NetworkFinalReviewSheet

BookChapters:4,5,6, 7.1-7.3

**Key Topics**

**Routing protocols used in the Internet**

Routing Protocol = specifies how routers communicate with each other to distribute information that enables them to select routes between nodes on a computer network

OSPF (ISIS ignored) Page 443 = Open Shortest Path First, Data – Intra-AS data, Link State Routing Protocol that uses flooding of link-state information, Routers Know all b/c of Dijkstra – Operates in Local AS (all subnets). Load-insensitive Algorithm

BGP Page 446 = eBGP and iBGP both deal with Inter Data (AS-to-AS data), e external, I internal, Distance Vector but really Path Vector Routing Protocol, Load-insensitive Algorithm, Pairs of routers exchange routing information over semi-permanent TCP connections using port 179. Each such TCP connection, along with all the BGP messages sent over the connection, is called a BGP connection.

Gateway router connects to other ASes.

Internal router connects among other routers within same AS.

eBGP = a BGP connection that spams two ASes

iBGP = a BGP connection that between two routers in the same AS

BGP Attributes: The AS-PATH attribute contains the list of ASs through which the advertisement has passed. The NEXT-HOP is the IP address of the router interface that begins the AS-PATH.

**Distance Vector and Link State approaches, examples of each**

**Link State(Section 5.2.1)** – OSPF - the network topology and all link costs are known, that is, available as input to the LS algorithm. In practice this is accomplished by having each node broadcast link-state packets to all other nodes in the network, with each link-state packet containing the identities and costs of its attached links. In practice (for example, with the Internet’s OSPF routing protocol, discussed in Section 5.3) this is often accomplished by a link-state broadcast algorithm ­[Perlman 1999]. The result of the nodes’ broadcast is that all nodes have an identical and complete view of the network. Each node can then run the LS algorithm and compute the same set of least-cost paths as every other node. Dijkstra’s is frequently used or Prim’s.

**Distance Vector/Path Vector(Section 5.2.2):** Algorithm is iterative, asynchronous, and distributed. It is distributed in that each node receives some information from one or more of its directly attached neighbors, performs a calculation, and then distributes the results of its calculation back to its neighbors. It is iterative in that this process continues until no more information is exchanged between neighbors. (Interestingly, the algorithm is also self-terminating—there is no signal that the computation should stop; it just stops.) The algorithm is asynchronous in that it does not require all the nodes to operate in lockstep with each other. Bellman-Ford’s equation/algorithm is frequently used. Poison Reversed needed; page 439.

Differences: Page 440.

**IPv4 (addressing, header format, fragmentation (IN MTU SECTION), subnetting, CIDR)**

Addressing

An IP address is technically associated with an interface, rather than with the host or router containing that interface.

Address format: Each IP address is 32 bits long (equivalently, 4 bytes), and there are thus a total of 232 (or approximately 4 billion) possible IP addresses. These addresses are typically written in so-called dotted-decimal notation

**Table

Description automatically generated**

Subnet Mask: indicates that the leftmost 24 bits of the 32-bit quantity define the subnet address. The 223.1.1.0/24 subnet thus consists of the three host interfaces (223.1.1.1, 223.1.1.2, and 223.1.1.3) and one router interface (223.1.1.4). Any additional hosts attached to the 223.1.1.0/24 subnet would be required to have an address of the form 223.1.1.xxx. The IP definition of a subnet is not restricted to Ethernet segments that connect multiple hosts to a router interface.

CIDR: Classless interdomain Routing – CIDR generalizes the notion of subnet addressing. As with subnet addressing, the 32-bit IP address is divided into two parts and again has the dotted-decimal form a.b.c.d/x, where x indicates the number of bits in the first part of the address.

The x most significant bits of an address of the form a.b.c.d/x constitute the network portion of the IP address and are often referred to as the prefix (or network prefix) of the address. An organization is typically assigned a block of contiguous addresses, that is, a range of addresses with a common prefix (see the Principles in Practice feature). In this case, the IP addresses of devices within the organization will share the common prefix. When we cover the Internet’s BGP routing protocol in we’ll see that only these x leading prefix bits are considered by routers outside the organization’s network. That is, when a router outside the organization forwards a datagram whose destination address is inside the organization, only the leading x bits of the address need be considered.

