CPS221 Lecture: The Network Layer

last revised 10/9/2014

Objectives

1. To discuss IP numbering (both IPv4 and IPv6)
2. To discuss routing strategies and ARP
3. To explain DNS
4. To explain DHCP
5. To explain NAT
6. To discuss the coming transition to IPv6

Materials:

1. Projectable of packet flow through multiple nodes
2. Projectable of a system with a router (F6.13)
3. Projectable of a routing table (FT6.1)
4. Projectable of domain name structure (F19.2)
5. Projectable of generic domain labels (F19.1)
6. Projectable of hierarchy of DNS servers (F19.6)
7. Projectable of domain name resolution (F19.12)
8. IPv6 day podcast

I. Introduction

A. The next layer up in the network stack deals with end-to-end communication between two hosts, using the services of the link layer.

1. Recall that, at the link layer, the units of information transmitted are called frames. In this layer, they are called datagrams.

2. On the sending host, the network layer software receives a packet from the upper layers, encapsulates it as a network layer datagram, and arranges for it to be delivered to the receiving host.

3. On the receiving host, the network layer software receives a network layer datagram from the link layer, strips off the network layer datagram encapsulation, and delivers it to the upper layers of the protocol stack.

4. Network layer functionality is also present on the intervening nodes, which are just responsible for passing on the datagram to the next node (the next hop). Note, though, that layers above the network layer are not involved (and are generally not even present) on the intervening nodes.

5. PROJECT: Diagram showing packet flow through multiple nodes

6. Conceptually, network layer functionality is a software function - but in practice on the intermediate nodes it may be performed by specialized hardware known as routers.

B. In TCP/IP, this layer is also called the IP layer.

C. There are several major issues concerning this layer we need to discuss.

1. The mechanism used for addressing systems at this level - IP numbers.

2. How routing is done

   (This includes how IP numbers are translated into link layer physical (MAC) addresses.)

3. How symbolic domain names (e.g. www.cs.gordon.edu) are mapped to IP addresses.

4. How an IP address is actually assigned to a system

5. A widely-used strategy known as Network Address Translation (NAT)

D. We will also say a bit about the coming transition to a new version of the Internet protocols called IPv6, which will have significant impact on this layer.

II. IP Addresses

A. As you know, addressing at this level is specified by the system of IP numbers.

1. We tend to think of an IP number as being associated with a computer (host).

    a) But actually, it is associated with a physical interface.

    b) On a computer with multiple interfaces, each active interface will have its own IP number

    SHOW on laptop (System Preferences | Network)

    c) However, for simplicity we will often speak about a host's IP - by which we mean the IP of its interface (or of the interface being used in a given transaction if there is more than one)

2. Conceptually, an IP number is a 32 bit binary number - but it is conventionally written as 4 decimal numbers in the range 0..255 (dotted octet notation)

3. An IP number consists of two parts: a prefix that identifies a particular network (the most significant bits) and a host number that identifies a particular host on the network.

    A consequence of this is that if a computer is moved to a different physical location that is on a different network, its IP number will have to change. (Recall the analogy we used earlier: the MAC address of an interface is like a person's SSN, which doesn't change no matter where he moves; but the IP is like a postal address which changes when the person moves.)

B. While most of the possible IP numbers are available to identify hosts, certain numbers are reserved for special purposes:

1. Addresses whose first octet is 127 are loopback addresses - i.e. they refer to the same computer.

   a) Though any address beginning with 127 can be used, the convention is to use `127.0.0.1`.

   b) Conventionally, the hostname `localhost` is associated with this IP.

2. Addresses in the ranges `10.0.0.0` to `10.255.255.255`, `172.16.0.0` to `172.31.255.255`, and `192.168.0.0` to `192.168.255.255` are reserved for internal use within a network and will never appear in "the outside world". (BTW: note that all of the IPs used on Gordon's network are in one of these blocks. We'll talk about how this is handled later.)

