CS6250 Computer Networks Study Guide with students' answers

Lesson 1: Introduction, History, and Internet Architecture

What are advantages and disadvantages of a layered architecture?

Advantages:

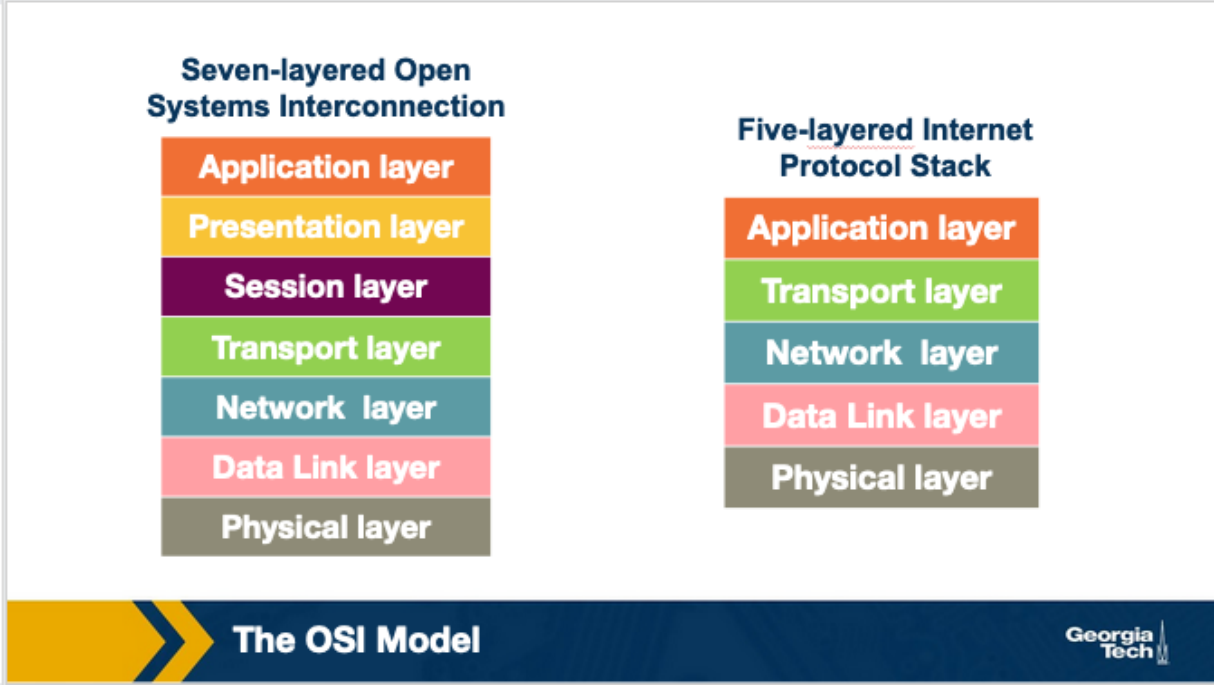
- Modularity
- Scalability
- Flexibility to add/delete components (allows for cost effective implementations)

Disadvantages:

- Violation of the goal of layer separation
- Overhead costs/performance
- Duplication of functionality

What are the differences and similarities of the OSI model and five-layer Internet model?

The application + presentation + session layer of OSI are the same as the application layer in the Internet model. Everything else is the same.



What are sockets?

A socket is an endpoint for sending or receiving communication over a network. Sockets are the interface between the application and transport layers.

Describe each layer of the OSI model.

"Please Do Not Throw Salami Pizza Away"

1. Application
    a. Protocols are specific to each use case (SMTP email, HTTP web). At the application layer, we refer to the packet of information as a **message**.
2. Presentation
    a. The presentation layer plays the intermediate role of formatting the information that it receives from the layer below and delivering it to the application layer. For example, some functionalities of this layer are formatting a video stream or translating integers from big endian to little endian format
3. Session
    a. The session layer is responsible for the mechanism that manages the different transport streams that belong to the same session between end-user application processes. For example, in the case of teleconference application, it is responsible to tie together the audio stream and the video stream
4. Transport
    a. The transport layer is responsible for the end-to-end communication between end hosts. In this layer, there are two transport protocols, namely TCP and UDP. The services that TCP offers include: a connection-oriented service to the applications that are running on the layer above, guaranteed delivery of the application-layer messages, **flow control** which in a nutshell matches the sender's and receiver's speed, and a **congestion-control** mechanism, so that the sender slows its transmission rate when it perceives the network to be congested. On the other hand, the UDP protocol provides a connectionless best-effort service to the applications that are running in the layer above, without reliability, flow or congestion control. At the transport layer, we refer to the packet of information as a **segment**
5. Network

    a. In this layer, we refer to the packet of information as a **datagram**. The network layer is responsible for moving datagrams from one Internet host to another. A source Internet host sends the segment along with the destination address, from the transport layer to the network layer. The network layer is responsible to deliver the datagram to the transport layer in the destination host. The protocols in the network layer are: 1) The IP Protocol, which we often refer to as “the glue” that binds the Internet together. All Internet hosts and devices that have a network layer must run the IP protocol. The IP protocol defines a) the fields in the datagram, and b) how the source/destination hosts and the intermediate routers use these fields, so the datagrams that a source Internet host sends reach their destination. 2) The routing protocols that determine the routes that the datagrams can take between sources and destinations.
6. Data Link - moves frames from node to node (host/router/L2 switch). It provides link to link reliability
7. Physical
    a. The physical layer facilitates the interaction with the actual hardware and is responsible for transferring bits within a frame between two nodes that are connected through a physical link. The protocols in this layer again depend on the link and on the actual transmission medium of the link. One of the main protocols in the data link layer, Ethernet, has different physical layer protocols for twisted-pair copper wire, coaxial cable, and single-mode fiber optics.

Provide examples of popular protocols at each layer of the five-layered Internet model.

Application - HTTP/SMTP/SMB/FTP/DNS
Transport - TCP/UDP
Network - IP
Data Link - Ethernet/WIFI/PPP
Physical - Link dependent. For example, twisted-pair copper wire or coaxial for Ethernet. Single mode fiber optics. DSL. See Chapter 1 in Kurose-Ross book.

What is encapsulation, and how is it used in a layered model?
Each layer adds it own headers to the message when sending a message that will be used by the receiver

What is the end-to-end(e2e) principle?
The principle states that since certain functionality (error detection for example) must be implemented on an end-to-end basis: “functions at the lower levels may be redundant or of little value when compared to the cost of providing them at the higher level.”
Intelligence and application level features are left to the hosts. This includes things like processing messages and blocking requests. The core of the network (levels 1-3) is very simple. Application level features should not exist in lower layers

What are the examples of a violation of e2e principle?

- Firewalls are one example as they operate at lower levels (Network?) and can drop messages between hosts.
- Network Address Translation (NAT) is another example. A NAT will rewrite source/destination addresses from higher layers (Transport). They prevent direct communication between hosts

What is the EvoArch model?
It’s an attempt to explain why some protocols survive and others die off. It can also be used to explain the staying power of older, suboptimal protocols. It looks at two things, the number of protocols in an upper layer that depend on a protocol and the amount of competition a protocol has in the same layer.

Explain a round in the EvoArch model.
EvoArch is a discrete-time model that is executed over rounds:

1. Add random nodes in different layers
2. Make connections to lower level nodes based on generality probability
3. Update the node evolutionary values
4. Remove nodes that fall below a certain threshold

What are the ramifications of the hourglass shape of the internet?
IPv4, TCP, and UDP provide a stable framework through which there is an ever-expanding set of protocols at the lower layers (physical and data-link layers), as well as new applications and services at the higher layers. But at the same time, these same protocols have been difficult to replace or even modify significantly. TCP/UDP have a lot of products that depend on them. This acts as a shield for IPv4 which has TCP and UDP as products

Repeaters, hubs, bridges, routers operate on which layers?

- Repeaters - Layer 1 (Physical)
- Hubs - Layer 1 (Physical)
- Bridges - Layer 2 (Data link)
- Routers - Layer 3 (Network)
- Switches - Layer 2 (Data Link)

What is a (learning) bridge, and how does it "learn"?
A bridge is a Layer2 device which forwards frames to the next node. When a bridge first receives a message, it sends it to all links to learn which are correct. These learned values are stored in a forwarding table.
The bridge consults the forwarding table so that it only forwards frames on specific ports, rather than over all ports.

What is a distributed algorithm?
An algorithm that runs over many systems at the same time.
Wikipedia: "Distributed algorithms are a sub-type of parallel algorithm, typically executed concurrently, with separate parts of the algorithm being run simultaneously on independent processors, and having limited information about what the other parts of the algorithm are doing."

