COMP3721 Week Twelve Lab Synopsis

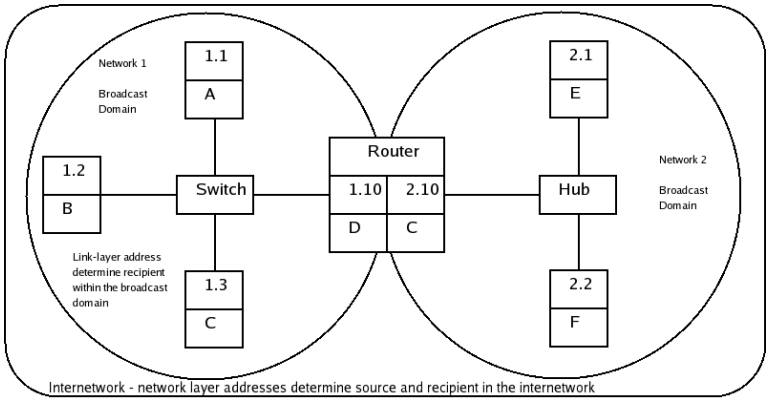
Network Addressing

- *Data-link addresses* (such as the MAC address in Ethernet) determine the recipient of a frame within a *broadcast domain*
  - the data-link address must then be unique within the broadcast domain
  - note that data-link addresses are not necessarily required. For example, in a point-to-point network, the source and destination are implicit
- The source and destination of a packet within an *inter-network* are specified with *network-layer addressing*
  - network addresses are normally composed of two components
    - a network id
      - if two hosts share the same network id, they are then within the same broadcast domain and can communicate directly (via a switch/hub)
      - if two hosts have different network ids, they can only communicate indirectly through a router (or series of routers)
    - a host id – the portion of a network address that identifies a specific host within a network
  - Communication is normally premised on the user providing the network-layer address
  - For example, to access a website, you enter a hostname (i.e. www.microsoft.com). That hostname is resolved to a network address by the web browser through a protocol called DNS. From then on, all communications is based on IP addresses (TCP/IP network layer).

Simplified Addressing Example

- As an example of these two layers of addressing and the role of each consider the following example
  - Data-link addresses are represented by letters: A, B, C, ...
    - A special case is the broadcast data-link address, BRDCST, that specifies a frame should be delivered to all nodes within the broadcast domain (network in TCP/IP terminology)

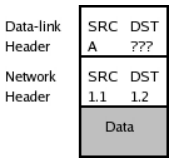
- Network-layer addresses have the following form:
  - #.#, i.e. 1.1, 3.2, where the number preceding the period is the network id and the number trailing the period is the host id.



- Note that the link-layer address C appears in both networks. This is not a problem as long as it it unique within each network (broadcast domain)

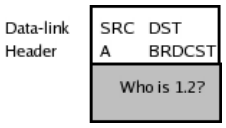
Scenario One – 1.1 sends a packet to 1.2

1. Host 1.1 must determine whether the packet must be routed or not
   1. the network ids match (1 == 1) so the packet is local and need not be routed
2. To deliver the packet locally, the packet must be constructed with all addressing information

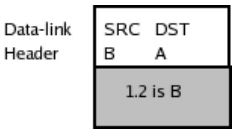


- The destination data-link address is unknown. The packet could be broadcast in which case the recipient (B) would receive it. But so would all other machines in the broadcast domain and switches would not be able to provide any advantage over hubs
- A protocol is required that can resolve a network layer address to the appropriate data-link address. In TCP/IP this is called ARP (address resolution protocol)

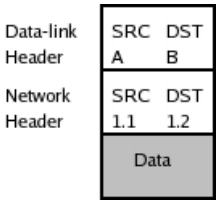
3. An ARP request is sent:



4. B receives the ARP request (it was broadcast to everyone in network 1), sees the request for network address 1.2, generates and sends an ARP response



5. Host 1.1 completes the original packet by filling in the destination link-layer address and sends the packet

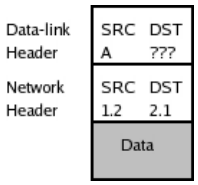


- the whole processes of ARP does not seem efficient on the face of things
  - it took three packets (ARP request, ARP response, data packet) to transmit one original packet
  - the ARP request had to be broadcast to all network nodes
- in reality, hosts will cache the ARP response for a period of time to avoid ARPing for every packet
  - you can check the ARP cache on your machine with the 'arp' command. Try 'arp –help' from the command prompt

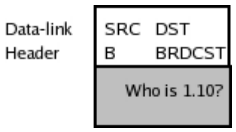
Scenario One – 1.2 sends a packet to 2.1

1. Determine whether the packet must be routed or not
   1. the network ids are different so the packet must be routed (sent to the router rather than the final destination)
   2. note that if 1.2 had tried to use the local delivery process, it would have to ARP for 2.1. But 2.1 would not see the ARP as the router acts as a broadcast boundary and thus the packet would not be delivered.