3. Addresses in the range `169.254.0.0` to `169.254.255.255` are reserved for autoconfiguration.

4. Addresses whose host number portion is all 1's (255) are reserved for broadcast messages to all nodes on the appropriate network - e.g. `192.168.255.255` would be used for a message directed to all hosts on the network identified by the prefix `192.168`.

C. Of course, if an IP address consists of a prefix that identifies a network and a host specifier within that network, the question arises as to where the line is drawn.

1. Originally (pre 1993), this was handled by a scheme of network classes. (Classful addressing)

a) Class A networks use 8 bits (the first octet) to specify the network and 24 bits (the remaining 3 octets) to specify a particular host. (A class A network could therefore have nearly 2^{24} = over 16 million distinct hosts).

b) Class B networks use 16 bits (the first 2 octets) to specify the network and 16 bits (the remaining 2 octets) to specify a particular host. (A class B network could therefore have nearly 2^{16} = 65536 distinct hosts).

c) Class C networks use 24 bits (the first 3 octets) to specify the network and 8 bits (the remaining octet) to specify a particular host. (A class C network could therefore have nearly 2^8 = 256 distinct hosts).

d) The first few bits of the first octet specifies what class an address belonged to.

(1) If the first bit was 0 (IP's in the range `0.0.0.0` to `126.255.255.255` - recall that 127 as a prefix is reserved), the address was interpreted as that of a host on a class A network.

(2) If the first two bits were 10 (IP's in the range `128.0.0.0` to `191.255.255.255`) the address was interpreted as that of a host on a class B network.

(3) If the first three bits were 110 (IP's in the range `192.0.0.0` to `223.255.255.255`) the address was interpreted as that of a host on a class C network.

(4) IP's whose first three bits were 111 (`224.0.0.0` on up) were used for other purposes (so-called "class D" and "class E" addresses)

2. However, this scheme of classes was very wasteful of possible addresses. For example, suppose a network had 400 hosts. It couldn't use class C addresses, but giving it a class B block of addresses would result in wasting over 60,000 possible addresses!

3. In 1993, a scheme called CIDR - Classless Interdomain Routing - was adopted.

    a) In this scheme, a network could be divided into subnets, with additional prefix bits used to specify the subnet.

       For example, a class B network might be divided into 16 equal-size subnets, with 20 bits used as a prefix to specify the network (16 bits) and subnet (4) and the remaining 12 bits used to specify a host within a particular subnet. (Maximum of almost 2^{12} = 4096 hosts per subnet).

    b) It is not necessarily the case that the subnets will be of equal size - e.g. a class C block (256 addresses in all) might be divided into one subnet encompassing 128 addresses, and two encompassing 64 addresses each.

        (1) The larger subnet would use a 25 bit prefix - 24 to specify the basic class C block and one to specify the subnet - leaving 7 bits to specify one of 128 (2^7) hosts.

        (2) The two smaller subnets would use 26 bit prefixes - thus leaving 6 bits in each case to specify one of 64 (2^6) hosts

    c) Of course, now to interpret an address one needs to know both the address and the number of bits of it which are used as a prefix. Conventionally this is written as address/prefix bits - e.g. at the time I was writing this lecture the IP for my laptop was `172.16.13.94/16` - that it is, it is host `13.94` on the network having the prefix `172.16.`

    d) Given this information, it is possible to determine the range of IPs that are part of the subnet the address belongs to.

        (1) For example, `172.16.13.94/16` belongs to a subnet encompassing 2^{16} (65536) addresses in the range `172.16.0.0` to `172.16.255.255`.

(2) On the other hand, if the address were specified as `172.16.13.94/21`, it would be understood as referring to one of 2^{11} (2048) hosts in the range `172.16.8.0` to `172.16.15.255`.

(3) (In all cases, the highest address in the subnet range is the broadcast address for the subnet - so, for example, `172.16.15.255` is the broadcast address for a subnet of which `172.16.13.94/21` is a part.

e) Given the number of bits in the prefix, it is also possible to calculate a subnet mask - which is an IP like number that has 1's in positions corresponding the prefix and zeroes elsewhere.