Explain the Spanning Tree Algorithm.
Find the shortest path to the root for each node in a graph. In the initiation, each node tells all other nodes that it is the root. Each node takes all messages from surrounding nodes and decides the root and the path to root based on:

- Node ID - The root of the configuration has a smaller ID

- Distance to root node if the roots have equal IDs
- Both roots IDs are the same and the distances are the same, then the node breaks the tie by selecting the configuration of the sending node that has with the smallest ID

In addition, a node stops sending configuration messages over a link (port), when the node receives a configuration message that indicates that it is not the root, e.g. when it receives a configuration message from a neighbor that: a) either closer to the root, or b) it has the same distance from the root, but it has a smaller ID.

What is the purpose of the Spanning Tree Algorithm?
To remove loops from a graph and provide the shortest path from each node to the root node. Removing loops prevents broadcast storms

Lesson 2: Transport and Application Layers

What does the transport layer provide?
The transport layer provides an end-to-end connection between two applications that are running on different hosts (on the same or different networks).

What Is a packet for the transport layer called?
Segment

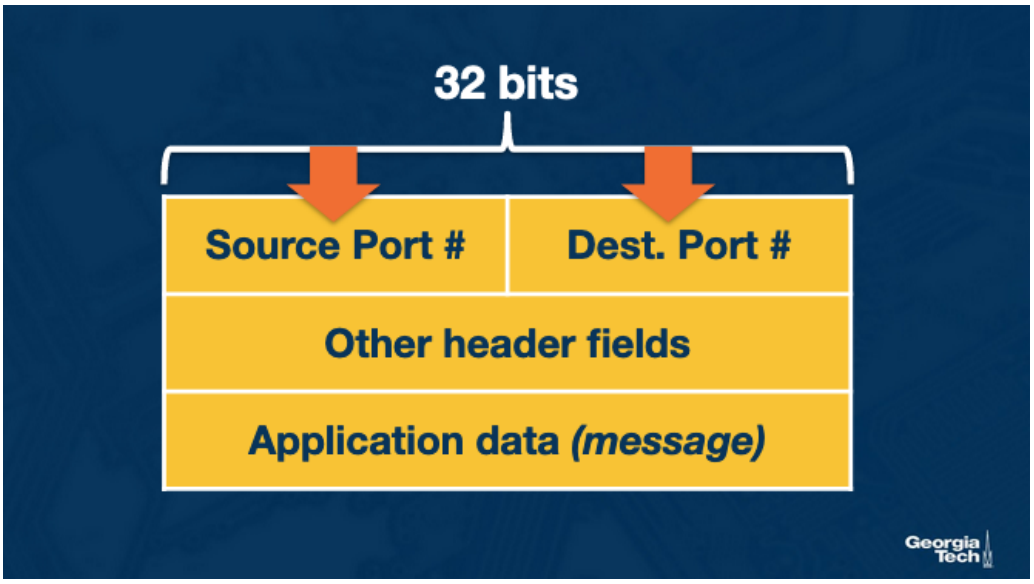
What are the two main protocols within the transport layer?

1. User datagram protocol (UDP)
2. Transmission Control Protocol (TCP)

What is multiplexing, and why is it necessary?
The sending host will need to gather data from different applications (different ports), and encapsulate each data chunk with header information (that will later be used in demultiplexing) to create segments, and then forward the segments to the network layer. We refer to this job as **multiplexing**

- Multiplexing is the functionality by which multiple applications in the same host can use the network simultaneously. It is provided by the transport layer.
- Multiplexing is needed to route the traffic coming into the host to the correct application (via sockets).
- Multiplexing uses sockets (transport protocol, IP address, port) to identify which application (on the local host) is listening to which remote application.

Describe the two types of multiplexing/demultiplexing.



Socket identifiers

A socket is one endpoint (IP address & port) of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to.

- **Connectionless (UDP):**
  - The identifier of a UDP socket is a two-tuple that consists of a destination IP address and a destination port number.
  - The UDP headers of a transport-layer segment include the source port and the destination port (No IP address, that is used in the network layer).
  - When multiplexing, a sender host takes a message from the application layer, appends the UDP headers (source port & destination port) to form a segment and forwards it to the network layer. This is encapsulation.
  - When demultiplexing, the transport layer at the receiving host identifies the correct socket by looking at the destination port in the incoming segment.
  - **Note:** the host will forward the segments to the same destination process via the same destination socket, even if the segments are coming from different source hosts and/or different source port numbers

- **Connection oriented (TCP):**
    - The identifier of a TCP socket is a four-tuple that consists of the source IP address and port number, and the destination IP address and port number.
    - The TCP headers of a transport-layer segment include the source port and the destination port (No IP address, that is used in the network layer), and other bits (like a special connection-establishment bit).
    - A connection is first established before client & server exchange data.

What are the differences between UDP and TCP?

- **UDP**
    - Connectionless (no three way handshake),
    - Unreliable (Best effort, no delivery guarantee, no congestion control). Provides very basic functionality and relies on the application-layer to implement the remaining.
    - Offers less delays and better control over sending data.
        - No congestion/flow control or similar mechanisms
        - No connection management overhead
    - The UDP header is 64 bits long, composed of:
        - Source port, destination port
        - Length (of UDP segment, header + data)
        - Checksum (1s complement of the sum of: source port, destination port, length and application data).

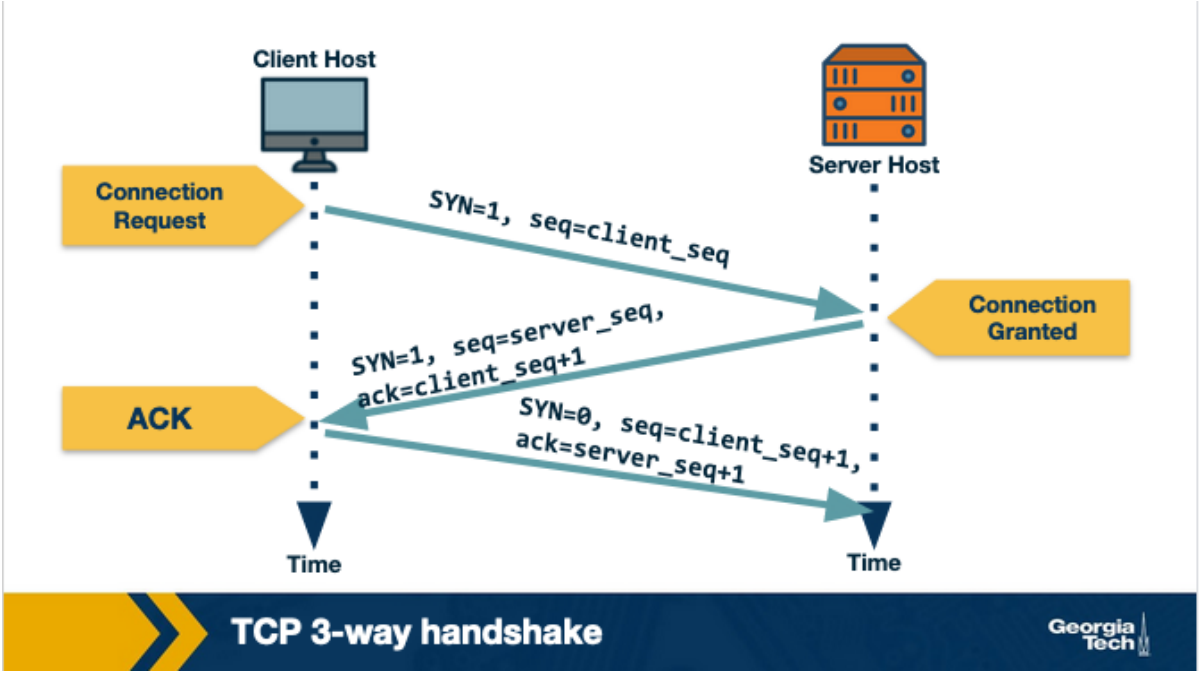
- **TCP**
    - Connection oriented
    - Reliable (TCP guarantees an in-order delivery of the application-layer data without any loss or corruption), flow control, congestion control. Provides some strong primitives with a goal to make end-to-end communication more reliable and cost-effective.

When would an application layer protocol choose UDP over TCP?

- **UDP**

    - Highest throughput, more control when sending data, for applications more sensitive to delays, but can handle data loss.
    - **Ex. App layer protocols:** NFS, SNMP, RIP, DNS, streaming data, internet telephony
- **TCP**
    - Delivery guarantee
    - **Ex. App layer protocols:** SMTP, HTTP, FTP, streaming data, internet telephony

Explain theTCP Three-way Handshake.



