1. What is *forwarding* ? Do we need an address for forwarding ?

- Given a packet at a node, ***finding which output port it needs to go to is called "forwarding***" **–** it is a per-node function, whereas routing may encompass several nodes.
- The ***forwarding function is performed by every node in the network*** including hosts, repeaters, bridges and routers.
- Forwarding **is *trivial in the case of a single-port node*** (or in a dual port node, where the destination is on the port other than the input port) – in this case you don't need even addresses.
- The earliest bridges used to forward packets from collision domain to collision domain, <u>without filtering</u>. This was ***basic layer-2 forwarding*** or also called "***flooding".***
- For ***flooding, you do not require addresses*** (though LANs use addresses for other reasons). Flooding is useful in order to reach routers if the routing tables and addresses cannot be relied upon (eg: in distributing routing info itself!)
- ***Forwarding is a data-plane activity*** and is performed on every packet received.
- The series of functions performed on every packet is also called the "critical path" or "***critical forwarding path***" in vendor lingo.

2. What is *filtering* ? How are techniques like TTL (time-to-live/hopcount), sequence numbers/timestamps, and forwarding tables related to the goal of filtering ?

- When a packet is received by a node, filtering is the task of
  - **<u>a)</u>** determine whether to forward the packet at all, and
  - **<u>b)</u>** which port(s) to forward the packet to.
- Filtering makes the network operation more efficient by ***reducing the number of "output ports" to which the packet needs to be sent to.*** For example:
  - Unicast packets need to go to only one output port, and that output port should the next step in the desired path to the destination. Multicast packets need to go to a (sub)set of ports. ***<u>The forwarding table</u>*** encodes this subset of ports and avoids the need to carry such information in the packet itself.
    - The set of forwarding tables together should satisfy the global condition that desired paths can be found by the concatenation of successive forwarding decisions.
  - Moreover, sending a packet to multiple ports results in multiple copies of the packet. Unless some ***global (or per-source/per-packet) filtering condition*** is enforced, infinite number of copies can be created.
    - ***TTL (time-to-live/hop count)*** ensures that every packet is flushed from the network after traversing N hops.
    - Sequence numbers or timestamps ensure ***that duplicates or spurious copies*** of packets are filtered out. In the case of some control information, ***older information can be considered duplicates even though the data content may not be the same.***
    - Of course, sequence numbers have ***other functions***: detection of <u>missing</u> packets, determining <u>packet ordering</u>, and to enable window-based <u>flow control</u>. Timestamps also have a <u>timing</u> function which can be used to estimate RTTs, and <u>ordering</u> of packets.
  - In other words, **<u>forwarding tables, TTL fields, sequence numbers fundamentally enable filtering.</u>**
  - Note that some aspects of filtering (duplicate removal) may also be considered as "reliability" functions.

3. Do we need to have *forwarding tables* for forwarding ?

- Recall that the primary ***purpose of forwarding table is to enable "<u>filtering</u>***" i.e. **<u>a)</u>** determine whether to forward at all, and **<u>b)</u>** which port(s) to forward the packet to.
  - Forwarding tables are not necessary for forwarding. As explained earlier, ***simple flooding*** can be used which does not require forwarding tables or addresses. But its filtering capabilities are the poorest.

- Another way to avoid forwarding table complexity is ***to code the path, i.e. the sequence of output port numbers from source to destination for unicast packets*** -- so that any intermediate node can directly infer through computation alone (no table lookup) what the output port number is.
    - As you may have realized***, source routing by putting the entire path in the header is very inefficient***. Also, in connectionless networks like IP, the path from source to destination (i.e. the sequence of intermediate nodes) may change from packet to packet – routes become hard to predict.
    - This does not mean that there is no useful information coded in the addresses or the header to help forwarding. In IP addresses, there is some information coded however -- the network address which is used in its forwarding decision. TTL is another useful piece of info in the header.
- The forwarding table is therefore an efficient way to maintain forwarding relationships which cannot be easily coded into addresses.
    - It is efficient because of the property that the concatenation of forwarding decisions is a ***loop-free, reasonably short path*** to the destination. This property is ensured by ***a control-plane protocol (routing or learning).***
    - Other path properties like QoS, policy constraints etc may also be attempted by complex routing algorithms.
    - Bridging uses a <u>learning</u> algorithm (based upon source address) to "figure out" how to filter.
- Note that the forwarding table is a ***a level of indirection*** at the intermediate node.
    - Recall that any multiplexing + indirection gives you a virtualization. In this case, we have the multiplexing (sharing) of the intermediate node and a level of indirection (the forwarding table) to create a ***virtual link abstraction across a non-meshed network.***

