Link State Routing & Inter-Domain Routing

CS640, 2015-02-26

Announcements

- Assignment #2 is due Tuesday

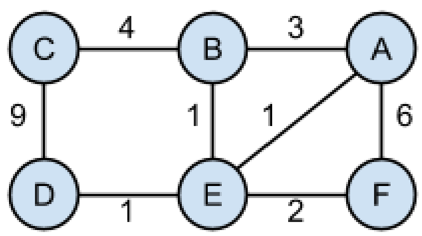
Overview

- Link state routing
- Internet structure
- Border Gateway Protocol (BGP)
- Path vector routing
- Inter-domain routing policies

Link State Routing

- Reliably flood LSPs
    - LSPs
        - Source router ID
        - List of neighbors and link costs
        - Sequence #
        - Time to live
    - Router generates LSP when conditions change -- e.g., link fails
        - Assign next seq #
    - Flooding algorithm
        - If no stored LSP from source
            - Store LSP
            - Broadcast LSP on all links except the link on which it was received
        - If seq # of received LSP > stored LSP
            - Replace stored LSP
            - Broadcast LSP
        - If sequence # of received LSP ≤ stored LSP
            - Do nothing
- Dijkstra's algorithm
    - Variables
        - V = nodes in network G
        - $\ell(i,j)$ = link cost between nodes i & j
        - SPT = shortest path tree between source node and all other nodes
        - S = source node
        - C(v) = cost of path between S and v

- Algorithm
    - Initialize SPT = {S}
    - For each n not in SPT
        - C(v) = ℓ(S,n)
    - While SPT <> V
        - SPT = SPT U {w} such that C(w) is the min for all w in (V - SPT)
        - For each v not in SPT
            - C(v) = cost of minimum path from S to v via nodes in SPT
    - Result is table of entries <From, To, Interface, Cost> for all nodes in the network
- Example -- find the SPT from B



    - SPT = {B}
    - Set C(v) = ℓ(S,v) for all V - SPT
      C(E) = 1
      C(A) = 3
      C(C) = 4
      C(D) = ∞
      C(F) = ∞
    - SPT = {B} U {E}
    - Recalculate C(v) based on SPT = {B,E}
      C(A) = min(3, 1+1) = 2
      C(C) = min(4, 1+∞) = 4
      C(D) = min(∞, 1+1) = 2
      C(F) + min(∞, 1+2) = 3
    - SPT = {B,E} U {A}
    - Recalculate C(v) based on SPT = {B,E,A}
      C(C) = 4
      C(D) = 2
      C(F) = 3
    - SPT = {B,E,A} U {D}
    - Recalculate C(v) based on SPT = {B,E,A,D}
      C(C) = 4
      C(F) = 3
    - ...
    - Algorithm stops when SPT = {A,B,C,D,E,F}

- No count to infinity problem
    - Fast propagation of link state packets via reliable flooding
    - All route computation is local
- Open Shortest Path First (OSPF) Implements link state routing
- Link state routing is preferred due to
    - Fast convergence
    - Loop free
    - Scalable:
        - m = # links, n = # nodes
        - Dijkstra = O(m log n)
        - Bellman-ford (DV) = O(mn)

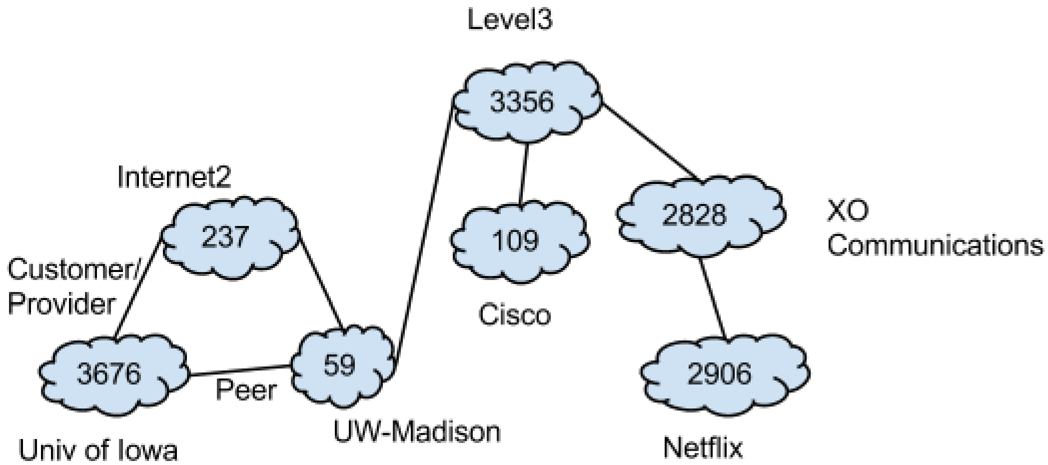
Intra- vs. Inter-domain Routing

Intra-domain	*Inter-domain*
• Within an administrative domain (e.g., campus network) • Emphasis on efficiency -- find *an optimal* path • Used with 10s of routers	• Between administrative domains (i.e., the Internet) • Emphasis on reachability -- find *a* path • Used with 1000s of networks

Internet Structure

- Network == *autonomous system* (AS)
    - Assign unique AS number (ASN) -- UW-Madison is AS 59
    - Some networks (e.g., home networks, small enterprises) let their Internet Service Provider handle inter-domain routing for them, so they are not assigned an ASN
    - ≈ 46K active ASes
- Types of ASes
    - Stub AS -- connect to a single provider
    - Multi-homed AS -- connect to multiple providers
    - Transit AS -- send/receive traffic for nodes within the AS and for other connected ASes
- Types of AS relationships
    - Customer-provider -- customer AS pays provider AS to send/receive traffic on its behalf
    - Peer -- two ASes send/receive traffic between themselves for nodes within their own networks, but not for nodes external to these networks; no money is exchanged