- Step 1: TCP Client sends special segment with no data - connection-request:

  SYN = 1

  seq = client_seq (Client-generated initial sequence number, random value)
- Step 2: Server sends back a special “connection-granted” segment called SYNACK - and allocates buffer & resources:

  SYN = 1

  ack = client_seq + 1

seq = server_seq (Server-generated initial sequence number, random value)

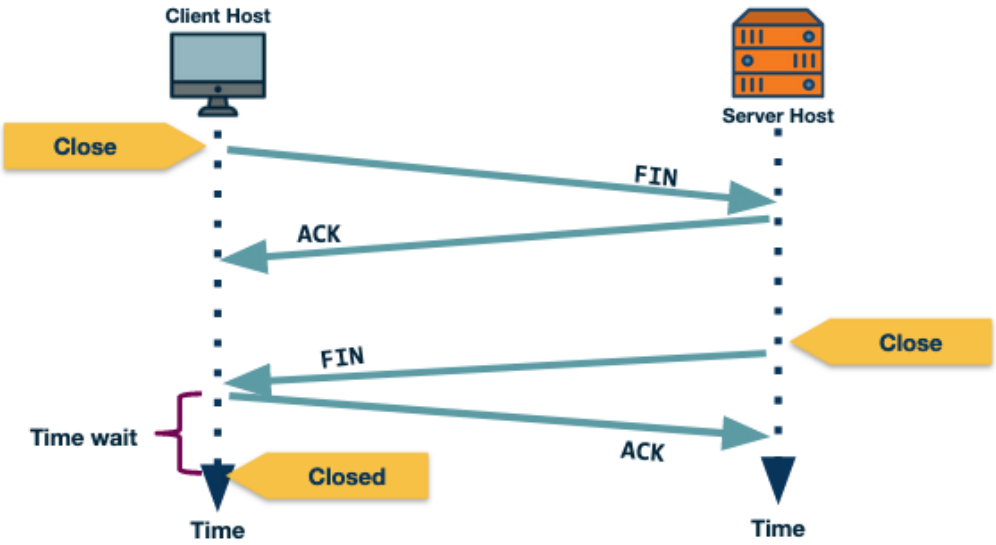
- Step 3: Client (receives SYNACK segment) sends acknowledgement and allocates buffer & resources:

  SYN = 0

  ack = server_seq + 1

  seq = client_seq + 1

Explain theTCP connection teardown.



- Step 1: Client sends special segment with no data:

  FIN = 1

- Step 2: Server acknowledges the connection closes request:

  ACK

- Step 3: Server sends segment to indicate the connection is closed:

  FIN = 1

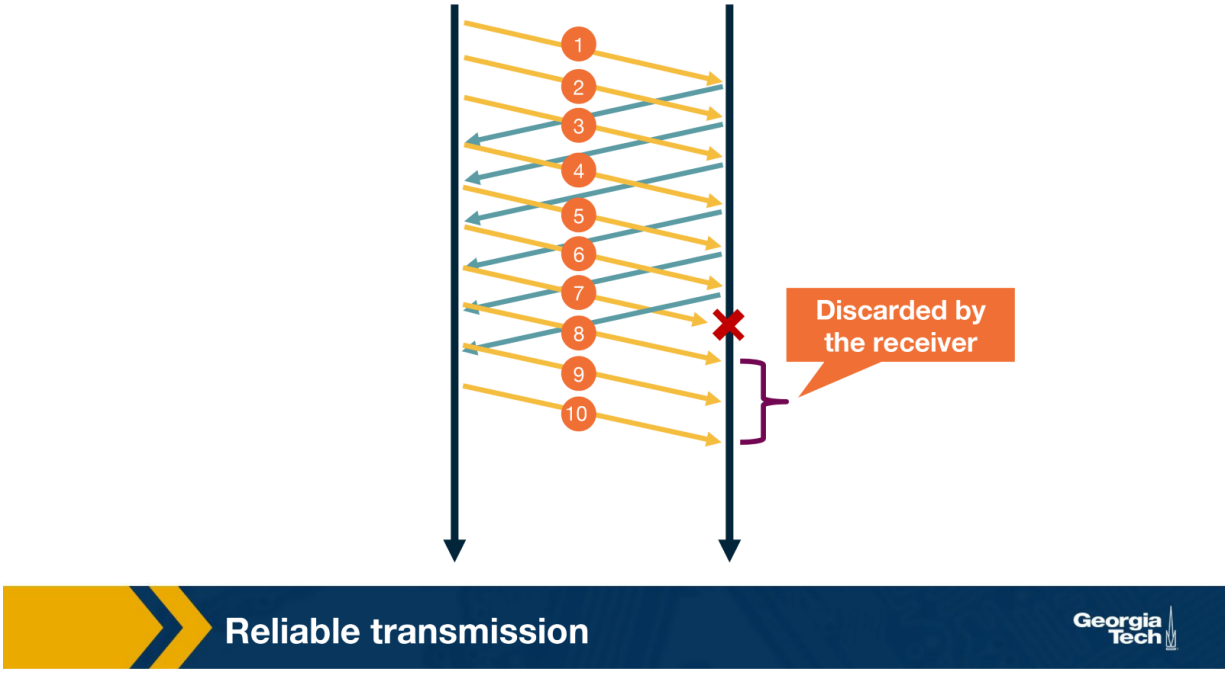
- Step 4: Client sends acknowledgement to server - and another one some time later in case it is lost:

  ACK

What is Automatic Repeat Request or ARQ?
ARQ is a mechanism used in the transport layer to ensure reliability.
If the sender has not received an acknowledgement from the remote host regarding a specific segment in a given period of time, it will automatically resend it.

What is Stop and Wait ARQ?
The simplest way to implement ARQ, the sender sends a packet and waits for its acknowledgement from the receiver. The trick is to determine the timeout, too small and you have too many retransmissions, too large, and you add delays. Has a low performance.

What is Go-back-N?
In Go-back-N, the receiver notifies the sender of a missing packet, by sending an ACK for the most recently received in-order packet. The sender would then send all packets from the most recently received in-order packet, even if some of them had been sent before. The receiver can simply discard any out-of-order received packets. A single packet error can cause a lot of unnecessary retransmissions.

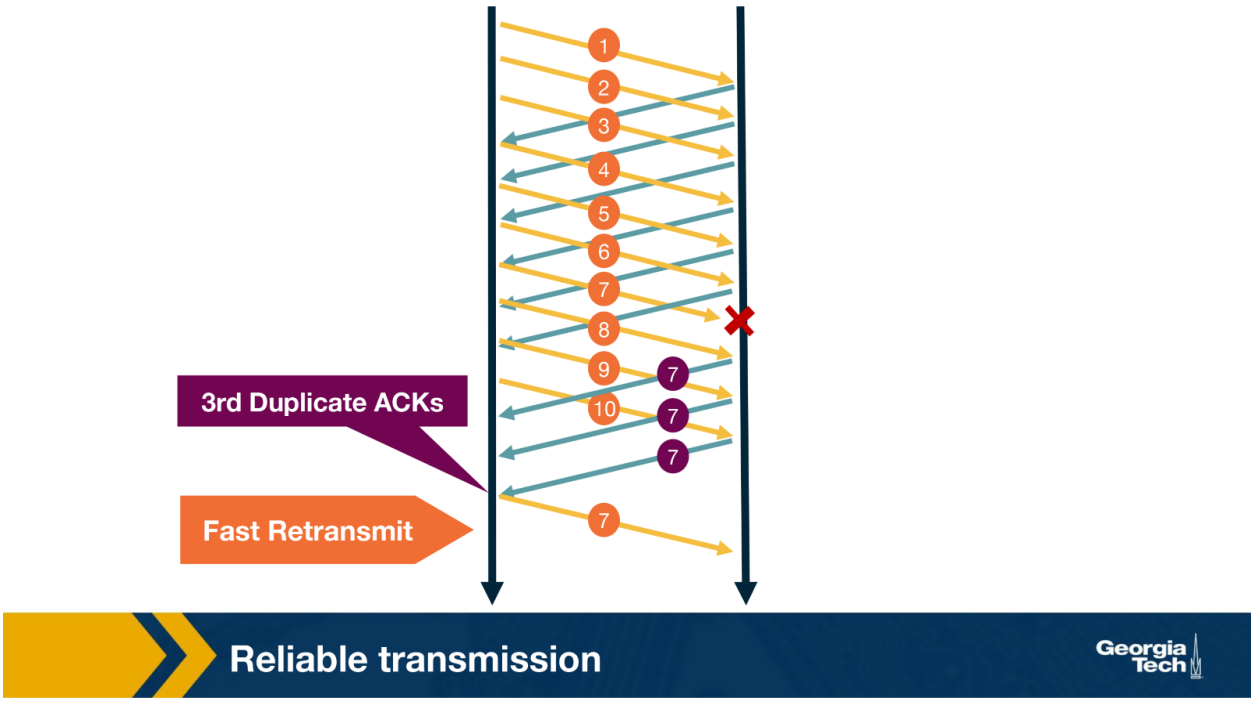


To address this, the sender can send at most N segments without waiting for acknowledgements, typically referred to as the window size. As it receives

acknowledgement from the receiver, it is allowed to send more packets based on the window size.