4. What is ***routing***? How is it different from ***forwarding*** and ***bridging*** ?

- Routing is the function of ***finding a path to the destination.***
    - Operationally, routing is the ***component <u>that sets up the forwarding tables</u> at all intermediate nodes.***
    - To setup this "***<u>local</u>***" forwarding table, ***it has to use some summarized form of "<u>global</u>" knowledge.*** The summarization may be done using some ***global properties*** (eg: a tree is loop-free; concatenation of local properties (eg: distance-vector method)). Else, the concatenation of local forwarding decisions is meaningless and may or may not lead to the destination
    - Routing is meaningless without the concept of intermediate forwarding nodes and the need for efficient filtering functionality (forwarding tables).
    - Routing is what makes the network operation efficient by sending packets <u>*only*</u> where they need to go.
        - A route is dependent upon the location of both the source and the destination.
    - Routing improves user performance by increasing the chances of their packets
        - a) reaching the destination (eg: random routing would have low probability of delivery !)
        - b) not waste time going through longer paths than necessary (including loop-avoidance/mitigation)
        - c) Sends packets out only in one direction (for unicast) and never floods data packets. (note that control packets may be flooded).
- Routing is a ***control-plane function***. I.e. it is <u>usually NOT performed on a packet-by-packet basis</u>.
    - Routing and forwarding are ***complementary***, because both are needed to allow packets to efficiently reach the destination.
- The core challenge in routing is the ***tradeoff between consistency, completeness and scalability.***
- **<u>Bridging vs Routing:</u>** In a LAN, the ***routing-like control function in a LAN is done by bridges*** which set up spanning trees and learn on which interface destination addresses reside.
    - Bridges "forward" and "filter" packets in the data-plane, and perform the above routing-like functions in the control plane.
    - The problem with bridges is that if they do not know the address of a destination, they flood the packet into all outgoing ports except the one it came on.
    - Flat addressing means that bridges cannot aggregate addresses to save entries in the table.

    - It also means that bridges cannot differentiate between directly connected hosts and indirectly connected hosts.
    - Routing, aided by address aggregation solves that problem
-

5. What is the complexity of routing in general ? How can it be controlled?

- Routing is ***not complex if the demands on filtering are not high***. For example, one could build a non-scalable network by simply flooding.
- For regular routing which is chartered to find paths (and that too minimum distance paths) to the destination, the complexity of routing ***depends upon the size of the network measured in terms of number of (virtual) nodes***.
    - Specifically, the <u>size of the routing table</u> is proportional to the number of nodes (or virtual nodes) seen by the node in the network, i.e. <u>O(N).</u>
        - This basic complexity is ***independent of the type of algorithm*** used (link-state, distance-vector).
    - The <u>scope of control traffic</u> (how far control traffic from one router travels) is also proportional upon the number of nodes seen by the router, i.e. O(N).
    - The information exchanged between nodes to <u>***maintain a single entry*** is proportional to the number of edges in the network i.e. the square of the number of nodes in the network, i.e. O (N^2).</u>
    - <u>The total control traffic in the network is in the worst case proportional to the product of the above components, i.e., upto O(N^4) !!</u>
- The complexity can be fundamentally reduced by ***address aggregation*** - i.e. making entire networks look like a single virtual node as seen by routers, thus reducing N , the number of "virtual nodes" in the "network" <u>seen</u> by the routing algorithm.
    - Address aggregation means that the "destination" field in the routing table can have full destination IP addresses or just prefixes of different lengths. It is this flexibility which allows considerable scalability in routing compared to bridging.
    - The cost of address aggregation is to introduce sub-optimality in routing. Optimal routing, simply stated, requires individual routes for individual pairs of hosts; whereas aggregation implies that the network routes only to subnets, not hosts. But this suboptimality is trivial compared to gains in efficiency of control traffic
    - Another way to control complexity is the use of ***default routes*** at the periphery of the Internet. This is essentially a form of "passing the buck." The buck stops at border or core routers where the routers maintain full routing tables.