For example, suppose the first 21 bits of the CIDRized address a.b.c.d/21 specify the organization’s network prefix and are common to the IP addresses of all devices in that organization. The remaining 11 bits then identify the specific hosts in the organization. The organization’s internal structure might be such that these 11 rightmost bits are used for subnetting within the organization, as discussed above. For example, a.b.c.d/24 might refer to a specific subnet within the organization.

**Understand what MTU is and what impact it has.**

MTU: The maximum amount of data that a link-layer frame can carry. Because each IP datagram is encapsulated within the link-layer frame for transport from one router to the next router, the MTU of the link-layer protocol places a hard limit on the length of an IP datagram. Having a hard limit on the size of an IP datagram is not much of a problem. What is a problem is that each of the links along the route between sender and destination can use different link-layer protocols, and each of these protocols can have different MTUs.

The solution is to fragment the payload in the IP datagram into two or more smaller IP datagrams, encapsulate each of these smaller IP datagrams in a separate link-layer frame; and send these frames over the outgoing link. Each of these smaller datagrams is referred to as a fragment.

Fragments need to be reassembled before they reach the transport layer at the destination. Indeed, both TCP and UDP are expecting to receive complete, unfragmented segments from the network layer. The designers of IPv4 felt that reassembling datagrams in the routers would introduce significant complication into the protocol and put a damper on router performance. Sticking to the principle of keeping the network core simple, the designers of IPv4 decided to put the job of datagram reassembly in the end systems rather than in network routers.

To allow the destination host to perform these reassembly tasks, the designers of IP (version 4) put identification, flag, and fragmentation offset fields in the IP datagram header. When a datagram is created, the sending host stamps the datagram with an identification number as well as source and destination addresses. Typically, the sending host increments the identification number for each datagram it sends. When a router needs to fragment a datagram, each resulting datagram (that is, fragment) is stamped with the

source address, destination address, and identification number of the original datagram. When the destination receives a series of datagrams from the same sending host, it can examine the identification numbers of the datagrams to determine which of the datagrams are actually fragments of the same larger datagram. Because IP is an unreliable service, one or more of the fragments may never arrive at the destination. For this reason, in order for the destination host to be absolutely sure it has received the last fragment of the original datagram, the last fragment has a flag bit set to 0, whereas all the other fragments have this flag bit set to 1. Also, in order for the destination host to determine whether a fragment is missing (and also to be able to reassemble the fragments in their proper order), the offset field is used to specify where the fragment fits within the original IP datagram.

**Know how to describe common NAT functions and why it is needed.**

Enables private home networks.

The NAT-enabled router does not look like a router to the outside world. Instead, the NAT router behaves to the outside world as a single device with a single IP address.

NAT translation table at the NAT router include port numbers as well as IP addresses in the table entries. This mapping of port to addresses allows the NAT router to forward information to a private home network host. The incoming packet has a destination port which the NAT router uses to forward to the correct private home network host with a simple look up in the NAT translation table. The reverse involves receiving an outgoing packet from one of the private hosts and adjusting the packet’s destination port based on the NAT translation table.

**Link layer (addressing, ARP, unicast vs. broadcast)**

A link-layer address is variously called a LAN address, a physical address, or a MAC address.

For most LANs (including Ethernet and 802.11 wireless LANs), the MAC address is 6 bytes long/48bits, giving 2^48 possible MAC addresses. These 6-byte addresses are typically expressed in hexadecimal notation, with each byte of the address expressed as a pair of hexadecimal numbers. Although MAC addresses were designed to be permanent, it is now possible to change an adapter’s MAC address via software. IEEE manages the MAC address space.

Mac Function:

used “locally” to get frame from one interface to another

physically connected interface (same subnet, in IP-addressing sense)

* 48-bit MAC address (for most LANs) (formerly) burned in NIC ROM, also sometimes software settable

Address Resolution Protocol ARP: Because there are both network-layer addresses (for example, Internet IP addresses) and link-layer addresses (that is, MAC addresses), there is a need to translate between them. For the Internet, this is the job of the Address Resolution Protocol (ARP)

ARP resolves IP addresses only for hosts and router interfaces on the same subnet. If a node in California were to try to use ARP to resolve the IP address for a node in Mississippi, ARP would return with an error.

Each host and router has an ARP table in its memory, which contains mappings of IP addresses to MAC addresses. Figure 6.18 shows what an ARP table in host 222.222.222.220 might look like. The ARP table also contains a time-to-live (TTL) value, which indicates when each mapping will be deleted from the table.