What is selective ACKing?
The receiver acknowledges a correctly received packet even if it is not in order. The out-of-order packets are buffered until any missing packets have been received at which point the batch of the packets can be delivered to the application layer. The sender retransmits only those packets that it suspects were not received. TCP would need to use a timeout as there is a possibility of ACKs getting lost in the network.

What is fast retransmit?
Fast retransmit occurs when the sender retransmits a segment that has not yet timed out, but for which it has received 3 duplicate ACKs.



What is transmission control and why do we need to control it?
Transmission control is a mechanism in the transport layer to control the transmission rate. It is used to determine and adapt the transmission rate given the changing host and network conditions.

UDP lets the application developers implement the mechanisms for transmission control, while TCP handles it for the user and deals with issues like fairness in using the network.

What is flow control and why do we need to control it?
Flow control is a mechanism used to dynamically control the sender's transmission rate based on the receivers' buffer availability (called receive window) to protect the receiver's buffer. This avoids overflowing the receiver's buffer.

What is congestion control?
Congestion control is a mechanism used to dynamically control the sender's transmission rate to avoid congestion in the network (longer queues, packet drops, etc.).

What are the goals of congestion control?

- Efficiency. High throughput, or network utilization.
- Fairness. Each user should have its fair share (we will assume, equal bandwidth) of the network bandwidth.
- Low delay. High throughput (with large buffers) would lead to long queues in the network leading to delays. Applications that are sensitive to network delays such as video conferencing will suffer. Thus, we want network delays to be small.
- Fast convergence. A flow should be able to converge to its fair allocation fast, so that even short flows will get their fair share of the network.

What is network-assisted congestion control?
We rely on the network layer to provide explicit feedback to the sender about congestion in the network. For instance, routers could use ICMP source quench to notify the source that the network is congested. However, under severe congestion, even the ICMP packets could be lost, rendering the network feedback ineffective

What is end-to-end congestion control?

As opposed to the previous approach, the network here does not provide any explicit feedback about congestion to the end hosts. Instead, the hosts infer congestion from the network behavior and adapt the transmission rate. This largely aligns with the end-to-end principle adopted in the design of the networks.

How does a host infer congestion?

- Through packet delay. As networks congest, queues in the router buffers build up. As packet round trip times increase (estimated based on ACKs) can be an indicator of congestion in the network, but packet delays tend to be variable, so it is not an straightforward indicator.
- Through packet loss. As the network congests, routers start dropping packets. Packets can be lost due to other reasons such as routing errors, hardware failure, TTL expiry, error in the links, or flow control problems. Early implementations of TCP used packet loss as a signal for congestion.

How does a TCP sender limit the sending rate?

A TCP sender cannot send faster than the slowest component, which is either the network or the receiving host. A sender uses ACKs as a pacing mechanism. TCP uses a congestion window which is similar to the receive window used for flow control. It represents the maximum number of unacknowledged data that a sending host can have in transit (sent but not yet acknowledged). TCP uses a probe-and-adapt approach in adapting the congestion window. Under regular conditions, TCP increases the congestion window trying to achieve the available throughput. Once it detects congestion, the congestion window is decreased.

$$LastByteSent - LastByteAcked <= min\{cwnd, rwnd\}$$

- **LastByteSent – LastByteAcked** represents the number of unacknowledged data.
- **cwnd** represents the congestion window
- **rwnd** represents the receiver window.

Explain Additive Increase / Multiplicative Decrease (AIMD) in the context of TCP.

- **Additive Increase**
    - Linearly increases the number of packets sent until a packet is lost (timeout)
    - The idea behind additive increase is to increase the window by one packet every RTT (Round Trip Time). So, in the additive increase part of the AIMD, every time the sending host successfully sends a cwnd number of packets it adds 1 packet to cwnd.
    - Also, in practice, this increase in AIMD happens incrementally. TCP doesn't wait for ACKs of all the packets from the previous RTT. Instead, it increases the congestion window size as soon as each ACK arrives. In bytes, this increment is a portion of the MSS (Maximum Segment Size).
    - Increment = MSS × (MSS / CongestionWindow)

- **Multiplicative Decrease**
    - Cuts the congestion window in half after a packet is lost to reduce network congestion
    - When the TCP sender detects that a timeout occurred, then it sets the CongestionWindow (cwnd) to half of its previous value. This decrease of the cwnd for each timeout corresponds to the "multiplicative decrease" part of AIMD. For example, suppose the cwnd is currently set to 16 packets. If a loss is detected, then cwnd is set to 8. Further losses would result in the cwnd being reduced to 4 and then to 2 and then to 1.
    - TCP Reno uses two types of packet loss detection as a signal of congestion.
        - First is the triple duplicate ACKs and is considered to be mild congestion. In this case, the congestion window is reduced to half of the original congestion window.
        - The second kind of congestion detection is timeout i.e. when no ACK is received within a specified amount of time. It is considered a more severe form of congestion, and the congestion window is reset to the Initial Window.

What is a slow start in TCP?

For new connections, to speed up the increase of the congestion window, the source host starts by setting cwnd to 1 packet and doubles it (exponential growth) after each RTT (Round Trip Time) until it reaches a *slow start threshold*, after which it starts using AIMD. For example, When it receives the ACK for this packet, it adds 1 to the current cwnd and sends 2 packets. Now when it receives the ACK for these two packets, it adds 1 to cwnd for each of the ACK it receives and sends 4 packets.

Is TCP fair in the case where two connections have the same RTT? Explain.
Yes, because both connections will be adjusting their congestion windows at a similar pace.

Is TCP fair in the case where two connections have different RTTs? Explain.
No, because the connections with the smaller RTT will be adjusting its congestion windows faster because it relies on received ACKs.

Explain how TCP CUBIC works.
Uses an aggressive scaleup, instead of Additive Increase, up to the previous Wmax where packet loss was experienced, then slows down, if no loss is experienced, it scales up again.
TCP CUBIC is RTT-fair because the scaleup time is based on the time elapsed since the last loss event and instead of the usual ACK-based timer used in TCP Reno.

Explain TCP throughput calculation.

$$BW < \frac{MSS}{RTT} * \frac{1}{\sqrt{p}}$$

BW == Bandwidth, MSS == Maximum Segment Size, RTT == Round Trip Time

P == probability loss (the network delivers 1 out of every p consecutive packets followed by a single packet loss).

Lesson 3: Intradomain Routing

What is the difference between forwarding and routing?
We refer to forwarding as the action of transferring a packet from an incoming link to an outgoing link within a single router.
By routing we refer to how routers work together using routing protocols to determine the good paths over which the packets travel from the source to the destination node.

What is the main idea behind link state routing algorithm?
The link state routing algorithm looks to determine the shortest paths (determined by the link costs) between a source node and all other nodes in the network. The link costs and the network topology are **known to all nodes.**

What is an example of a link state routing algorithm?
An example of a link state routing algorithm is Dijkstra's algorithm.
Updated -- Another exdample of a link state routing algorithm is Open Shortest Path First (OSPF) algorithm.

Walk through an example of the link state routing algorithm.

- **Initialization step**: We note that the algorithm starts with an initialization step, where we initialize all the currently known least-cost paths link state from u to its directly attached neighbors. We know these costs because they are the costs of the immediate links. For nodes in the network that are not directly attached to u, we initialize the cost path as infinity. We also initialize the set N' to include only the source node u.

- **Iteration step**: After the initialization step, the algorithm follows with a loop that is executed for every destination node v in the network. At each iteration, we look at the set of nodes that are not included in N', and we identify the node (say w) with the least cost path from the previous iteration. We add that node w into N'. For every neighbor v of w, we update D(v) with the new cost which is either the old cost from u to v (from the previous

iteration) or the known least path cost from source node u to w, plus the cost from w to v, whichever between the two quantities is the minimum.

- The algorithm exits by returning the shortest paths, and their costs, from the source node u to every other node v in the network.

What is the computational complexity of the link state routing algorithm?
The complexity of the algorithm is in the order of n squared O(n^2). The algorithm searches through n (n+1) / 2 nodes.

What is the main idea behind distance vector routing algorithm?
Each node maintains its own distance vector, with the costs to reach every other node in the network. The neighboring nodes exchange their distance vectors to update their own view of the network.

