1.	**The Network Layer**	Transport segment from sending to receiving host, encapsulates segments into datagrams
2.	**Forwarding**	Move packets fro routers input to appropriate router output. Data plane
3.	**Routing**	Determine route taken by packets from source to destination. Control plane
4.	**Data Plane**	- Local, per-router function - Determine how datagram arriving in inpur port is forwarded to output port - Forwarding function
5.	**Control Plane**	- Network-wide logic - Determine how datagram is routed among routers along end-to-end path from source to destination - Two approaches - Traditional routing; routers - Software-defined routing; remote servers
6.	**Traditional Routing**	Individual routing algorithm components in each and every router interact in the control plane
7.	**SDN Routing**	A distinct (remote) controller interacts with local control agents
8.	**Forwarding Table**	Table that maps destination address to a routers outbound links
9.	**Longest Prefix Matching**	Finds the longest matching entry in the table
10.	**Switching Fabric**	Connects the routers input ports to its output ports. Can be:

		- Memory - Bus - Crossbar
11.	**Input Port Queueing**	- Can occur when fabric is slower than input ports combined. HOL-blocking can occur here - Queuing delay and loss due to input port buffer overflow possible
12.	**Output Port Queueing**	- Buffering can occur when arrival rate via switching fabric exceeds output line speed - Queuing and loss due to output port buffer overflow possible
13.	**Scheduling Mechanisms**	- Choosing next packet to send on the link
14.	**FIFO Scheduling**	Packets are sent in order of arrival to queue
15.	**Priority Scheduling**	Packet with highest priority in the queue is sent Issue: Starvation
16.	**Round-Robin Scheduling**	Packets are sent in a cyclical fashion, one from each class. Solves starvation, however unfair to queues with many more packets
17.	**Weighted Fair Queuing Scheduling**	Generalized round-robin, but each class gets weighted amount according to volume of packets
18.	**IP Datagram Fragmentation**	Datagram is fragmented into smaller datagrams and then reassembled

		at the destination host. This accomodates MTU of link level. Utilized fragflag and offset fields in header to reassemble datagram.
19.	**MTU**	(Maximum Transfer Unit) The largest data unit an ethernet network will accept for transmission.
20.	**IP Addressing**	32-bit identifier for host/router interface, consisting of network identifier and host identifier. Interface: connection between host/router physical link
21.	**Class A IP Address**	- Few networks, many hosts - 7-bits for network, 24-bits for host - Begins with 0, decimal 1-126
22.	**Class B IP Address**	- Medium networks, medium hosts - 14-bits for network, 16-bits for host - Begins with 10, decimal 128-191
23.	**Class C IP Address**	- Many networks, few hosts - 21-bits for network, 8-bits for host - Begins with 110, decimal 192-223
24.	**Subnet**	A logical subset of a larger network, created by an administrator to improve network performance or to provide security.
25.	**Subnet Mask**	Tells how many bits of host ID are subnet ID and how many are the host ID
26.	**Extended Network ID**	Network ID and subnet ID (not host ID)

27. **Default Class Masks**

Class A:
11111111.00000000.00000000.0000000

Class B:
11111111.11111111.00000000.0000000

Class C:
11111111.11111111.11111111.0000000

28. **Example Subnet Problem**

ip: 192.10.17.22
mask: 255.255.255.240

240 in binary => 1111 0000
22 in binary => 0001 0110
this means the first 4 bits of host ID are subnet ID,

so,

subnet ID: 0001 = 1
host #: 0110 = 6

and extended network # = 192.10.17.1

29. **Example Subnet Problem**

ip: 130.157.224.240
mask: 255.255.255.240

240 in binary => 1111 0000
240 in binary => 1111 0000
this means the first 4 bits of host ID are subnet ID,

so,

subnet ID: 1111 = 1
host #: 0000 = 0

		and extended network # = 130.157.224.15
30.	**Exercise: Given a network address of 192.168.100.0 and subnet mask 255.255.255.192: How many subnets are created, and how many hosts per subnet?**	192 = 1100 0000 (Class C) Since 2 one-bits, 4 subnets created Since 6 zero-bits, 2^6 hosts created
31.	**Exercise: Given a company with six individual departments and each department having ten computers or networked devices, what mask could be applied to the company network to provide the subnetting necessary to divide up the network equally?**	Six departments -> Need six subnets Ten computers -> Need minimum 10 hosts per subnet 1110 0000 -> Works but not equally divided 1111 0000 -> Works and equally divided
32.	**CIDR**	Classless InterDomain Routing Address format: a.b.c.d/x where x is number of bits in subnet portion of address. Ex: 200.23.16.0/23 11001000 00010111 0001000 (end subnet portion, start host portion) 0 00000000 23 bits for the subnet, so the host will be 9 bits. 2^9 = 510 possible hosts
33.	**IP Broadcast Address**	255.255.255.255. All hosts on the same subnet get datagrams sent to this address
34.	**DHCP**	Dynamic Host Configuration Protocol: dynamically get address from a server. Can also return address of first-hop router for the client, address

		of DNS server, and network mask 4 steps: DHCP discover DHCP offer DHCP request DHCP ack (DHCP is application layer protocol)
35.	**NAT**	Network address translation. Outgoing datagrams: Replace source socket with NAT socket Remember in NAT translation table every source socket to NAT socket translation pair Incoming datagrams: Replace NAT socket with source socket
36.	**IPv6**	Addresses limited IPv4 addresses. Reduced header size. No fragmentation.
37.	**Tunneling**	IPv6 routers can wrap IPv6 datagrams in IPv4 header for backward compatibility