ARP packet has several fields, including the sending and receiving IP and MAC addresses. Both ARP Query (multicast, broadcast to FF-FF-etc Mac Address indicating all in subnet) and ARP Response packets (unicast because we know where the broadcast came from, so we can specify the sender MAC address) have the same format. The purpose of the ARP query packet is to query all the other hosts and routers on the subnet to determine the MAC address corresponding to the IP address that is being resolved.

**Unicast – single sender single receiver**

**Multicast – single sender multiple receiver**

Students often wonder if ARP is a link-layer protocol or a network-layer protocol. As we’ve seen, an ARP packet is encapsulated within a link-layer frame and thus lies architecturally above the link layer. However, an ARP packet has fields containing link-layer addresses and thus is arguably a link-layer protocol, but it also contains network-layer addresses and thus is also arguably a network-layer protocol. In the end, ARP is probably best considered a protocol that straddles the boundary between the link and network layers—not fitting neatly into the simple layered protocol stack we studied in Chapter 1. Such are the complexities of real-world protocols!

**Shared media (channel) sharing. Approaches, tradeoffs. FDM, TDM, CSMA/CD, ALOHA**

FDM: the frequency spectrum of a link is divided up among the connections established on the link. Each connection gets a dedicated frequency band for the duration of the connection. The width of this band is called the bandwidth. This is inspired from telephone networks and radio towers b/c those technologies use FDM first. Hence the misnomer.

TDM: time is divided into frames of fixed duration, and each frame is divided into a fixed number of time slots. The network dedicates one time slot in every frame for established connections across a link.

CDMA: Code division multiple access - CDMA assigns a different code to each

node. Each node then uses its unique code to encode the data bits it sends. If the codes are chosen dcarefully, CDMA networks have the wonderful property that different nodes can transmit simultaneously and yet have their respective receivers correctly receive a sender’s encoded data bits (assuming the receiver knows the sender’s code) in spite of interfering transmissions by other nodes.

CSMA/CD: Carrier Sense Multiple Acess/Collision Detection - listen before transmit; if channel sensed idle transmit entire frame or if channel sensed busy defer transmission.

Collision Detection: collisions detected within short time, colliding transmissions aborted, reducing channel wastage, collision detection easy in wired, difficult with wireless

collisions can still occur with carrier sensing:

• propagation delay means two nodes may not hear each other’s just- started transmission

collision: entire packet transmission time wasted

• distance & propagation delay play role in in determining collision probability

CSMA/CD reduces the amount of time wasted in collisions

* transmission aborted on collision detection

ALOHA: Net-05-Link-part1.pdf Slides 23 – 27.

**Routing (Section 5.2) and forwarding. Differentiate between control plane and data plane.**

Control Plane (Remote controllers) above Data Plane(Routers).

**Data Plane** functions of the network layer—the per-router functions in the network layer that determine how a datagram (that is, a network-layer packet) arriving on one of a router’s input links is forwarded to one of that router’s output links. Forwarding-Related.

**Traditional IP forwarding** where forwarding is based on a datagram’s destination address. **Generalized forwarding** where forwarding and other functions may be performed using values in several different fields in the datagram’s header. SDN - OpenFlow

**Control Plane** functions of the network layer—the network-wide logic that controls how a datagram is routed among routers along an end-to-end path from the source host to

the destination host. Routing-related.

Centralized routing algorithm computes the least-cost path between a source and destination using complete, global knowledge about the network. That is, the algorithm takes the connectivity between all nodes and all link costs as inputs. This then requires that the algorithm somehow obtain this information before performing the calculation. The calculation itself can be run at one site (e.g., a logically centralized controller as in Figure 5.2) or could be replicated in the routing component of each router (e.g., as in Figure 5.1). The key distinguishing feature here, however, is that the algorithm has complete information about connectivity and link costs. Algorithms with global state information are often referred to as link-state (LS) algorithms, since the algorithm must be aware of the cost of each link in the network.

Decentralized routing algorithm the calculation of the least-cost path is carried out in an

iterative, distributed manner by the routers. No node has complete information about the costs of all network links. Instead, each node begins with only the knowledge of the costs of its own directly attached links. Then, through an iterative process of calculation and exchange of information with its neighboring nodes, a node gradually calculates the least-cost path to a destination or set of destinations. The decentralized routing algorithm we’ll study below in Section 5.2.2 is called a distance-vector (DV) algorithm, because each node maintains a vector of estimates of the costs (distances) to all other nodes in the network. Such decentralized algorithms, with interactive message exchange between neighboring routers is perhaps more naturally suited to control planes where the routers interact directly with each other, as in Figure 5.1.

