.level3.com



$\longrightarrow$ 10 Gbps backbone (same as Purdue)

$\longrightarrow$ 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

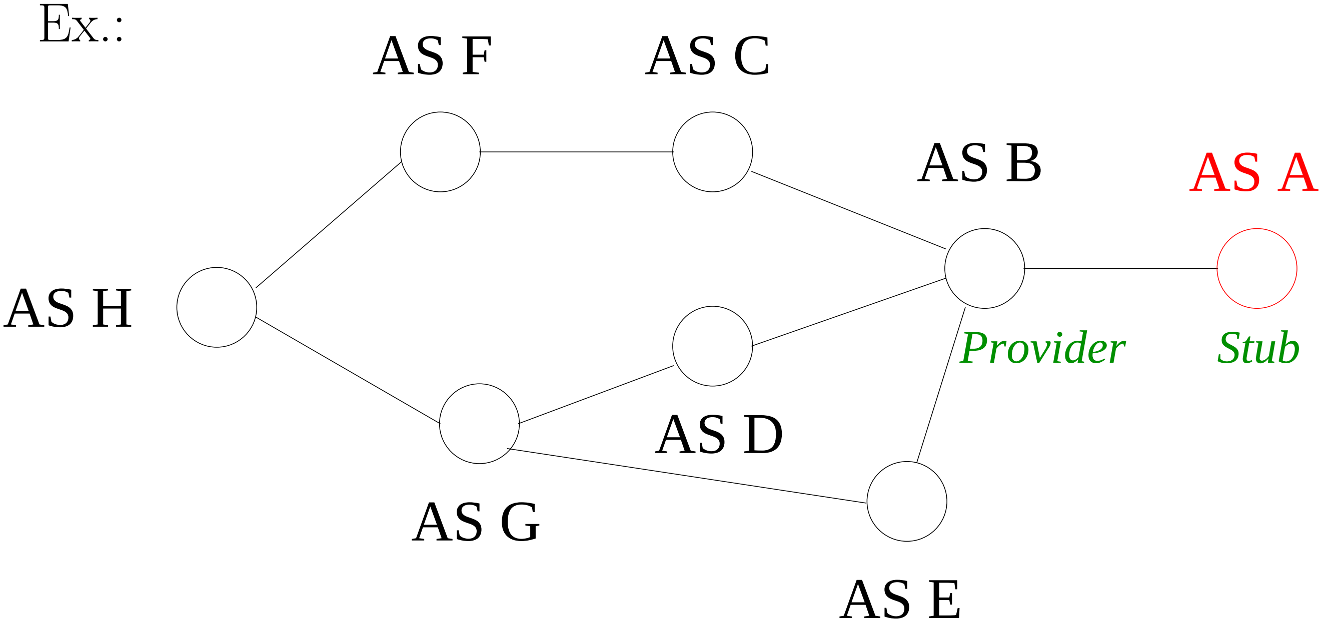
⟶ delay (e.g., avrg. 25ms US), loss (e.g., 0.05%)

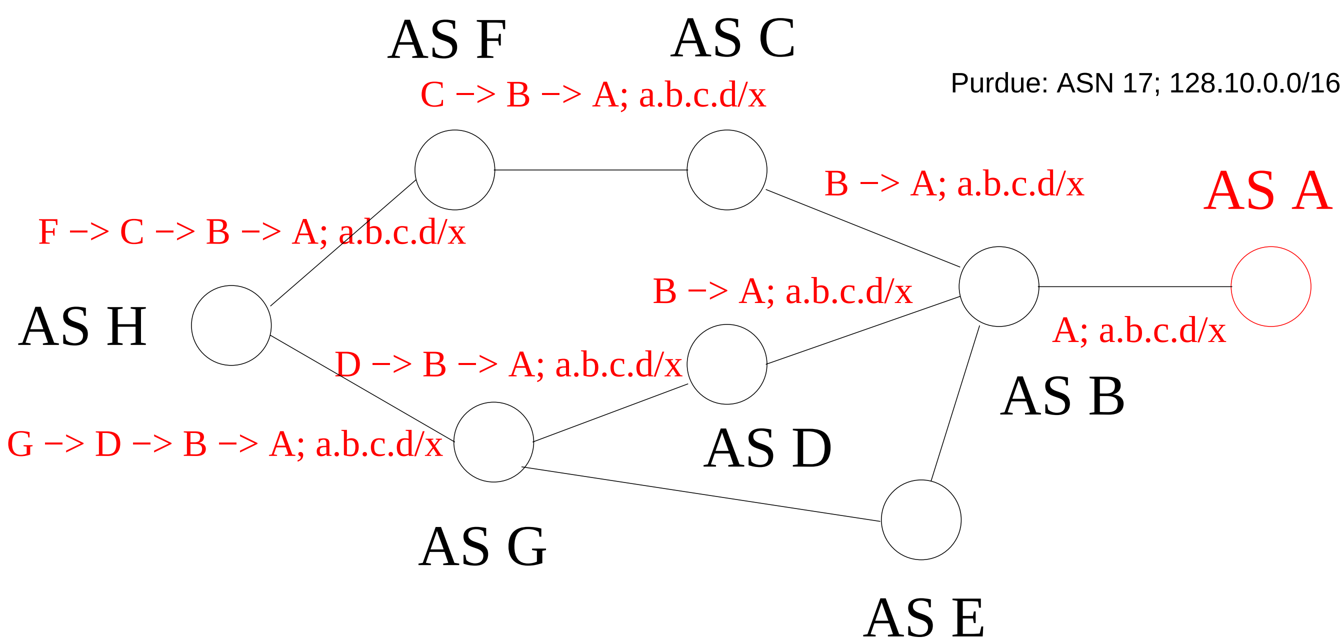
⟶ pricing (e.g., 1 Mbps: below $100)

