COMP2207: IP Routing Protocols

The network layer
A host’s view of routing

• A host needs to know how/where to send an IP packet, which will be one of:
  – Send directly to destination if it is on the same local subnet
  – Forward to router if destination IP is not in the local subnet

• Hosts are usually unaware of routes beyond their own subnet
  – They just send locally, or to their default router
  – The subnet router probably participates in a site routing protocol
  – And the site probably routes to the Internet via a policy-rich routing protocol called Border Gateway Protocol (BGP)
Routing domains involved

- Host A
- Site A
- ISP 1
  - Intra-site routing (e.g. RIP, OSPF)
- ISP 2
  - Inter-network routing (e.g. BGP)
- Site B
  - Intra-site routing (e.g. RIP, OSPF)
- Host B
To determine whether to deliver locally or forward to a router, a host maintains a (small) routing table:

- Lists known networks, and how to reach them
- Can be built from information learnt by DHCP or IPv6 RAs

Includes the local subnet in which the host resides:
- Can be determined from IP address and netmask

And a catchall ‘default’ route:
- For any destination not explicitly in the host routing table
Example: local subnet view

- Consider a host 152.78.68.162
- DHCP assigns default router 152.78.68.190 and netmask 255.255.255.192
Example: Linux IPv4 routing table

$ /sbin/ifconfig
eth0 Link encap:Ethernet HWaddr 00:30:48:23:58:DF
    inet addr:152.78.68.162 Bcast:152.78.68.191 Mask:255.255.255.192

$ netstat -nr
Kernel IP routing table
    Destination     Gateway                  Genmask Flags MSS Window  irtt Iface
    152.78.68.128 0.0.0.0                  255.255.255.192   U          0        0          0   eth0
    0.0.0.0         152.78.68.190         0.0.0.0                   UG       0        0          0   eth0

In this example the netmask is 255.255.255.192 (first 26 bits set to 1)
Network address is (255.255.255.192 AND 152.78.68.162), i.e. 152.78.68.128.
192 = binary 1100 0000
162 = binary 1010 0010
AND       1000 0000 = 128
Thus the network prefix is 152.78.68.128/26

The routing table says:
“Put anything for 152.78.68.128/26 out on eth0 (my Ethernet interface)”
“Route everything else (default route is 0.0.0.0) via the router with IP address
152.78.68.190, via eth0 (my Ethernet interface)”
Beyond the default router

• We said that IP packets not delivered locally are sent via the default router
  – What happens then?

• The default router needs to know the next router or subnet/link to forward the packet to
  – May be many hops (routers) from source to destination
  – This is where routing protocols come into play

• Routers exchange reachability information for specific networks (prefixes) using routing protocols
Getting from A to B

Packet forwarded from A to B
Src addr: 152.78.64.3
Dst addr: 191.63.85.6

A
R1
"Internet"
R2
B

Subnet
152.78.64.0/24

152.78.64.3

193.63.85.6

Subnet
193.63.85.0/24

R1 is default router for A’s local subnet
R2 is default router for B’s local subnet

Some set of routers on "the Internet" needs to forward the packets from R1 to R2 such that the packet can be delivered to B
Routing within a site

• Can use static routes
  – Configure all routes manually
  – *Problematic if topology changes or network faults occur*

• Normally use dynamic routing
  – A number of intra-site routing protocols exist, e.g.
    • Routing Information Protocol (RIP) – distance vector algorithm
    • Open Shortest Path First (OSPF) – link state algorithm
    • IS-IS – link state algorithm
Aggregating prefixes

• In principle all Internet routers would need to know the presence of every subnet on the Internet

• In practice, this isn’t the case
• A subnet’s network prefix can be aggregated, ‘hiding’ specific routes to other external networks
• For example, Southampton has the allocation 152.78.0.0/16; external routers only need to know how to route to that prefix, not to each specific /23, /24, /25 or smaller subnet inside Southampton’s network
  – So we just advertise 152.78.0.0/16 to the world
Site (campus) example: aggregation
Routing tables

• Routing is all about the routers maintaining routing tables that tell them for a given destination which interface or next hop router to send an IP packet to

• The routing table in its simplest form has a list of destination IP prefixes, and the interface or next hop to use

• A routing protocol allows router to build or exchange routing information
Types of routing protocol

• Distance Vector
  – Talk only to directly neighbouring routers in a site
  – Exchange best route (shortest distance) information for any known prefixes with direct neighbours
    • e.g. RIP

• Link state
  – Talk to all routers to establish full knowledge of the routers/topology in a site
  – Routers flood information messages describing their connected neighbours around entire site network
    • e.g. IS-IS, OSPF
RIP: Routing Information Protocol

– One of the original IP routing protocols
– Uses hop count as a metric
– RIPng defined to add IPv6 support (RFC 2080)

• Uses distance vector algorithm
  – Router sends whole routing table periodically (every 30 s) in messages to directly connected routers
    • Destination network (prefix) and distance (cost) in hops
  – Receiving routers use messages to update their view of the best route (lowest distance) to a given network (prefix)
Updating routing table entries...