- Example



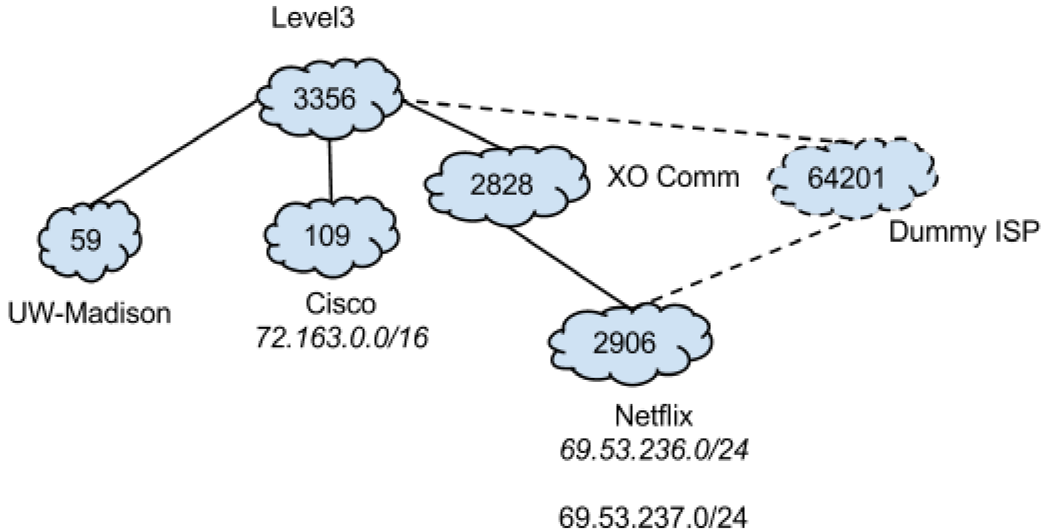
- Internet Exchange Points (IXPs)
    - Physical location where many ASes come together and can set up lots of peering relationships with other ASes
    - Popular in Europe -- one of the largest IXPs has ≈375 ASes, which have established about 50,000 peering links (maximum number of links is ≈375^2)

Border Gateway Protocol (BGP)

- Protocol for inter-domain routing
- Every AS has a *BGP speaker* which sets up a BGP session with each of its neighbors
- Speakers advertise/exchange
    - Local address space -- i.e., IP address range(s) for hosts in this network
    - Other reachable networks -- only speakers for transit ASes do this
        - Provide full path (i.e. list of ASes) used to reach other network
- Send updates when
    - Destination becomes reachable
    - Better path to destination becomes available
    - Best path becomes unusable -- switch to worse path
    - Destination becomes available
- Challenges
    - Scalability -- ≈46K ASes, ≈200K prefixes
    - Conflicting business goals -- transit AS may pick paths based on latency, hop count, monetary cost (i.e., how much they pay another provider), etc.
    - Flexibility
    - Need for trust -- assume that another AS has advertised a valid path
        - Advertisements of false paths can lead to black holes
        - Secure BGP designed to address this, but not widely deployed

Path Vector Routing

- For each network prefix, send the full path of ASes needed to reach that network
- Example



  - Netflix advertises to XO Communications:
    69.53.236.0/24 2906
  - XO Communications advertises to Level3:
    69.53.236.0/24 2828, 2906
  - Cisco advertises to Level3:
    72.163.0.0/16 109
  - Level3 advertises to UW-Madison:
    69.53.236.0/24 3356, 2828, 2906
    72.163.0.0/16 3356, 109
- AS may receive multiple advertisements to reach the same prefix
  - Pick one based on local policy
  - Advertise that path, adding own ASN
  - Example
    - Netflix advertises to Dummy ISP:
      69.53.236.0/24 2906
    - Dummy ISP advertises to Level3:
      69.53.236.0/24 64201, 2906
    - Level3 prefers path through XO Communications, so it advertises that to UW-Madison
- AS may need to advertise multiple contiguous prefixes
  - Can advertise separately or as one aggregate prefix
    - Separate - allows for different paths for different prefixes
    - Aggregate - minimizes number of forwarding table entries (and advertisements)
  - Example
    - Netflix actually owns the prefixes 64.53.236.0/24 and 64.53.237.0/24
    - Netflix advertises to XO Communications and Dummy ISP:
      69.53.236.0/24 2906
      69.53.237.0/24 2906

- Dummy ISP advertises to Level3:
  69.53.236.0/24 64201, 2906
  69.53.237.0/24 64201, 2906
  -OR-
  69.53.236.0/23 64201, 2906
  -OR-
  Just one
- Similar for XO Communications to Level3
- Assume Dummy ISP & XO Communications both choose the 1st option, Level3 can advertise to UW-Madison:
  69.53.236.0/24 3356, 64201, 2906
  69.53.237.0/24 3356, 64201, 2906
  -OR-
  69.53.236.0/23 3356, 64201, 2906
  -OR-
  69.53.236.0/24 3356, 2828, 2906
  69.53.237.0/24 3356, 2828, 2906
  -OR-
  69.53.236.0/23 3356, 2828, 2906
  -OR-
  69.53.236.0/24 3356, 2828, 2906
  69.53.237.0/24 3356, 64201, 2906
  -OR-
  69.53.236.0/24 3356, 64201, 2906
  69.53.237.0/24 3356, 2828, 2906