⟶ availability (e.g., 99.999%)

⟶ etc.

Ex.:


AS F
AS C
AS B
AS A
AS H
Provider
Stub
AS D
AS G
AS E


AS F
AS C
Purdue: ASN 17; 128.10.0.0/16
C −> B −> A; a.b.c.d/x
B −> A; a.b.c.d/x
AS A
F −> C −> B −> A; a.b.c.d/x
B −> A; a.b.c.d/x
AS H
A; a.b.c.d/x
D −> B −> A; a.b.c.d/x
AS B
G −> D −> B −> A; a.b.c.d/x
AS D
AS G
AS E

BGP-update procedure:

Upon receiving BGP update message from neighbor to target AS A

1. Store AS-PATH reachability info for target A

   → `AdjIn` table (one per neighbor)

2. Determine if new path to A should be adopted

   → policy

   → path should be unique

   → BPG table (`locRIB`) & IP routing table update

   → inter-domain: IP table update from BGP

3. Determine who to advertise reachability for target A

   → 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:

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

Other factors for route instability:

- $\longrightarrow$ selfishness (e.g., fluttering)
- $\longrightarrow$ BGP's vector path routing: inherently unstable
- $\longrightarrow$ more common: slow convergence
- $\longrightarrow$ target of denial-of-service (DoS) attack

Route amplification:

$\longrightarrow$ shortest AS path $\neq$ shortest router path

$\longrightarrow$ e.g., may be several router hops longer

$\longrightarrow$ AS graph vs. router graph

$\longrightarrow$ inter- vs. intra-domain routing: separate subsystems

$\longrightarrow$ policy: company in Denmark

Route asymmetry:

$\longrightarrow$ routes are not symmetric

$\longrightarrow$ estimate: $> 50\%$

$\longrightarrow$ mainly artifact of inter-domain policy routing

$\longrightarrow$ various performance implications

$\longrightarrow$ source traceback

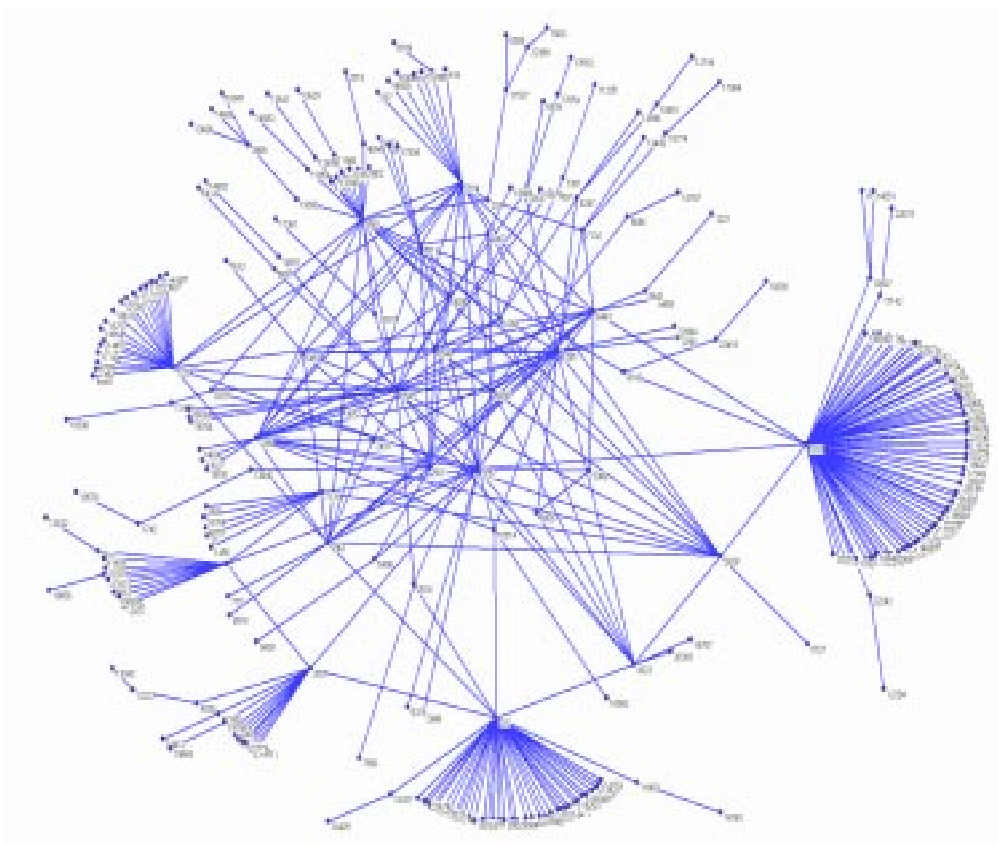
Black holes:

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

Topology:

⟶ who is connected to whom

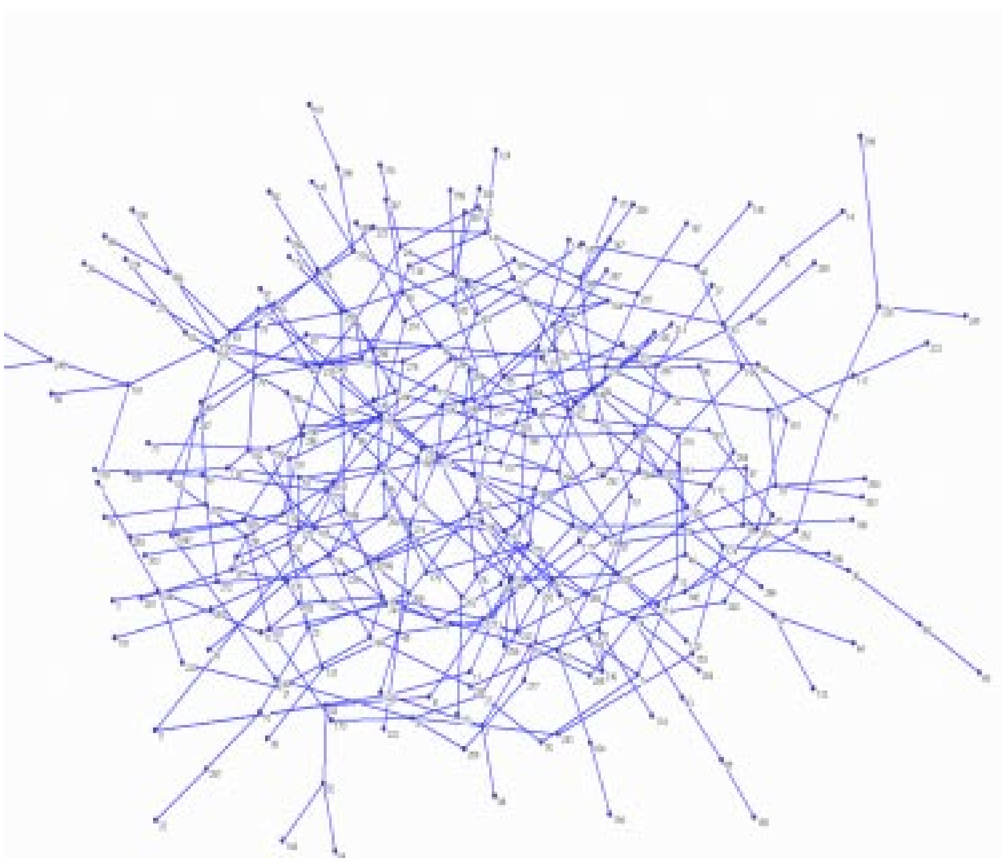
⟶ Internet AS graph (segment of Jan. 2002)



Contrast with random graph: same number of nodes and edges

$\longrightarrow$ random graph: choose each link with prob. p

$\longrightarrow$ independently: prob. of k neighbors is p^k



Phenomenon:

$\longrightarrow$ $\Pr\{u \text{ has } k \text{ neighbors}\} \propto k^{-\alpha}$ $(2 < \alpha < 3)$

$\longrightarrow$ called power-law graph

In contrast to random graph:

$\longrightarrow$ $\Pr\{u \text{ has } k \text{ neighbors}\} \propto p^k$

$\longrightarrow$ probability is exponentially small in k

$\longrightarrow$ UUNET (AS 701) has $>$ 2500 neighbors!

