ICS 351: Today's plan

- netmask exercises
- network and subnetwork design
- dynamic routing
- RIP
- distance-vector routing

Netmask exercises

- how many bits in this netmask: 255.128.0.0
- using this netmask and the address 10.127.66.100, show the network number and broadcast address
- repeat with the netmask 255.252.0.0 (which has how many bits?)
- and 255.255.248.0, 255.255.224.0, 255.255.255.192
- Is 255.128.192.240 a valid netmask?

Network and Sub-network design considerations

- Private networks (behind NATs) can be designed without too much concern for efficient allocation
- For public (routable) IP addresses, we obtain a range of IP addresses
- must decide how to subdivide them
- usually, the decision of which hosts to attach to which router interface is dictated by connectivity, security, or similar factors
- so must decide which subnetwork gets which addresses
- each subnetwork must have a size that is a power of two (e.g. 64 in prelab 3)
- two addresses in each subnetwork are reserved: all zeros host part gives the network number, all ones host part gives broadcast address
- so if a host part has n bits, the subnetwork can have at most $2^n - 2$ hosts

Network and Sub-network design requirements

- each host in each subnetwork needs a unique IP number
- these can be assigned statically or dynamically
- server machines usually need static IP addresses, whereas client machines can use static or dynamic IP addresses
- static IP addresses must be assigned manually, changing them require overhead and sometimes causes user complaints
- dynamic IP addresses are assigned by DHCP, can be changed relatively easily
- (usually) we cannot easily get new IP numbers, and the future can be uncertain!
- so many people request more than they need

Rational Sub-network Design

- figure out network, subnetwork, and sub-sub-network (etc) topology
- figure out the minimum current size for each subnetwork
- if additional addresses are available, estimate future growth of each subnetwork
- dynamic addresses can be reassigned more easily, so it is OK to be more generous with dynamic addresses
- if the future is uncertain, adopt a reasonable policy, e.g. all subnetworks have 27-bit netmasks and subnetworks numbers are handed out as needed

Network Design Exercise

- three routers, A, B, and C, each connected to the other two
- router A is also connected to the wider internet
- router A connects to three other subnetworks:
- o X, with at least 10 static addresses and 20 dynamic addresses,
- o Y, with at least 50 dynamic addresses,
- o and Z, with at least 2 static addresses.
- router B connects to two other subnetworks:
- o H, with at least 5 static addresses but a good chance of future growth, and
- o K, with at least 30 dynamic addresses and 30 static addresses
- router C connects to two other subnetworks:
- o P, with at least 5 static addresses
- o Q, with at least 5 dynamic addresses
- You have been given the IP address range 10.11.12.0/24
- assign IP addresses for this network

dynamic routing

- in lab 3, routes are set by hand, which is static routing
- when using proxy arp, the router automatically decided to ARP for another network to which it had a route (after proxy arp was enabled)
- it is also possible to build routing tables automatically: dynamic routing
- this is because once it is configured, each router knows to which networks it is connected
- if each router distributes this information to all other routers, every router can build a corresponding routing table

dynamic routing example

- router 1 connected to 10.0.1/24 and 10.0.2/24
- router 2 connected to 10.0.2/24 and 10.0.3/24
- router 1 could tell router 2 how to reach 10.0.2/24, and router 2 could tell router 1 how to reach 10.0.3/24
- this is all automatic
- for any correct routing protocol, the only questions are:
- o how long does the network take to reach a consistent state again (to converge) after a change? (e.g. 30s, 10min)
- o how much network traffic does the routing protocol add? (e.g. 10Mb/s/router)

RIPv2

- Routing Information Protocol, version 2 (v2 supports network masks)
- RIP generally used within a single Autonomous System (AS), i.e. within an organization (IGP, Interior Gateway Protocol)
- reliably finds shortest paths in networks that don't change very often
- links may have a "metric" (cost) associated with them, or "1" may be used for each link, but in any case metrics should be static
- limited to networks with a maximum metric (distance, cost) of 15 between any two nodes
- somehow, router must be configured to know which interfaces to run RIP on
- defined in RFC 2453, http://tools.ietf.org/html/rfc2453

distance-vector routing

- Bellman-Ford, applying Bellman's equation in the Ford and Fulkerson algorithm
- the basic idea is that each router sends its routing table to all other routers on the same network
- when receiving such a message (routing update), a router must update its own routing table:

distance-vector algorithm

- when receiving a routing update, a router must update its own routing table:
- o any new route is simply added, with the next hop being the router that sent us the routing update, and the metric being the metric received plus the metric of the link over which it was received
- o any existing route is replaced if the new route has better metric (after adding the link metric)
- o any existing route is replaced if the routing update comes from the router listed as next hop
- routes through a given next hop G are *timed out* if no routing updates are received from G within a certain period (about 6 times longer than the 30s routing update time)
- if a route is deleted, it is actually marked as having an infinite cost (for RIP, infinity is 16)
- also, when sending a routing update to neighbor G, "poison" routes through G by giving them infinite metric (an algorithm known as **split horizon with poisoned reverse**)

RIPv2 details

- routing tables sent to multicast address
- split horizon is required, poisoned reverse is optional
- whenever the routing table changes (e.g., as a result of an interface changing, or of receiving a routing update), a triggered update is sent out, perhaps with only the new route(s)
- triggered updates should be limited to about 1 every 5 seconds
- regular updates are sent every 30 seconds

counting to infinity

- suppose that a router R loses one of its connections (to network N), but does not send a triggered update, and does not do split horizon
- then, its neighbor G will tell it about a route to N with metric 2
- R will use a route to N with metric 3, next hop G
- at the next update, G will know R has a route to N with metric 3, will have a route with metric 4, next hop R
- then R updates its route to N to have metric 5
- then G updates its route to N to have metric 6
- and so on
- this only stops when one of the routers reaches 16 (infinity), and could potentially take a long time to delete bad routes