6. How has the evolution of inter-domain routing related to the evolution of the Internet backbone(s) ?

- Initially, there was only one backbone network, the ***ARPANET***.
- ***Early years:***
    - The Internet as we knew it really began in 1985 when NSF funded 5 supercomputer centers to be linked up via a 56kbps network using TCP/IP. It also funded the setup of regional networks and campus networks which interconnected to the ARPANET backbone leading to a 3-tier network. The NSFNET was also a new backbone.
    - In 1987, traffic increased and NSF contracted to Merit (<u>www.merit.edu/internet</u>) to upgrade the 56kbps to T-1 lines connecting about 170 LANs.
    - In late 1990, Merit, IBM and MCI spun off a quasi-independent organization called Advanced Network Services (ANS) which operated the upgraded backbone at T-3 (45 Mbps) speeds connecting 3500 networks.
        - During this time, the campus and regional networks operated RIP and the interconnection was achieved through GGP.
        - Later the interconnection protocol was changed to EGP, and the scaling needs of campus and regional networks drove the development of OSPF.
- ***Commercialization:***

- In 1991, PSInet and Uunet argued that the backbone operations and regional networks should be commercialized. They founded the Commercial Internet Exchange (CIX). But most traffic moved over NSFnet.
- Separately Metropolitan Fiber Systems (MFS – now part of MCIWorldcom) began to create Metropolitan area Ethernets (MAEs – later called "exchanges") – fiber optic rings that served businesses in major urban areas.
- In early 1993, NSF announced that it was getting out of the backbone business and it would contract with vendors to create a series of Network Access Points (NAPs) where private commericial backbone providers could connect to exchange traffic. The MAE's and NAPs became NAPs. In April 1995, NSFNET was shut down.
- Campus networks connect to ***POPs (points of presence)*** or NAPs of regional providers. This architecture posed several problems.
    - The use of EGP as a backbone routing protocol was insufficient to account for the richer architectural variety (EGP would only allow a tree-type interconnection) and control of policies. This led to the development and adoption of BGP-4. Actually the push towards BGP had started earlier with the need to have multiple peering points between NSFNET and ARPANET.
    - The management of NAPs and the role of CIX became important. To allow the growth of the Internet, the policy at NAPs, ***CIX was to have no settlements,*** i.e., small ISPs would peer with large ISPs without paying extra money.
    - ***Private peering*** also took off to allow higher quality wide area inter-connectivity.
    - ***Route Arbiter (RA):*** At the NAP, multiple networks/ASs join => multiple border peers need to be maintained. Maintaining peering relationships between N routers leads to huge amounts of control traffic. This is a problem.
    - To solve this, NAP routers peer with one node, called the "Route Arbiter" (RA) or the router server (RS). The RA maintains policies and routing databases, and need not transfer traffic. I.e. the RA provides a simple solution to the control plane scaling problem only.

6a. Give me an introduction to bandwidth services offered by ISPs and components used in that process.

