Routing

Brighten Godfrey

CS 538 September 15 2011
Choosing paths along which messages will travel from source to destination.

Often defined as the job of Layer 3 (IP). But...

- Ethernet spanning tree protocol (Layer 2)
- Distributed hash tables, content delivery overlays, ... (Layer 4+)
Problems for intradomain routing

- Distributed path finding
- Optimize link utilization (traffic engineering)
- React to dynamics
- High reliability even with failures
- Scale
Problems for interdomain routing

All of intradomain’s problems

Bigger scale

Multiple parties
- No central control
- Conflicting interests
- Attacks

Harder to change architecture
- Intradomain evolution: RIP, ISIS, OSPF, MPLS, OpenFlow, ...
- Interdomain: BGP.
The two classic approaches

Distance vector & Link state

Far from the only two approaches!

• We’ll see more later...
Distance vector routing

Original ARPANET: distance vector routing

Remember vector of distances to each destination and exchange this vector with neighbors

- Initially: distance 0 from myself
- Upon receipt of vector: my distance to each destination = min of all my neighbors’ distances + 1

Send packet to neighbor with lowest dist.

Slow convergence and looping problems

- E.g., consider case of disconnection from destination
- Fix for loops in BGP: store path instead of distance
Protocol variants

- ARPANET: McQuillan, Richer, Rosen 1980; Perlman 1983
- Intermediate System-to-Intermediate System (IS-IS)
- Open Shortest Path First (OSPF)

Algorithm

- Broadcast the entire topology to everyone
- Forwarding at each hop:
  - Compute shortest path (Dijkstra’s algorithm)
  - Send packet to neighbor along computed path
Link state vs. distance vector

Disadvantages

• Need consistent computation of shortest paths
  - Same view of topology
  - Same metric in computing routes
• Slightly more complicated protocol

Advantages

• Faster convergence
• Gives unified global view
  - Useful for other purposes, e.g., building MPLS tables

Q: Can link state have forwarding loops?
Algorithm:

- Broadcast the entire topology to everyone
- Forwarding at source:
  - Compute shortest path (Dijkstra’s algorithm)
  - Put path in packet header
- Forwarding at source and remaining hops:
  - Follow path specified by source

Q: Can this result in forwarding loops?
Advantages

- Essentially eliminates loops
- Compute route only once rather than every hop
- Forwarding table (FIB) size = \#neighbors (not \#nodes)
- Flexible computation of paths at source

Disadvantages

- Flexible computation of paths at source
- Header size (fixable if paths not too long)
  - Use local rather than global next-hop identifiers
  - \( \log_2(\#\text{neighbors}) \) per hop rather than \( \log_2(\#\text{nodes}) \)
- Source needs to know topology
  - Potentially problematic if source is end-host
Key task of intradomain routing: optimize utilization

**Classic TE:** optimize OSPF weights

- Need to propagate everywhere: can’t change often
- Single path to each destination

**Modern TE:** load balance among multiple MPLS paths

- e.g., TeXCP (Kandula, Katabi, Davie, Charny, 2005)

[Kandula et al, “Walking the Tightrope”, SIGCOMM 2005]
BGP: Border Gateway Protocol

Distance vector variant: Remember path instead of distance ("path vector" instead of "distance vector")

- Avoid DV’s transient loops; but more importantly...
- **Support policies:** can pick any path offered by neighbors, not necessarily the shortest
Common policies

Route selection: prefer customer over peer over provider
Common policies

Route export: “valley-free”: to/from customer only
What’s to come

Today: interdomain routing basics

Upcoming meetings: advanced routing challenges

- scalability
- reliability
- selfishness
- security

By early next week:

- Project comments
- Presentation topics