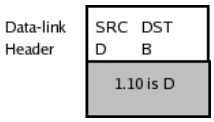
2. Construct a packet for delivery to the router
    1. the network-layer addresses will specify the inter-network source and destination
    2. the link-layer addresses will specify the network source machine (1.2 on this hop) and destination (the router)



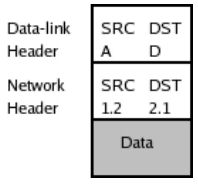
        - again, the link-layer address is unknown (the data link address of the router in this case)
3. An ARP request is created and sent. The host however must know the network address of the router, or it cannot create the ARP request. In TCP/IP, the router address is a configuration item known as the *default gateway*



4. The router sends back an ARP response



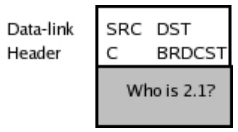
5. The data packet is completed and sent within the link



6. The router receives the packet. The role of a router is to move packets towards their network-layer destination.
    - In this case, the router can easily determine the correct action – the destination address is in network 2 and the router is directly connected to network 2 – thus we have a local delivery scenario

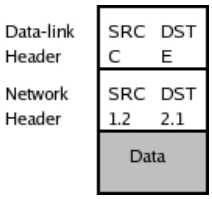
- In reality, each router maintains a routing table, populated by a routing protocol, that determines the correct link to send a packet out on (we will cover routing protocols later)

7. The router ARPs for 2.1 on network 2



8. Host 2.1 sends back an ARP response

9. The packet is forwarded to the network-layer destination



- Note how the addresses at the link-layer need not refer to the same machine(s) as the network-layer addresses. Each addressing layer is serving a separate purpose.

TCP/IP Addressing – IP version 4 (IPv4)

- IPv4 addresses are 32-bit values
  - normally written as four 8-bit decimal values, each separated by dot (quad-dotted format) – i.e. 192.168.4.12
- 32-bit address subdivides into two components:
  - network id
  - host id
- The division between network id and host id is determined by the subnet mask
- The subnet mask is also a 32-bit value
  - it functions as a binary bitmask to isolate the network id portion of the address

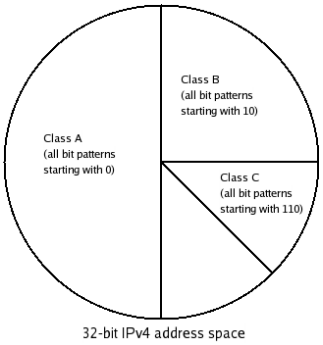
- example:
  - `IP: 192.168.4.12  11000000 10101000 00000100 00001100`
  - `Mask: 255.255.255.0  ^11111111 11111111 11111111 00000000`
  - `Net id: 192.168.4.0 <-- 11000000 10101000 00000100 00000000`
- the network mask is always a number of contiguous 1s followed by contiguous 0s
- this allows for a shorthand notation (CIDR notation)
  - 255.255.255.0 is 24 1s followed by 8 0s and can be abbreviated as /24
  - this notation also makes it easy to see where the split lies between network id and hosts id
    - /8 = 8-bit network id, 24-bit host id
    - /16 = 16-bit network id, 16-bit host id
    - /24 = 24-bit network id, 8-bit host id

- just as in the earlier example on network addressing, whether a packet is routed or not depends on whether the source and destination network ids match
  - example:
    - source:
      - IP: 192.168.4.40/23
      - ```
        IP:       11000000 10101000 00000100 00101000
        Mask:    ^11111111 11111111 11111110 00000000
        Network:  11000000 10101000 00000100 00000000
                       192.     168.       4.       0
        ```
    - destination1:
      - IP: 192.168.5.199
      - ```
        IP:       11000000 10101000 00000101 11000111
        Mask:    ^11111111 11111111 11111110 00000000
        Network:  11000000 10101000 00000100 00000000
                       192.     168.       4.       0
        ```
      - same network id – no routing necessary
    - destination2:
      - IP: 192.168.3.254
      - ```
        IP:       11000000 10101000 00000011 11111110
        Mask:    ^11111111 11111111 11111110 00000000
        Network:  11000000 10101000 00000010 00000000
                       192.     168.       2.       0
        ```
        ```

- different network ids – this packet must go through a router (generally the default gateway) to get to the destination network (and machine)

IP Address Classes

- There is a single pool of 2^{32} unique IPv4 addresses available
- different users have different requirements in terms of the number of hosts they must support
- the 32-bit address space is then divided into orthogonal (separate, non-overlapping) classes