- ISPs offer services like leased lines (T-1, T-3, OC-3 etc), frame relay (see next), and dial up services (modem connectivity, ISDN, ADSL etc).
- Frame relay is one of the most economical ways for corporations to hook up to the Internet. Purchasing sufficient point-to-point leased line connections: prohibitively expensive (eg: T-3 line coast-to-coast is millions of dollars/year). With frame relay, corporations can buy enough bandwidth to meet their existing needs and to easily expand as traffic requirements increase.
- ISPs are increasingly getting into Web hosting/Data centers etc. But we will ignore that for the time being.
- ISP Backbone selection criteria:
    - ***Physical connections:*** The ISP should be able to show a decent map of healthy physical topology which can provide consistent, adequate bandwidth for the whole traffic trajectory. But note that existence of OC3/12 does not guarantee overall high-speed access since your traffic could flow over some backdoor T1 or frame-relay clouds slowing overall experience… *End-to-end performance is key*.
    - *Potential ISP bottlenecks:* Oversubscription of links. The typical max over-subscription is 4:1 (especially PoP-NAP backhaul links). ISP like to tell stories how their backbone is undersubscribed (or over-provisioned). But the key is the access links and peering capabilities/contracts they have with their providers…
    - *ISP Internet access redundancy:* ISP's connection to NAP or POP-NAP connections may go down. A redundant network with switch-over capabilities to handle outages at all layers is important. Large ISPs advertise SONET rings which handle failures at the physical layer (fiber cut, bad interface card). IP routing or ATM technology provides re-routing services, but ISPs are augmenting them with auto-reroute capabilities in MPLS.
    - *Hops, NAPs to destination:* More hops => more potential for traffic to be delayed, dropped, garbled, mis-routed etc… How many NAP cross-overs is also important because these are points of congestion … Major ISPs claim less than 5 hops to destination…

    - *Traffic Exchange Agreements/Peering:* Important that ISP be part of all of this.

Demarcation point: Where responsibility gets split up.

- *Customer Premises Equipment (CPE):* is where a router, CSU/DSU, cabling and monitoring equipment is placed. If equipment not pre-approved by ISP, the customer may be responsible for management of the equipment
- *Router Collocation:* Placing ISP routers at customer premises (since real-estate is scarce). In this case, the ISP administers the router remotely.
- *If agreements state that customer manages equipment: they participate in routing policies hands-on…*
-

7. Explain how Classless Inter-Domain Routing (CIDR) profoundly affects the scalability of global Internet routing.

- CIDR was developed to allow the "subnetting" (flexible address space division between network address and host address) idea to be extended to the network part of the IP address too (called "**supernetting**"). Recall that subnetting was a method to make address allocation more efficient.
    - This effectively would make the addressing "*classless*" for the purposes of routing. Since *inter-domain routing* protocols are the ones that provide routing to the network-part of the IP address, the protocol is called CIDR.
    - The extension of subnetting to the network part ***also affects address aggregation*** especially because long class C prefixes need no longer be maintained in the core for every class C network
- ***Implications of being "classless":***
    - In CIDR, we ***prefer the use of the term "prefix" over "network***" because it's more clear that no Class is being implied.
    - It also uses the ***notation: 128.221.13.20/20 emphasizing that 20-bits are the network address***. This is also called the <prefix.length> notation.
    - The prefix may well be 2 bits, i.e. the prefix can be shorter than the natural mask (i.e. what was formerly a network "class"). This is why the term "***supernetting".***
        - Eg: 198.213.0.0/16 has 16-bit mask shorter than natural 24-bit (class C) mask. The 16-bit prefix is invalid in the class C (natural mask) sense because the class C natural mask has 24-bits.
        - Eg: 198.24.0.0/20 and 198.24.56.0/21 {"***more specific***"} can be aggregated as 198.24.0.0/18 {"***less specific***"}. Note that between each pair of dots, there are 8 bits. /20 means four bits into the third number.
        - The dotted-decimal notation is confusing when we try to interpret the CIDR masks with the prefixes ending on non-octet boundaries.
    - The ***masks can be on arbitrary bit boundaries*** and don't have to be on byte boundaries (like the earlier classful boundaries). Note that this still means that networks have to have address spaces that are a power of 2.
        - Appletalk allows the prefix to end on any number, not a bit value. The CIDR approach is preferred because it facilitates bit-anding in the forwarding path, and also allowing efficient longest-prefix match algos.
    - ***Longest-prefix matching***: since a routing table can now contain several addresses where part of the prefix would match, the forwarding algorithm has to be modified to match the destination IP address with longest prefix, and not just the first prefix. In other words, it has to *match the most-specific prefix.*
- ***Implications on inter-domain routing, address aggregation and address allocation:***
    - CIDR enables powerful forms of aggregation and thus helping to reduce the size of routing tables in Internet cores.
        - Earlier, every class A, class B and class C network needed to be advertised. The number of class B networks and class C networks is huge (64K class B and 2^24 class C networks !!). The routing tables can be as huge as that !
        - The key idea is that the <prefix.length> notation allows all the ***more specific routes handled by an ISP to be summarized into one aggregate***, provided that ***they all refer to non-overlapping subsets of a larger address block.***