• Each router maintains a RIP routing table
  – Destination network
  – Cost (hop count); 0 if network is directly connected
  – Next hop to use for best path

• If a router receives an update with a lower cost path to a destination network, it updates its entry for that destination

• If a router receives a higher cost path later for the same destination from the same neighbour, the higher value is used
  – Because topologies can change
RIP example, a router’s perspective

RIP advertisements as Prefix, Cost

R1

R2

R3

R4

R5

"Internet"

152.78.64.0/24

152.78.64.0/24 0

152.78.64.0/24 1

152.78.64.0/24 2

Prefix advertised from R2 to R3 is ignored as higher cost (2 vs 1) than R1 to R3

R1 is default router for 152.78.64.0/24
RIP limitations

• Updates only sent every 30 seconds
  – Thus “news” takes time to travel
  – Updates are not acknowledged (UDP); so a possibility for message loss

• Metrics are simple hop count values
  – Limited to max value of 15 - a value of 16 means unreachable
  – Thus only suitable for smaller homogeneous networks
  – Can’t express any richer policy or metrics, just hops
  – Problematic if want to weight certain links with higher priority

• Can be slow to converge to a steady state

• Routers don’t have knowledge of network topology
  – Just what their direct neighbours tell them
  – Can lead to ‘count to infinity’ problem – see Tanenbaum
Link state routing

• De facto enterprise routing algorithm
  – Campuses will typically run one of IS-IS or OSPF

• Steps:
  1. Discover neighbours
  2. Determine cost metric to each neighbour
  3. Construct link state information packet
  4. Flood this message to all site routers in same area
  5. Use messages to build topology, and then compute shortest paths for prefixes served by any given router

• All routers learn the full topology
Discovering neighbours

- Send a broadcast message on all interfaces
  - Adjacencies built by received “hello” messages
  - Then measure line cost
  - Typically based on bandwidth/delay
Building link state packets

- Each router creates link state packets based on neighbours and costs to reach them

(a)
Distributing link state

• Flood link-state packets around the site network
• Link state packet includes (Src ID, Seq, Age)
  – Source ID to uniquely identify the node
  – Sequence to allow receiver to determine if message is a new one to process and flood, or one to discard
    • Sender increments Sequence number with each message
    • Floods on all links except one received on
  – Age
    • Decrement once per second
    • Prevents old messages persisting in the network
  – All messages are acknowledged to senders
Computing shortest path tree

• Once all routers have the topology, we need to compute paths
• Uses Dijkstra’s algorithm (from 1959!)
  – Determines shortest path through a graph from an initial node to any given destination
  – Algorithm works by expanding from the starting node, considering cheapest neighbour with each iteration
    • Thus the algorithm finds the next ‘closest’ node with each iteration, until it reaches the destination
  – similar algorithm on sat navs, to really get from A-B!
In more detail...

• Start by setting the tentative cost to reach all nodes from initial node to infinity, and cost of reaching initial node to 0
• Mark all nodes unvisited
• For the current node
  – For all unvisited neighbours of the current node
    • If the cost via the link from current node to the neighbour is less than the existing tentative cost, update the cost
  – Mark the current node as visited
  – If destination node is visited, stop
  – Else make the node with lowest cost the current node and repeat the iteration
• Shortest path can be found by following the reverse path
  – Having noted path the cheapest cost update came from
Routing between sites

- Inter-site routing uses exterior routing protocols
  - You advertise your network prefixes to neighbouring networks
  - You may or may not offer transit to other networks
  - Policy is often more important than path costs

- De facto protocol is **Border Gateway Protocol (BGP)**
  - Works between **Autonomous Systems (AS)**
  - Each ISP has a unique AS Number (ASN), assigned by RIRs
    - AS numbers are up to 32 bits long
  - It’s distance vector-like, but routing in BGP includes information about the AS path associated with using a given route, the costs of the paths, and many other richer attributes
    - These attributes enable policy to be applied
BGP AS paths example

Information F receives from its neighbors about D

From B: "I use BCD"
From G: "I use GCD"
From I: "I use IFGCD"
From E: "I use EFGCD"
BGP operation

• In configuration, specify IP of neighbour and AS, e.g.
  – `neighbour 152.40.30.20 remote-as 3456`
  – Creates a BGP peering session (over TCP, port 179)
    • Initially sends whole routing table, then incremental UPDATEs
  – Then you advertise routes you know of to your neighbours
    • Contains network prefix, prefix length, AS path, next hop, ...
  – Neighbour may then choose whether to use that route
    • Can use knowledge of ASNs to decide whether to accept/use the route

• BGP is loosely distance vector, but includes many improvements
  – e.g. you know the full path, you can detect loops if your own AS is on a path you receive
BGP examples

• Looking glass types functions
  – For example see http://www.bgp4.as/tools
  • Views of a given ASN from a selection of others
  • e.g. JANET is ASN 786
  • In practice, our university doesn’t use BGP, as it only has one
    route to JANET, so therefore has no routing decision to make

• Other interesting BGP data:
  – www.potaroo.net: adjacencies, prefixes announced
  – http://www.cidr-report.org/cgi-bin/as-report?as=as786&view=2.0
  – http://bgp.potaroo.net/bgprpts/rva-index.html
BGP IPv4 route growth
ASN growth
Average ASN path length
More reading

• **Autonomous Systems**
• **BGP, IS-IS, ASN,**
• Tanenbaum chapter 5 – essential reading!
• If you’d like to find out more about Dijkstra’s algorithm **watch this**