For example, the subnet mask for the subnet that `172.16.13.94/16` belongs to is `255.255.0.0`. (This mask can be bit-wise anded with the an IP address [equivalent to C/C++/Java &] to extract the prefix from the IP - e.g. `172.16.13.94` anded with `255.255.0.0` is `172.16.0.0`.

D. The use of CIDR eliminated much of the waste of IP numbers that pure Classful addressing created. Still, even with CIDR and another strategy we will look at later (NAT), it remains the case that there are not enough IP numbers to meet the need of assigning an IP to every device that can be connected to the internet. (In fact, in February 2011 the last unassigned /8 block of addresses was assigned to a Regional Registry)

For this reason and others, a new set of internet protocols call IPv6 is emerging. Whereas IPv4 uses 32 bits for an IP number, IPv6 uses 128 - which will not run out in the foreseeable future, since in IPv6 there are 2^{128} possible IPs = over 100 trillion.

(More on the transition to IPv6 later)

III.Routing

A. One of the major responsibilities of the network layer is end-to-end routing of packets. (This contrasts with the link layer, which is only responsible for getting a packet to a neighbor to which it is physically connected.) This is, of course governed by the IP of the destination system.

B. There are two cases

1. The destination system is on the same subnet as the sending system.

2. The destination system is on a different subnet

3. The sending system can easily distinguish these two cases by comparing the subnet prefix portion of itself and the destination IP

C. In the case of sending to another node on the same subnet, direct delivery can be used.

1. In this case, the sending system is physically connected to the destination (via a shared channel such as ethernet or wireless, or via a network of switches).

2. Therefore, all the network layer needs to do is to instruct the link layer to physically deliver the datagram to the correct physical address.

3. Of course, this begs an important question. The network layer identifies interfaces by IP numbers, while the link layer uses MAC addresses. How does the network layer resolve an IP to a MAC address for use by the next layer down?

a) The answer is a protocol called Address Resolution Protocol (ARP).

b) To resolve an IP to a MAC address, the network layer broadcasts a special kind of packet to the subnet (using the broadcast address for the subnet), in effect saying "I need the physical address for the interface that has such and such IP".

c) This broadcast is ignored by all interfaces on the subnet except the one using the requested IP, which responds with its MAC address.

d) To avoid needing to make repeated ARP requests, the software on the sending system can cache a physical address for a period of time.

D. But how does the network layer handle delivery of a datagram to a node on a different subnet?

1. For this to work, a subnet needs a designated Gateway system, which actually has at least two interfaces connected to two different subnets - including either a "higher level" subnet or another gateway.

2. Typically, this gateway service is provided by a router - which is a node that only handles the three lowest layers of the protocol stack. (Though it could also be handled by a general-purpose computer running suitable software).

3. The basic idea is this: datagrams that can be delivered directly on the local subnet are delivered this way, while datagrams destined for elsewhere on the Internet are sent to the gateway, which in turn forwards them in the correct direction. (The IP number remains that of the endpoint host, but the physical address to which the datagram is sent is the router.

PROJECT: Example of a router

4. The router's operation is governed by a routing table. When multiple entries might match, the entries are ordered using the "longest matching prefix" rule.

PROJECT: Example routing table

5. Consider each of the following possibilities:

a) Host `180.70.65.135` sends a packet to `180.70.65.136`. It can do this directly, since both are on the same subnet.

b) Host `180.70.65.135` sends a packet to `201.4.16.37`. It sends the datagram to the router, which in turn forwards it over its interface m1.

c) Host `180.70.65.135` sends a packet to `202.12.34.56`. It sends the datagram to the router, which in turn passes it on to via interface to `180.70.65.200`, which in turn forwards it on to its next step.

IV.DNS

A. The transport and network layers of the protocol stack require that hosts be specified by IP address; but applications typically reference servers by URL - which typically includes a domain name plus often other information specifying a specific file.