The DV routing algorithm is:

- iterative (the algorithm iterates until the neighbors do not have new updates to send to each other)
- asynchronous (the algorithm does not require the nodes to be synchronized with each other)
- distributed (nodes send information to one another, calculations are not happening in a centralized manner).
- based on the Bellman Ford Algorithm.

Walk through an example of the distance vector algorithm.
Each node x updates its own distance vector using the Bellman Ford equation: $Dx(y) = \min_v\{c(x,v) + Dv(y), Dx(y)\}$ for each destination node y in the network. A node x, computes the least cost to reach destination node y, by considering the options that it has to reach y through each of its neighbor v. So node x considers the cost to reach neighbor v, and then it adds the least cost from that neighbor v to the final destination y. It calculates that quantity over all neighbors v and it takes the minimum.

When does the count-to-infinity problem occur in the distance vector algorithm?

The count-to-infinity problem occurs in the distance vector algorithm when the cost of a link increases and two nodes think they can get to a third node through each other based on their previous outdated costs. This link cost change took a long time to propagate among the nodes of the network

How does poison reverse solve the count-to-infinity problem?
The way it works is that a node a, that uses node b to get a node c, will tell node b that the cost for its path to node c is infinity (Da(c)=infinity). Node b assumes that node a has no path to node c except through node b, so it will never send packets to node c via node a.

Poison reverse helps prevent the count-to-infinity problem only for 2 nodes, it does not solve a general count to infinity involving 3 or more nodes that are not directly connected.

What is the Routing Information Protocol (RIP)?
The Routing Information Protocol (RIP) is based on the Distance Vector protocol.

- The metric for choosing a path could be shortest distance, lowest cost or a load-balanced path.
- Routing updates between neighbors are done periodically, using RIP advertisements which contain information about sender’s distances to destination subnets.
- Each router maintains a routing table, which contains its own distance vector as well as the router's forwarding table.

Destination Subnet	Next Router	Number of Hops to Destination
w	A	*2*
y	B	*2*
z	B	*7*
x	-	*1*
...	...	...

Routing Information Protocol

Georgia Tech

- A routing table has three columns:
  - destination subnet,
  - identification of the next router along the shortest path to the destination,
  - number of hops to get to the destination along the shortest path.
- A routing table will have one row for each subnet in the AS (autonomous system).
- If a router does not hear from its neighbor at least once every 180 seconds, that neighbor is considered to be no longer reachable (broken link).
- Routers send request and response messages over UDP, using port number 520, which is layered on top of network-layer IP protocol. RIP is actually implemented as an application-level process.
- Some of the challenges with RIP include updating routes, reducing convergence time, and avoiding loops/count-to-infinity problems.

What is the Open Shortest Path First (OSPF) protocol?

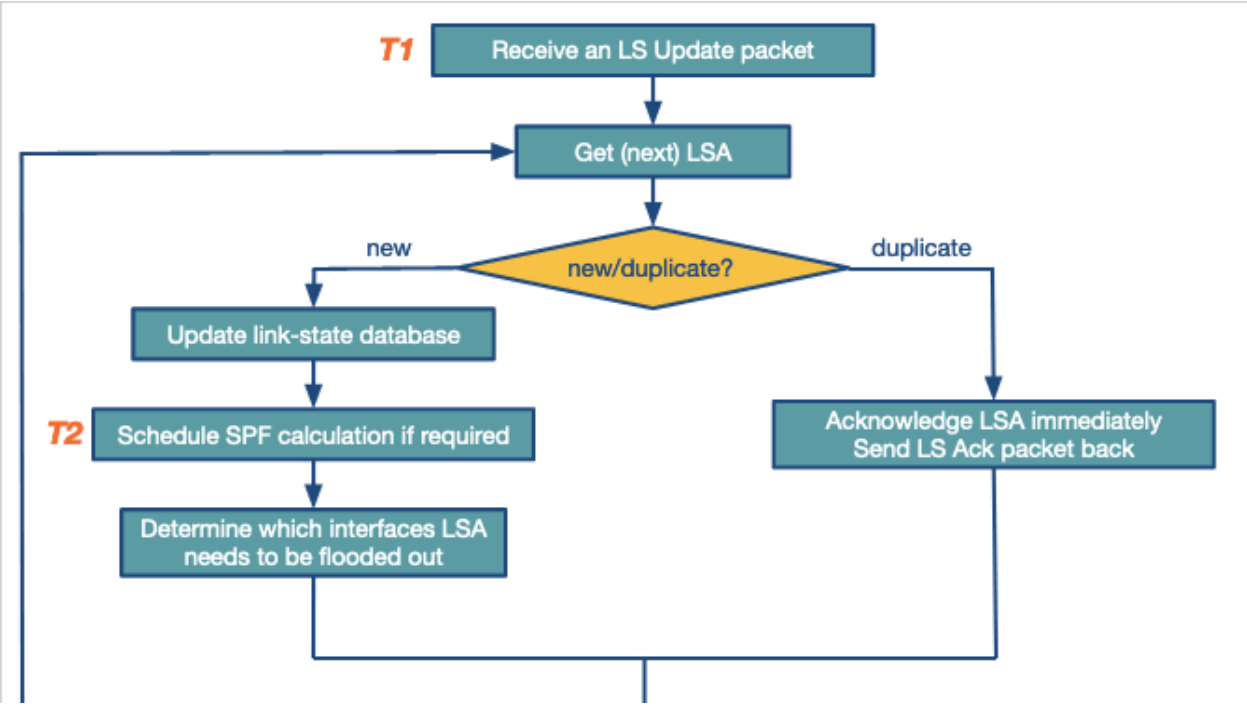
Open Shortest Path First (OSPF) is a routing protocol which uses a **link state routing algorithm** to find the best path between the source and the destination router.

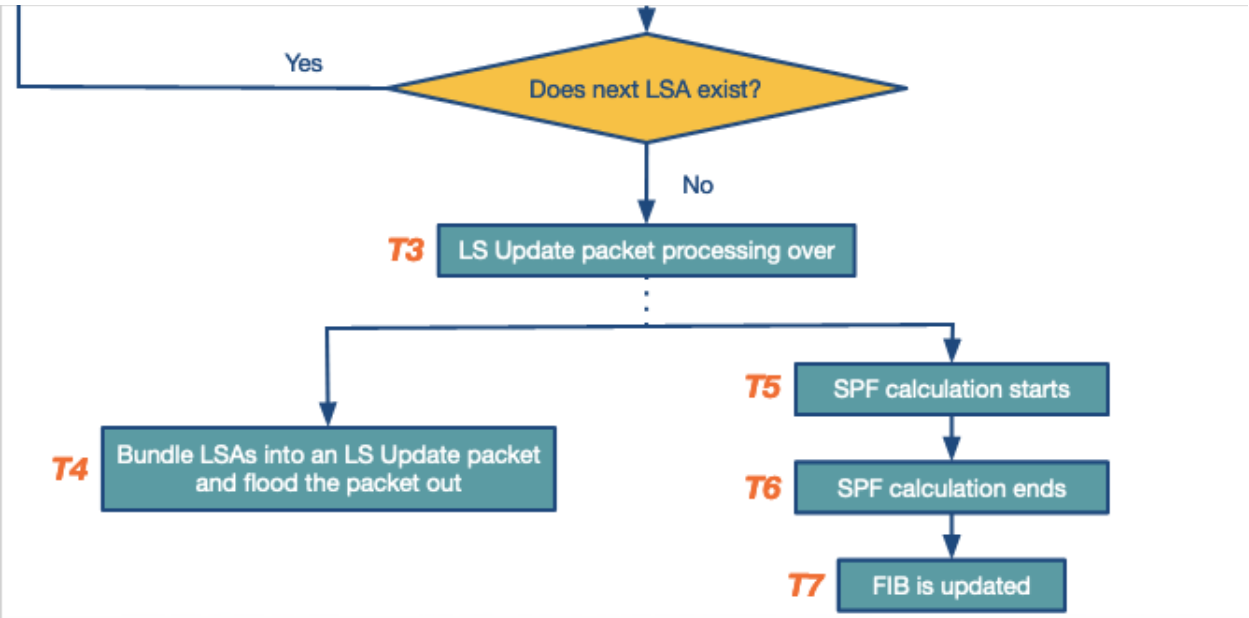
- OSPF was introduced as an advancement of the RIP Protocol.