$\longrightarrow$ $>$ 12500 domains in 2002

$\longrightarrow$ probabilistically UUNET should not exist

$\longrightarrow$ so things are not random

What's going on . . .

$\longrightarrow$ connection to airlines?

Ex.: Delta Airlines route map

- $\longrightarrow$ by design: hub and backbone architecture
- $\longrightarrow$ mixture of centralized/decentralized design
- $\longrightarrow$ small system: centralized is good
- $\longrightarrow$ large system: decentralization necessary

Small system with centralized design:

$\longrightarrow$ star topology

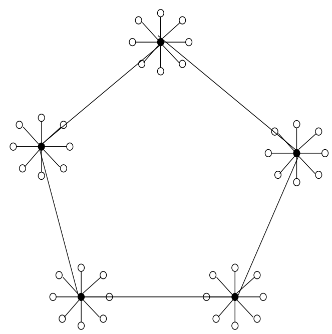
$\longrightarrow$ e.g., Southwest Airlines

$\longrightarrow$ essentially two conjoined star topologies

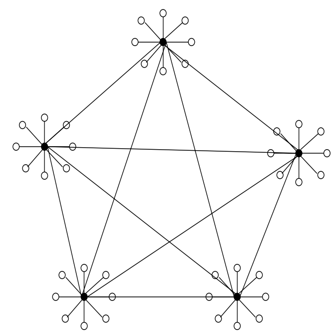
$\longrightarrow$ a matter of load balancing

$\longrightarrow$ backbone topology: trivial

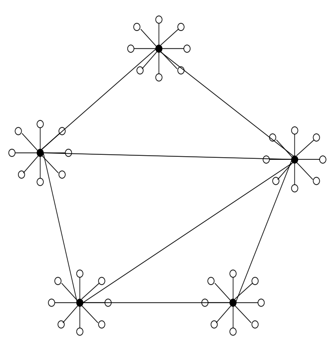
Simple backbone topologies comprised of stars:



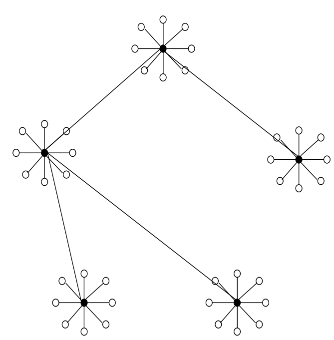
ring of stars



mesh of stars



random/planar backbone of stars



tree (hierarchy) of stars

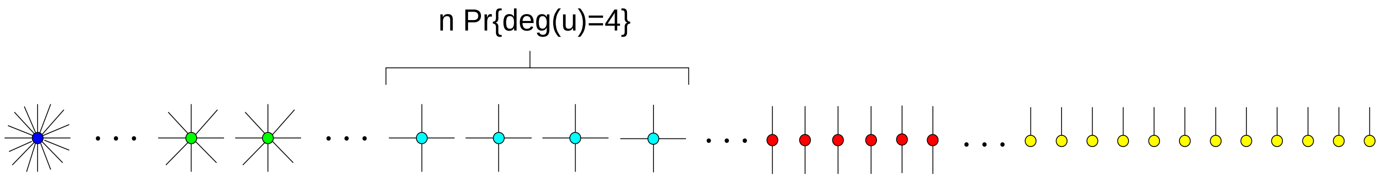
$\longrightarrow$ different star sizes: $\Pr\{\deg(u) = k\} \propto k^{-\alpha}$

$\longrightarrow$ cliques: peering at exchange points

$\longrightarrow$ tier'ed hierarchy

$\longrightarrow$ sparse backbone: random-like

View as "molecular stew" of lego-like building blocks:



$\longrightarrow$ "stir" stew of ingredients until graph is formed

$\longrightarrow$ 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:

$\longrightarrow$ "The rich get richer, the poor get poorer."

$\longrightarrow$ growth process: preferential attachment

$\longrightarrow$ attach to u with probability $\propto \deg(u)$

$\longrightarrow$ 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)