   Example: The URL for these lecture notes will be www.cs.gordon.edu/courses/cps221/lectures-2014/The%20Network%20Layer.pdf

   1. www.cs.gordon.edu is the domain name

   2. courses/cps221/lectures-2011/The%20Network%20Layer.pdf names a specific file

B. Some mechanism is needed to map from a domain name to the corresponding IP address.

   1. In the earliest days of the Internet, each system maintained a file (called hosts on Unix systems) that listed the names and IP numbers of systems it needed to know about.

      Example: Show /etc/hosts on a workstation

   2. But this system is obviously inadequate for several reasons

      a) On the global internet, a given host may need to contact many different hosts around the world for things like web access and mail. The size of the file that would be needed for this would be huge, and maintaining it manually would be impossible.

      b) Moreover, sometimes the IP associated with a given URL has to change.

C. The Internet therefore makes use of a domain name system, which provides an unambiguous mechanism for mapping names to IP numbers.

1. For example, consider the fully qualified domain name `www.cs.gordon.edu`

a) The rightmost name (.edu) specifies what is known as a top-level domain - the domain of educational institutions.

b) gordon is a particular institution within that domain.

c) cs is a subdomain of the gordon domain (The CS program at Gordon)

d) www is the name of a particular host belonging to the CS program, which has the IP address 10.100.150.17.

2. There are several types of top-level domains, of which two are most commonly seen.

PROJECT: Top-level of domain name space

a) Generic TLDs refer to broad categories (like .edu)

PROJECT: Generic domain labels

b) There is a TLD for each country (e.g. .uk, .ca, .cn ...)

D. The domain name system is supported by DNS servers.

PROJECT: Hierarchy of name servers

1. Note that each level in the hierarchy only needs to record IPs for the next level down

a) The .edu TLD server records the IP for the DNS for gordon.edu

b) Gordon's DNS records the IP for the DNS for cs.gordon.edu

c) The CS DNS records the IP for www.cs.gordon.edu

2. Every host records the IP for the DNS it uses, and each DNS records the IP of the next DNS up.

3. The process of mapping translating a domain name into an IP is called resolution. A host application can request the resolution of a domain name from the local DNS. If the local DNS cannot handle it, the request is forwarded up the hierarchy and then back down another branch of the hierarchy.

   PROJECT: Resolution of a domain name

   DEMO: host www.cs.gordon.edu

   (Note that, to resolve this, my computer contacted Gordon's DNS, which in turn needed to contact the CS DNS).

E. The DNS system also provides support for reverse resolution - i.e. translating an IP back to a domain name.

   DEMO: host 10.100.150.17

   1. Notice that, though www.cs.gordon.edu resolved to this address, the reverse translation resolved to david.cs.gordon.edu. What's going on?

   2. While the IP associated with a given domain name needs to be unique, the reverse does not have to be true. Several different domain names may refer to the same IP - in which case the reverse resolution process will give one of these names.

3. It is our practice in our department to give individual machines the names of biblical figures like david.cs.gordon.edu or joshua.cs.gordon.edu or amos.cs.gordon.edu.

4. However, we also create functional names likes www.cs.gordon.edu or files.cs.gordon.edu or dbms.gordon.edu. The functional names are stored in the DNS as aliases which we can easily change if we decide to move a particular service to a different physical machine.

F. Finally, note that there the domain name system distinguishes between fully qualified domain names and partially qualified domain names.

1. www.cs.gordon.edu is a FQDN, but www.cs is a PQDN

2. A FQDN can be resolved from anywhere on the internet, but a PQDN is meaningful only in a given context - e.g. on the gordon campus, www.cs would have the same meaning as www.cs.gordon.edu - but off campus only the latter name would resolve. (Compare calling a person by their first name in the context of this classroom versus the need to use a full name in other contexts)

V. **DHCP**

A. We now need to consider the question of how an IP number is assigned to a host.

1. For starters, blocks of IP numbers are assigned by IANA to various regional internet registries for the various areas of the world. RIRs, in turn, allocate blocks of IP numbers to ISPs in their part of the world.