- Include authentication of messages exchanged between routers, the option to use multiple same cost paths, and support for hierarchy within a single routing domain.
- **Hierarchy**
    - An OSPF autonomous system can be configured hierarchically into areas. Each area runs its own OSPF link-state routing algorithm, with each router in an area broadcasting its link state to all other routers in that area. Within each area, one or more area border routers are responsible for routing packets outside the area.
    - Exactly one OSPF area in the AS is configured to be the backbone area. The primary role of the backbone area is to route traffic between the other areas in the AS. The backbone always contains all area border routers in the AS and may contain non-border routers as well.
    - For packet routing between two different areas, it is required that the packet be sent through an area border router, through the backbone and then to the area border router within the destination area, before finally reaching the destination.
- **Operation**
    - First, a graph (topological map) of the entire AS is constructed. Then, considering itself as the root node, each router computes the shortest-path tree to all subnets, by running Djikstra's algorithm locally. The link costs have been pre-configured by a network administrator. The administrator has a variety of choices while configuring the link costs. For instance, s/he may choose to set them to be inversely proportional to link capacity, or set them all to one. Given a set of link weights, OSPF provides the mechanisms for determining least-cost path routing.
    - Whenever there is a change in a link's state, the router broadcasts routing information to all other routers in the AS, not just to its neighboring routers. It also broadcasts a link's state periodically even if its state hasn't changed.
- **Link State Advertisements**
    - Every router within a domain that operates on OSPF uses Link State Advertisements (LSAs). LSA communicates the router's local routing topology to all other local routers in the same OSPF area. In practice,

LSA is used for building a database (called the link state database) containing all the link states. LSAs are typically flooded to every router in the domain. This helps form a consistent network topology view. Any change in the topology requires corresponding changes in LSAs.

How does a router process advertisements?





1. Process begins when an LS update is received.
    a. Every Link State Advertisements (LSA) is unpacked, the OSPF protocol checks whether it is a new or a duplicate LSA, compared to the link-state (LS) DB.
        i. If it is a duplicate, it sends an LS ACK packet back immediately.
        ii. If it is new, it updates the LS DB, schedules a Shortest Path First (SPF) algorithm calculation and it determines which interface the LSA needs to be flooded out of.
2. Once the LS update packet has been processed, it prepares new LSAs updates into a new LS update packet and sends it to the next router.
3. After this, the SPF calculations are computed.
4. And finally the Forwarding Information Base (FIB) is updated. The information in the FIB is used to decide which outgoing interface card is the incoming packet forwarded to.

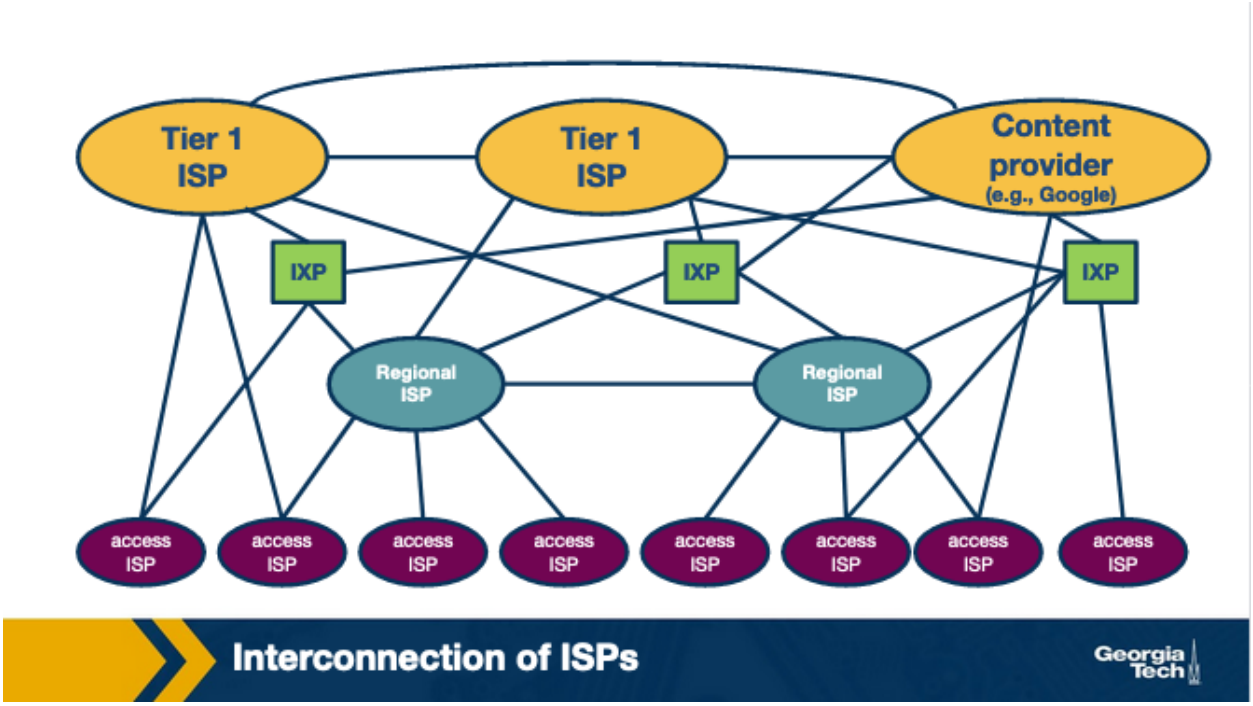
What is hot potato routing?

Hot potato routing is a technique/practice of choosing a path within the network, by choosing the closest egress point based on intradomain path cost (Interior Gateway Protocol/IGP cost).

Hot potato routing also effectively reduces the network’s resource consumption by getting the traffic out as soon as possible.

Lesson 4: AS Relationships and Interdomain Routing

Describe the relationship between ISPs, IXPs, and CDNs.



The basis of the internet ecosystem includes Internet Service Providers (ISPs), Internet Exchange Points (IXPs), and Content Delivery Networks (CDNs).

- **ISPs** can be categorized into three tiers or types: access ISPs (or Tier-3), regional ISPs (or Tier-2) and large global scale ISPs (or Tier-1). Tier-1 ISPs operate at a global scale, and essentially form the “backbone” network over which smaller networks can connect, regional ISPs connect to Tier-1 ISPs, and smaller access ISPs connect to regional ISPs.

- **IXPs** are interconnection infrastructures, which provide the physical infrastructure, where multiple networks (e.g. ISPs and CDNs) can interconnect and exchange traffic locally. As of 2019, there are approximately 500 IXPs around the world.

- **CDNs** are networks that are created by content providers with the goal of having greater control of how the content is delivered to the end-users, and

also to reduce connectivity costs. Some example CDNs include Google and Netflix.

This ecosystem we just described forms a hierarchical structure. There is competition at every level of the hierarchy. But, at the same time, competing ISPs need to cooperate to provide global connectivity to their respective customer networks. ISPs deploy multiple interconnection strategies depending on the number of customers in their network and also the geographical location of these networks.

What is an AS?

An **Autonomous System** (AS) is a group of routers (including the links among them) that operate under the same administrative authority.

Each AS implements its own set of policies, makes its own traffic engineering decisions and interconnection strategies, and also determines how the traffic leaves and enters the network.
Examples of AS can be ISPs and CDNs.

What kind of relationship does AS have with other parties?
Prevalent forms of business relationships between ASes:

- Provider-Customer relationship (or transit): This relationship is based on a financial settlement which determines how much the customer will pay the provider, so the provider forwards the customer's traffic to destinations found in the provider's routing table (including the opposite direction of the traffic as well).

- Peering relationship: In a peering relationship, two ASes share access to a subset of each other's routing tables. The routes that are shared between two peers are often restricted to the respective customers of each one. The

agreement holds provided that the traffic exchanged between the two peers is not highly asymmetric.

Note: Peering relationships are formed between Tier-1 ISPs but also between smaller ISPs. In the case of Tier-1 ISPs, the two peers need to be of similar size and handle similar amounts of traffic. Otherwise, the larger ISP would lack the incentive to enter a peering relationship with a smaller size ISP. In the case of peering between two smaller size ISPs, the incentive they both have is to save the money they would pay their providers by directly forwarding to each other their traffic, provided that there is a significant amount of traffic that is destined for each other (or each other's customers).

While peering allows networks to get their traffic forwarded without cost, provider ASes have a financial incentive to forward as much of their customers' traffic as possible. One major factor that determines a provider's revenue is the data rate of an interconnection. A provider usually charges in one of two ways:

- Based on a fixed price given that the bandwidth used is within a predefined range.
- Based on the bandwidth used. The bandwidth usage is calculated based on periodic measurements, e.g., on five min intervals. The provider then charges by taking the 95th percentile of the distribution of the measurements.

What is BGP?

The border routers of the ASes use the **Border Gateway Protocol (BGP)** to exchange routing information with one another. In contrast, the Internal Gateway Protocols (IGPs), operate within an AS and they are focused on "optimizing a path metric" within that network. Example IGPs include Open Shortest Paths First (OSPF), Intermediate System - Intermediate System (IS-IS), Routing Information Protocol (RIP), E-IGRP.