        - This means that a provider could get a large address block, i.e. a small prefix (eg: 10 bit prefix rather than a 16-bit class B or 24-bit class C prefix), and allocate smaller blocks to its customers.
        - But ***each of these customer sub-blocks (longer prefixes) need <u>not</u> be advertised***. Only the single small prefix needs to be advertized. In real terms, the provider would advertize one 10-bit prefix in this case instead of advertising 2^14 small 24-bit prefixes or 2^6 16-bit prefixes – ***several orders of magnitude improvement !!***
        - ***Customers however don't like provider based addressing because when they*** move from one ISP to another they need to renumber to remain in the other ISP's CIDR block. This can be a complex and costly administrative task if DHCP or NAT is not available.
        - Renumbering can be done with DHCP automatically. NAT allows avoidance of the renumbering problem by use of private address space.
    - This above allocation procedure of the provider getting large address blocks (small prefixes) from IANA and allocating sub-blocks is called ***provider-based address allocation***.
        - In the pre-CIDR era, sites would go to a centralized registry (IANA) to get an address prefix which:
            - A) does not take into account where that site connects to the Internet), and
            - B) allocates only classful prefixes
        - The crux of CIDR is that the ***Internet's generally hierarchical topology and administration is now being reflected in the addressing.***
            - To appreciate this fact, observe that in IEEE 802 addresses, the OUI field also implied an administrative hierarchy, but not a topological hierarchy (the IEEE address space is "flat").
                - The ***CIDR-based IP addresses are now overloaded: they have both an administrative significance and a topological significance, and both are expected to be hierarchical !!***

8. What are the central problems of inter-domain routing ?

- <u>*Full routing tables*</u> (reachability)
    - IGPs don't scale
    - Many default (last-resort) routes are maintained. This is equivalent to passing the buck of routing responsibility. The buck stops at inter-domain routers.
- <u>*Policy control:*</u> what types of policies, what mechanisms in protocols, how to ensure consistency, open problems
    - IGPs don't have technical hooks for managing interconnections between organizations that are administratively and politically independent of each other.
- <u>*Address aggregation mechanisms (see CIDR section}*</u>
- Support for <u>*redundancy*</u> in customer-ISP connections and managing its conflict w/ aggregation
    - More specific addresses => more traffic
- <u>*Load balancing*</u> support for customers
    - Internet routing is <u>*destination-based*</u> => one route-per destination. Hard to have separate routes to the same destination without playing some tricks like traffic splitting at the source, or between multiple inter-router links. Almost impossible to do it at the core because of longest-prefix match rule.
    - <u>*Longest-prefix matching*</u> => traffic tends to follow the path corresponding to the more-specific address advertisement. Also once the longest-prefix is matched, no traffic is sent towards the shorter prefix matches. This leads to hot spots on the longest-prefix paths, which are usually advertised because the ISP cannot aggregate them and they are backdoors, or backup links…
    - Moreover, *shorter hop paths* (AS-hops or router-hops) or *lower cost paths* are also open to dumping of traffic.
- Support for <u>*customer mobility*</u> (moving between ISPs)
    - Less flexibility
    - Proxy aggregation: aggregating someone else's address space is not allowed because it is tricky and can lead to black holes.
    - {More discussion below}

9. What are the tradeoffs which arise from the need to provide for a) address aggregation, b) Route stability c) redundant WAN connectivity d) Load balancing e) Need for customers to quickly switch between ISPs.

- It is well understood that route aggregation reduces the number of virtual nodes seen by interior routers, and thus dramatically reducing the control traffic and route table management complexity.