2. ISPs, in turn, allocate ISP numbers to organizations or individual hosts. For example, Gordon's ISP has allocated a block of 128 IPs to us.

3. If a block of IP numbers is allocated to a company or an organization, then it, in turn, assigns IP numbers to each host that is part of it.

B. Actually, to function on the network, a host system needs to know several pieces of information.

1. Its own IP address

2. Its subnet mask

3. The IP address of the gateway to send packets to that are destined for a system outside its subnet

4. The IP address of the DNS server to use.

C. One way to handle this is to use static configuration - all of these are specified by a configuration file that is read at operating system startup.

1. But this has several problems

a) It requires custom configuration when the OS is installed - something we don’t want to impose on a non-technical user.

b) If the network is reconfigured, each host on it may also need to be manually updated.

c) If a host is moved to a different network or subnet, the IP needs to be reconfigured.

d) Once an IP number is assigned this way, it belongs to that host whether it was currently on or not.

2. Nonetheless, this approach may be used for servers, since a server needs to have a known IP address that clients can use to contact it. (You will do something like this with the virtual systems you built in lab, because your systems will be used as servers in a later lab.)

D. Most computers get their IPs dynamically, through a mechanism known as DHCP (Domain Host Control Protocol).

1. Each subnet includes a DHCP server.

a) This may be a designated computer that is always running.

b) Or this functionality may be incorporated into the router than serves as the gateway for connecting the subnet to the larger network.

c) Or this may be a server that simply forwards DHCP requests to a DHCP server on another subnet.

2. A DHCP server is a program that has the task of assigning an IP number and providing other information to a host at boostrap time.

a) It is given a block of IP addresses to be assigned to hosts on its subnet

b) It listens on the DHCP server port - port 67. No other host responds to messages on this port.

c) During system bootstrap, the OS sends a broadcast message to the DHCP server port. (It must use broadcast, of course, because it does not know any IPs at this point).

d) The server returns an offer of configuration information (IP, subnet mask, router IP, DNS IP) as a broadcast message on the DHCP client port - port 68. (It must use broadcast because, though it knows the IP it will be assigning to the host, the host does not yet know its own IP. Port 68 is only used for DHCP response.)

e) When the host that requested this IP receives the offer, it records the assigned IP and other information and confirms its acceptance of the offer with the DHCP server.

f) DHCP servers may maintain a table of MAC addresses and IP numbers assigned, so that if a system is restarted, it can be given the same IP number again.

VI. NAT

A. One thing that may have bothered you in the above was the question of how an organization like Gordon deals with two issues:

1. Gordon has been allocated only 128 IPs - but it has a lot more hosts than that!

2. The IPs for computers on campus belong to private blocks like `172.16.0.0/16` which are not intended for use on the public Internet.

B. Both of these issues are tied up with a strategy known as Network Address Translation (or NAT).

1. In brief, in view of the shortage of IPv4 addresses, an organization like Gordon is typically assigned a much smaller number of IPs than it actually needs.

2. Therefore, Gordon's DHCP server assigns addresses to computers on campus in a private block such as `172.16.0.0/16`. Since this block contains 2^{16} = 65536 addresses, there are plenty to go around.

3. When a computer on campus accesses an off-campus site, a NAT server located conceptually between our campus network and our ISP performs a translation operation in which the IP in the request is replaced by one of a public IP before it is sent out over the Internet.

   This, of course, is straightforward.

C. But how does NAT handle traffic from external nodes back to internal nodes - given that the internal IPs of nodes are not visible outside of the local network?

1. In the case of one or more datagrams sent in reply to a message sent by a local node, this can be handled as follows:

a) The NAT server not only replaces the (internal) "sender" IP by the external IP, but it also replaces the sending port in the internal datagram by a port number that it assigns.

b) The server maintains a table in which it maps the port numbers it assigned to the original internal IP and port number.

c) When one or more reply datagrams come back, the NAT server uses this table to replace the external IP by the internal IP that made the original request and to replace the destination port by the port specified in the original datagram.

d) Then it sends the datagram to the internal network, where it is routed to the correct system.