How does an AS determine what rules to import/export?

Exporting Routes

Deciding which routes to export is an important decision with business and financial implications. Advertising a route for a destination to a neighboring AS, means that this route may be selected by that AS and traffic will start to flow through. Deciding which routes to advertise is a policy decision and it is implemented through route filters; route filters are essentially rules that determine which routes an AS will allow to advertise to other neighboring ASes.

Let's look at the different types of routes that an AS (let's call it X) decides whether to export:

- **Routes learned from customers**. These are the routes that X receives as advertisements from its customers. Since provider X is getting paid to provide reachability to a customer AS, it makes sense that X wants to advertise these customer routes to as many other neighboring ASes as possible. This will likely cause more traffic towards the customer (through X) and hence more revenue to X.

- **Routes learned from providers**. These are the routes that X receives as advertisements from its providers. Advertising these routes doesn't make sense, since X does not have the financial incentive to carry traffic for its provider's routes. These routes are withheld from X's peers and other X's providers, but they are advertised to X's customers.

- **Routes learned from peers**. These are routes that X receives as advertisements from its peers. It doesn't make sense for X to advertise to a provider A the routes that it receives from another provider B. Because in that case, these providers A and B are going to use X to reach the advertised destinations without X making revenue. The same is true for the routes that X learns from peers.

Importing Routes

Similarly as exporting, ASes are selective about which routes to import based, primarily, on which neighboring AS advertises them and what type of business relationship is established. An AS receives route advertisements from its customers, providers and peers.

When an AS receives multiple route advertisements towards the same destination, from multiple ASes, then it needs to rank the routes before selecting which one to import. The routes that are preferred first are the customer routes, then the peer routes and finally the provider routes. The reasoning behind this ranking is that an AS

- wants to ensure that routes towards its customers do not traverse other ASes unnecessarily generating costs,
- uses routes learned from peers since these are usually “free” (under the peering agreement),
- and finally resorts to import routes learned from providers as these will add to costs.

What were originally the design goals of BGP? What was considered later?

Original design goals of the BGP protocol:

- **Scalability**: Manage the complications of the internet growth, while achieving convergence in reasonable timescales and providing loop-free paths.
- **Express routing policies**: BGP has defined route attributes that allow ASes to implement policies (which routes to import and export), through route filtering and route ranking. Each ASes routing decisions can be kept confidential, and each AS can implement them independently of one another.
- **Allow cooperation among ASes**: Each individual AS can make local decisions (which routes to import & export) while keeping these decisions confidential from other ASes.

Later considerations:

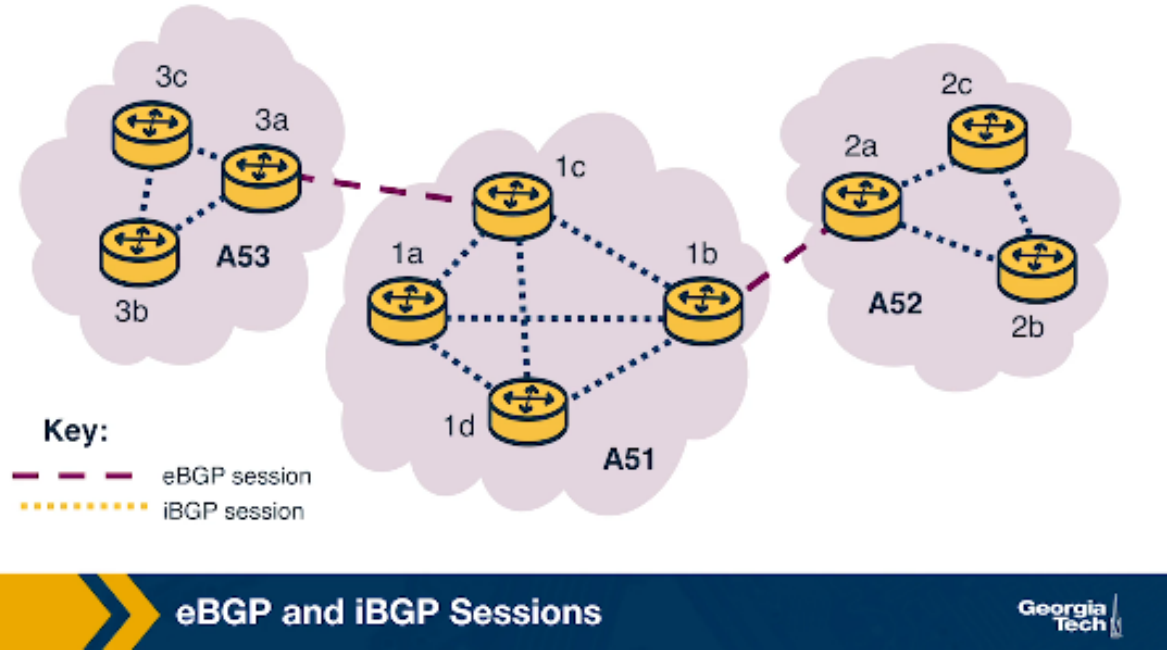
- **Security**: was not included in the original design goals for BGP. But as the complexity and size of the Internet has been increasing, so is the need to provide security measures. We notice an increasing need for **protection**

against malicious attacks, misconfigurations or faults, but also their **early detection**. These solutions have not been widely deployed or adopted due to multiple reasons that include difficulties to transition to new protocols and lack of incentives.

What are the basics of BGP?

A pair of routers, known as BGP peers, exchange routing information over a semi-permanent TCP port connection called a **BGP session**. To begin a BGP session a router will send an OPEN message to another router. Then the sending and receiving router will send each other announcements from their individual routing tables.

A BGP session between a pair of routers in two different ASes is called **external BGP (eBGP)** session, and a BGP session between routers that belong to the same AS is called **internal BGP (iBGP)** session.



Once a session is established between BGP peers, they exchange **BGP messages** to provide reachability information and enforce routing policies. We have two types of BGP messages:

- UPDATE
    - Announcements: These messages advertise new routes and updates to existing routes. They include several standardized attributes.
    - Withdrawals: These messages are sent when a previously announced route is removed. This could be due to some failure or a change in the routing policy.
- KEEPALIVE: These messages are exchanged to keep a current session going.

BGP prefix reachability: In the BGP protocol, destinations are represented by IP Prefixes. Each prefix represents a subnet or a collection of subnets that an AS can reach. Gateway routers running eBGP advertise the IP Prefixes they can reach according to the AS's specific export policy to routers in neighboring ASes. Then, using separate iBGP sessions, the gateway routers disseminate these routes for external destinations, to other internal routers according to the AS's import policy. Internal routers run iBGP to propagate the external routes to other internal iBGP speaking routers.

Path Attributes and BGP Routes
In addition to the reachable IP prefix field, advertised BGP routes consist of a number of BGP attributes. Two notable attributes are AS-PATH and NEXT-HOP.

- AS-PATH. Each AS -- as identified by the AS's autonomous system number (ASN) -- that the route passes through is included in the AS-PATH. This attribute is used to prevent loops and to choose between multiple routes to the same destination, the route with the shortest path.

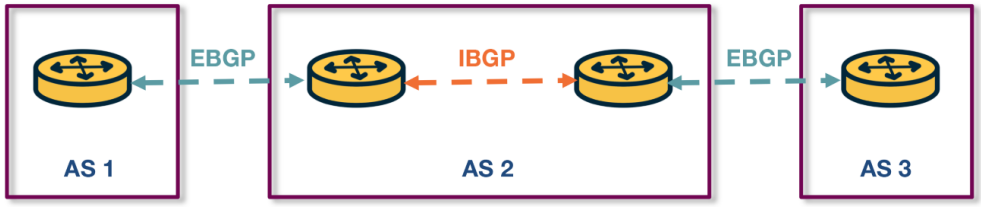
- NEXT-HOP. This attribute refers to the IP address (interface) of the next-hop router along the path towards the destination. Internal routers use the field to store the IP address of the border router. Internal BGP routers will have to forward all traffic bound for external destinations through the border

router. If there is more than one such router on the network and each advertises a path to the same external destination, NEXT-HOP allows the internal router to store in the forwarding table the best path according to the AS routing policy.

What is the difference between iBGP and eBGP?
iBGP & eBGP protocols are used to disseminate routes for external destinations.

eBGP is used for sessions between border routers of neighboring ASes and iBGP is used for sessions between internal routers of the same AS.