- **Address aggregation** is best *enabled through hierarchical address allocation, suppression of more-specific prefixes by lower tier ISPs (aggregation), and CIDR which provides the address structure and mechanism* to do it.
  - ***Hierarchical address allocation:*** IANA allocates address to ISPs, lower tier ISPs get address sub-blocks from higher tier ISPs, and enterprise customers get address sub-blocks from their ISPs.
  - ***Aggregation :*** ISPs aggregate address prefixes and advertise only smaller prefixes to their higher ISPs or to the core. This aggregation is done at the border routers of the ISP.
  - ***CIDR:*** CIDR allows prefixes to be of any length (esp. as small as necessary for aggregation). The class boundaries which enforced these restrictions have been removed.
  - ***Recommendations:***
    - Aggregate from the leaves (customer premises) as much as possible. In multi-homed situations, aggregate without creating ambiguity. Eg: You can only aggregate routes that you administer. The cost of redundancy and customer mobility is having lesser aggregation.
    - Moving from one provider to another => renumbering plan must be put in place.
    - Else NAT (network-address-translation) boxes should be put in place to translate the new addresses to the old. NAT has to handle several corner cases because IP addresses are used in transport protocols (eg: definition of a socket) and in application protocols (eg: FTP codes IP address as a character string !). NAT also breaks with security => NAT must be integrated with the box that does firewalls (which is why Checkpoint is so valuable).
    - DHCP can be used to multiplex a smaller public address space with a larger private address space. But with DHCP destinations which did not have “names” become harder to access directly (eg: home computers connected via an ISP which does DHCP).
    - For private (not-public) connectivity, private IP address spaces [10.0.0.0 (Class A space), 172.16.0.0 – 172.31.255.255 (16 class B spaces) and 192.168.0.0-192.168.255.255 (256 Class C spaces) are available – no permission from IANA needed].
- **Redundancy/Load balancing:** Customers would like to keep a backup link to another ISP in case one ISP goes down. But since interfaces can have only one IP address, it has to choose the numbering from one of the ISPs (say A.B.C.D/20) TO maintain backup connectivity, this IP prefix is advertized to the other ISP too !
  - However since A.B.C.D/20 is not allocated from the second ISP’s address space, it cannot aggregate it. This means that the first ISP may aggregate the prefix to a smaller value (say 10 –bits), but the second ISP will anyway announce the 20-bit prefix to the core. If the two ISPs share the same core, ultimately the core will contain both the 10-bit prefix AND the 20-bit prefix, defeating the goals of aggregation !
  - Moreover since CIDR requires forwarding based upon **longest prefix match**, the packets will be forwarded to the second ISP (which was intended to be the backup !!).
  - An alternate way to implement the backup system is for the customer to not advertize the 20-bit prefix to the second ISP until the first ISP goes down. This way the backup can be enabled as soon as the address propagates through the second ISP to the core. The core does not see the 20-bit prefix unless it is temporarily needed as a backup service.
  - The problem with this strategy is that the backup link cannot also be used for load-balancing. It becomes a pure backup. If the 20-bit prefix is advertized only sometimes, then a little bit of load-balancing can be done, but it introduces a lot of churn in the routing protocols leading all the way upto the core. This affects routing stability (see next major bullet)…
  - It is more flexible to have redundant links *(multihoming) to a same ISP* (at different PoPs) for the purposes of load balancing, backup links, insertion of granular routing information (eg: the prefix of two sub-domains which are not more-specific when compared to each other) without impacting the global routing system. Therefore, it has become popular to have multi-homing to a single ISP.

Of source it does not help if the ISP itself fails at multiple points i.e. the entire network comes down for the ISP.

- **Route stability:**
  - Route stability refers to the fact that routes between points A and B don't change often. Stability is important for building QoS and premium services over the system. TCP also gets confused by route flaps.
  - With large routing tables, the bandwidth and processing costs at routers increases dramatically – leading routers to take vacations. Now the other routers spuriously timeout in the case leading to route change, followed by immediate changeover. This is called ***route-flaps.***
  - Another benefit of route aggregation is that route flaps are limited in number, frequency and scope, which saves resources and makes the global Internet routing system more stable.