Static routing algorithms routes change very slowly over time, often as a result of human intervention (for example, a human manually editing a link costs).

Dynamic routing algorithms change the routing paths as the network traffic loads or topology change. A dynamic algorithm can be run either periodically or in direct response to topology or link cost changes. While dynamic algorithms are more responsive to network changes, they are also more susceptible to problems such as routing loops and route oscillation.

Load-sensitive algorithm link costs vary dynamically to reflect the current level of congestion in the underlying link. If a high cost is associated with a link that is currently congested, a routing algorithm will tend to choose routes around such a congested link. While early ARPAnet routing

algorithms were load-sensitive [McQuillan 1980], several difficulties were encountered [Huitema 1998]. Today’s Internet routing algorithms (such as RIP, OSPF, and BGP) are Load-insensitive, as a link’s cost does not explicitly reflect its current (or recent past) level of congestion.

Hot Potato Routing: routing is for router 1b to get packets out of its AS as quickly as possible (more specifically, with the least cost possible) without worrying about the cost of the remaining portions of the path outside of its AS to the destination

Route-Selection Algorithm Page 452: A 4 step process to determine best route.

**Know the difference between traditional switching and routing devices. What Internet layers, and what types of packet headers, does each operate on?**

**Why are VLANs useful? Difference between access and trunk ports.**

Trunk port carries between VALNS defined over multiple physical switches together. Access ports are entry port for a VLAN(?) --- An access port is a connection on a switch that transmits data to and from a specific VLAN. Because an access port is only assigned to a single VLAN, it sends and receives frames that aren’t tagged and only have the access VLAN value. This doesn’t cause signal issues because the frames remain within the same VLAN. If it does happen to receive a tagged packet, it will simply avoid it. This is a simpler configuration, but not the most efficient choice if the network is even moderately complex.

Trunk ports = tagging (external to multiple VLANS)

Access ports = no tagging (internal to single VLAN)

**Other virtualization approaches**

Encapsulation something with new label. Routing based on new label.

MPLS layer 2.5 (sits in between 2 and 3) Multi protocol label switching.

STP – Spanning Tree Protocol, solves looping routes but kills benefit of multiple paths. STP not necessarily needed in SDN.

**Data center networks**

**A day in the life of browsing the web.**

Attach laptop to network

Requests web page google.com

1st - DHCP Request – ip, sub netmask, gateway(first hop router), dns

Dchp -> udp -> ip -> eth -> phy

Sent to broadcast address FFFFFFFFFF 48 bitson LAN – ethernet all ones in the host part – IP all ones in host part, except mask

Received by DCHP server decaps and demuxplex up to udp

DCHP ACK is created w/ things we need ip, sub netmask, gateway, dns

Sent back to mac address of client via unicast

Switch in middle learned location of mac address when DCHP request came it. AKA Switch learning

Get ip address of google.com requires DNS look up

Comms with DNS server.

Knowing IP address means we need to get MAC address

Host creates query -UDP – IP – ETH

ARP query to resolve first hop router mac address – broadcast, received by router, which replies with ARP reply giving MAC address of router interface

(we know to use arp b/c we use the subnet mask with DNS server and find out it is not in the same subnet. In case dns in different subnet we need to talk with router. Thus ARP request, ARP reply. Know host knows MAC address of first hop router so send frame containing DNS query.)

Goes out ISP DNS server – based on forwarding information / longest string match to fnid prefix ip destination – via BGP AS to comcast as – comcast edge router sents to dns – demutiplex / deencapsulation send back to host

1st Now with DNS info we can send http request get/http 1.1 http header http->tcp->IP->->eth->phy

SYN flag is set in first sent segment (new TCP connection) starts 3wayhandshake (step 1)

2nd Web server gets syn segment frame to datagame to segment decapsulations

Web server creates connected web socket and returns to host SYNACK syn and ack bit set back to host

3rd Now send HTTP request to webserver - includes data, sets ack bit to acknowledge the initial sequences numbers (selected randomly)

Reply goes back which contains webpage -> webpage is displayed finally….

Tcp – segment

Ip – datagram

Eth – frame

All – causally packet

OSPF/ISIS = internal AS learning – Link State

BGP – external AS learning – Distance Vector but really Path Vector

**Wireless and WiFi 7.1 7.3**