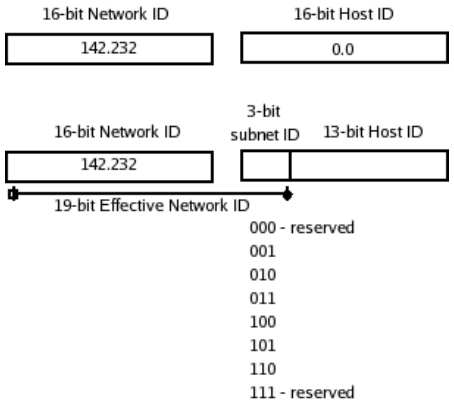
- Class A
  - Network ID must start with a 0_2 (this then consumes ½ of the total address space)
  - Network mask is /8
  - Of the 8 network ID bits, 7 are then variable (the first bit is fixed as a 0) and thus there are 2^7=128 potential classes A networks that can be allocated
  - Two of those 128 ids cannot be used (leaving 126)
    - Network ID cannot be all 0s, so that removes 1 possible class A network ID
    - Network ID 127 is reserved for loopback
      - packets sent to loopback never leave the machine, they simply loop back to the same interface they left on (useful for development)
    - Usable class net ids then range between 1 and 126
      - 10.0.0.0/8 is actually reserved as well for private networks)

- 24 host id bits allows for 2^{24}-2 = ~ 16 million hosts on a single network
  - -2 because the host id cannot be all 0s or all 1s
  - The design of TCP/IP would require all of these hosts to be within a single broadcast domain to use the network id “as is” (with the 8-bit mask)
  - this is far too large for a single broadcast domain
    - class A networks are thus broken down into multiple smaller networks through sub-netting (see below)
- Class B
  - Network ID must start with a 10_2 (this then consumes ¼ of the total address space and clearly does not overlap any of the class A addresses)
  - Network mask is /16
  - Of the 16 network ID bits, 14 are then variable (the first two bits are fixed as 10) and thus there are 2^{14} = 16384 potential classes B networks that can be allocated
    - Usable class net ids then range between 128.0 and 191.255
      - 172.16.0.0/16 is reserved for use on private networks
  - 16 host id bits allows for 2^{16}-2 = 65534 hosts on a single network
    - The design of TCP/IP would require all of these hosts to be within a single broadcast domain to use the network id “as is” (with the 16-bit mask)
    - this is far too large for a single broadcast domain
      - class B networks are thus broken down into multiple smaller networks through sub-netting (see below)
- Class C
  - Network ID must start with a 110_2 (this then consumes 1/8 of the total address space and clearly does not overlap any of the class A or B addresses)
  - Network mask is /24
  - Of the 24 network ID bits, 21 are then variable (the first two bits are fixed as 110) and thus there are 2^{21} = ~ 2 million potential classes C networks that can be allocated
    - Usable class net ids then range between 192.0.0 and 223.255.255

- 192.168.x.0/24 are reserved for use on private networks
- 8 host id bits allows for 2^8-2 = 254 hosts on a single network
  - this is too small for many organizations (a well-designed, switched network (broadcast domain) can easily scale to 1000 hosts
    - this limitation can be overcome by merging several contiguous class C networks into a single larger network (supper-netting – see below for a brief introduction)

Sub-netting

- Break a single (large) network address into multiple smaller networks (sub-nets)
- Accomplished by 'stealing' host id bits
- Example:
  - if you were given the class B address 142.232.0.0/16 and wanted to break it into 4 smaller networks
  - by stealing host id 2-bits, we get 4 unique, sub-net ids
    - however all 1s and all 0s are both reserved, so 2 subnet bits in reality only gives us 2 sub-nets
  - so instead, we borrow 3 bits which gives us $2^3 - 2 = 6$ working subnet ids



  - Subnetworks:
    - Subnet1: 142.232. 001 00000_2 . 00000000_2
      - 142.232.32.0/19
    - Subnet2: 142.232. 010 00000_2 . 00000000_2

- 142.232.64.0/19
- Subnet3: 142.232. 011 00000_2 . 00000000_2
  - 142.232.96.0/19
- Subnet4: 142.232. 100 00000_2 . 00000000_2
  - 142.232.128.0/19
- Subnet5: 142.232. 101 00000_2 . 00000000_2
  - 142.232.160.0/19
- Subnet6: 142.232. 110 00000_2 . 00000000_2
  - 142.232.192.0/19
    - What is the first IP address that can be allocated on this (sub-)network?
      - The 13 host id bits will be 00000 00000001
      - 142.232. 110 00000_2 . 00000001_2 =142.232.192.1
    - What is the last IP address that can be allocated on this (sub-)network?
      - The 13 host id bits will be 11111 11111110
      - 142.232. 110 11111_2 . 11111110_2 =142.232.223.254
    - How many hosts (IP addresses) can be allocated to this sub-net?
      - 13 host id bits --> $2^{13} - 2 = 8190$ hosts