Example: Suppose, on my computer (with internal IP `10.100.74.246`) I access a web page at at `72.14.204.104` (google.com). The following happens:

(1) The network layer on my computer sends out a web `GET` request packet specifying the particular page wanted, and containing the following protocol information:

(a) Source IP: `10.100.74.246`
(b) Source Port (assigned by the transport layer): `53276`
(c) Destination IP: `72.14.204.104`
(d) Destination Port: `80`

(2) Gordon's NAT server sends a packet out onto the web containing the same request, but with the following protocol information

(a) Source IP: One of Gordon's public IPs
(b) Source Port: A number assigned by NAT
(c) Destination IP: `72.14.204.104`
(d) Destination Port: `80`

(3) It also records in a table that the port it assigned is associated with an original request from port 53276 on 10.100.74.246.

(4) Google's server responds with an OK packet containing the requested page and the following protocol information:

(a) Source IP: 72.14.204.104
(b) Source Port: 80
(c) Destination IP: The same Gordon public IP
(d) Destination Port: The number assigned by NAT

(5) Gordon's NAT server sends a packet onto our internal network containing the page, but with the following protocol information.

(a) Source IP: 72.14.204.104
(b) Source Port: 80
(c) Destination IP: 10.100.74.246
(d) Destination Port: 53276

(6) My computer processes this packet as the response to its original request.

2. Of course, this will not work in the case of communications initiated by an outside system. For this reason, NAT cannot be used by itself for systems which have to be capable of being contacted from outside - instead, such systems need an external IP number as well.

   Example: NSG has given CS the use of two external IPs. One is used for our web server (www.cs.gordon.edu) and one to allow access to the workstation network from off campus. The CS DNS actually supplies a different resolution for certain names, depending on whether the request originates on or off campus.

a) If you request resolution for the name www.cs.gordon.edu off campus, the DNS will supply the external IP `216.236.251.139` But if you request resolution for this name from on campus, the DNS will supply the internal IP `10.100.150.17`.

b) If you request resolution for nabi.cs.gordon.edu (nabi is Hebrew for "prophet") from off campus, the DNS will supply the external IP 216.236.251.138. But if the request originates on-campus, the DNS will supply the internal IP `10.100.150.72` (which happens to be the IP for obadiah, which is the one workstation that is directly accessible from off campus - though from it you can ssh to other on-campus machines, of course.)

c) An off-campus request originating from one of these machines will use NAT as for any other on-campus system; but off-campus hosts can also originate contact with either of these machines using the external IP, which the NAT server will then translate to the internal IP. (Thus, these systems only know their internal IP - the external IP is translated by the NAT server when one of them is contacted.)

VII.Transition to IPv6

A. The current set of Internet protocols is version 4 - commonly referred to as IPv4. A new version known as IPv6 was developed in 1998. (Work was started earlier on a version 5, but was abandoned)

B. The new set of protocols addresses makes quite a few improvements - of which probably the most important is the use of 128 bit IPs in place of 32 bit IPs, which will provide a clean solution to the problem of IP space exhaustion.

C. Of course, transitioning to a new set of protocols is not a trivial operation. Revisions are needed to software protocol stacks on computers around the world, plus numerous pieces of hardware.

D. For a number of years, most new equipment has been capable of supporting both IPv4 and IPv6. The plan is to transition to IPv6 gradually, with both sets of protocols being supported in parallel for a period of time.

Example: When configuring networking on your virtual machine, you will see support for both IPv4 and IPv6 - though we will use just IPv4.

E. In the spring of 2011, major Internet organizations ran an IPv6 test day, in which IPv6 support was turned on in parallel with IPv4 support to see what problems might arise.

PLAY: IPv6 day podcast

F. However, even now - several years later - the transition is far from complete