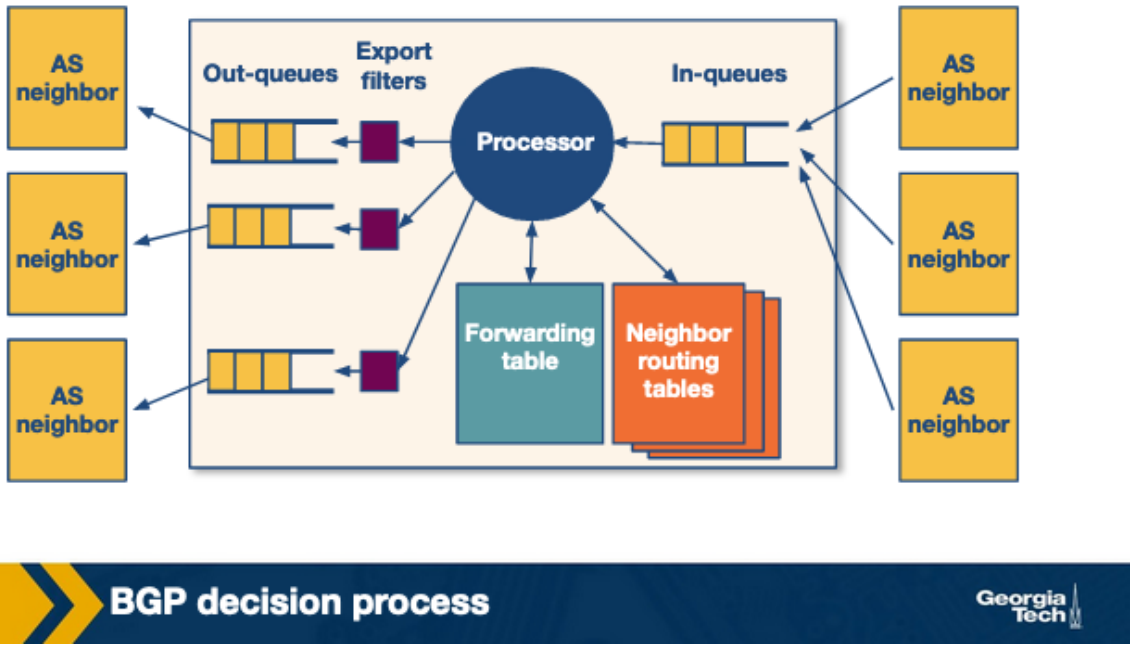
The dissemination of routes within the AS is done by establishing a full mesh of iBGP sessions between the internal routers. Each eBGP speaking router has an iBGP session with every other BGP router in the AS, so that it can send updates about the routes it learns (over eBGP).



What is the difference between iBGP and IGP-like protocols (RIP or OSPF)?
iBGP is not another IGP-like protocol (e.g. RIP or OSPF). IGP-like protocols are used to establish paths between the internal routers of an AS based on specific

costs within the AS. In contrast, iBGP is only used to disseminate external routes within the AS.

How does a router use the BGP decision process to choose which routes to import?



A router receives incoming BGP messages and processes them. It applies the import policies to exclude routes entirely from further consideration. Then the router implements the decision process to select the best routes that reflect the policy in place. The new selected routes are installed in the forwarding table. Finally, the router decides which neighbors to export the route to, by applying the export policy.

The decision process is how the router compares routes, by going through the list of attributes in the route advertisements. For each attribute, it selects the route

with the attribute value that will help apply the policy. If for a specific attribute, the values are the same, then it goes to the next attribute.

BGP decision process:

Step	Attribute	Controlled by
1	Highest LocalPref	local
2	Lowest AS path length	neighbor
3	Lowest origin type	neither
4	Lowest MED (Multi-Exit Discriminator)	neighbor
5	eBGP-learned over iBGP-learned	neither
6	Lowest IGP cost to border router	local
7	Lowest router ID (to break ties)	neither

What are 2 main challenges with BGP? Why?
Misconfiguration and faults. A possible misconfiguration or an error can result in an excessively large number of updates which in turn can result in route instability, router processor and memory overloading, outages, and router failures.

Solutions:

1. One way the risk can be reduced is by limiting the routing table size and also by limiting the number of route changes.
    - An AS can limit the routing table size using filtering.
        - Filter specific routes to encourage route aggregation.
        - Limit the number of prefixes advertised from a single source on a per-session basis.
        - Configure default routes into their forwarding tables.
        - Using route aggregation & exporting less specific prefixes where possible.

2. The other way is to limit the number of routing changes, specifically the propagation of unstable routes, by using a mechanism known as **flap damping**.
    - An AS will track the number of updates to a specific prefix over a certain amount of time. If the tracked value reaches a configurable value, the AS can suppress that route until a later time. Because this can affect reachability, an AS can be strategic about how it uses this technique for certain prefixes. For example, more specific prefixes could be more aggressively suppressed (lower thresholds), while routes to known destinations that require high availability could be allowed higher thresholds.

What is an IXP?

Internet Exchange Points (IXPs) are physical infrastructures that provide the means for ASes to interconnect and directly exchange traffic with one another.

- The ASes that interconnect at an IXP are called participant ASes.
- The physical infrastructure of an IXP is usually a network of switches that are located either in the same physical location, or they can be distributed over a region or even at a global scale.
- Typically, the infrastructure has fully redundant switching fabric that provides fault-tolerance, and the equipment is usually located in facilities such as data centers to provide reliability, sufficient power and physical security.
- The exchange of routes across the IXP is via BGP only.

What are four reasons for IXPs' increased popularity?

1. **IXPs are interconnection hubs handling large traffic volumes**: For some large IXPs (mostly located in Europe), the daily traffic volume is comparable to the traffic volume handled by global Tier 1 ISPs.

2. **Important role in mitigating DDoS attacks**: As IXPs have become increasingly popular interconnection hubs, they are able to observe the

traffic to/from an increasing number of participant ASes. In this role, IXPs can play the role of a “shield” to mitigate DDoS attacks and stop the DDoS traffic before it hits a participant AS.

3. **“Real-world” infrastructures with a plethora of research opportunities**: IXPs play an important role in today’s Internet infrastructure. Studying this peering ecosystem, the end-to-end flow of network traffic, and the traffic that traverses these facilities can help us understand how the Internet landscape is changing. IXPs also provide an excellent “research playground” for multiple applications. Such as security applications. For example BGP blackholing for DDoS mitigation, or applications for Software Defined Networking.

4. **IXPs are active marketplaces and technology innovation hubs**: IXPs are active marketplaces, especially in North America and Europe. They provide an expanding plethora of services that go beyond interconnection, for example DDoS mitigation, or SDN-based services. IXPs have been evolving from interconnection hubs to technology innovation hubs.

Which services do IXPs provide?

1. **Public peering**: The most well-known use of IXPs is public peering service - in which two networks use the IXP’s network infrastructure to establish a connection to exchange traffic based on their bilateral relations and traffic requirements. The costs required to set up this connection are - one-time cost for establishing the connection, monthly charge for using the chosen IXP port (those with higher speeds are more expensive) and perhaps an annual fee of membership in the entity owning and operating the IXP. However, the IXPs do not usually charge based on the amount of exchanged volume. They also do not usually interfere with bilateral relations between the participants unless there is a violation of the GTC. Even with the set-up costs, IXPs are usually cheaper than other conventional methods of exchanging traffic (such as relying on third parties which charge based on the volume of exchanged traffic). IXP participants also often experience better network performance and because of reduced delays and routing

efficiencies. In addition, many companies that are major players in the Internet space (such as Google) incentivize other networks to connect at IXPs by making it a requirement to peer with them.

2. **Private peering**: Most operational IXPs also provide a private peering service (Private Interconnects - PIs) that allow direct traffic exchange between two parties of a PI and don't use the IXP's public peering infrastructure. This is commonly used when the participants want a well-provisioned dedicated link capable of handling high-volume, bidirectional and relatively stable traffic.

3. **Route servers and Service level agreements**: Many IXPs also include service level agreements (SLAs) and free use of the IXP's route servers for participants. This allows participants to arrange instant peering with a large number of co-located participant networks using essentially a single agreement/BGP session.

4. **Remote peering through resellers**: Another popular service is IXP reseller/partner programs. This allows third parties to resell IXP ports wherever they have infrastructure connected to the IXP. These third parties are allowed to offer the IXP's service remotely, which allows networks that have little traffic to also use the IXP. This also enables remote peering - networks in distant geographic areas can use the IXP.

5. **Mobile peering**: Some IXPs also provide support for mobile peering - a scalable solution for interconnection of mobile GPRS/3G networks.

6. **DDoS blackholing**: A few IXPs provide support for customer-triggered blackholing, which allows users to alleviate the effects of DDoS attacks against their network.

7. **Free value-added services**: In the interest of 'good of the Internet', a few IXPs such as Scandinavian IXP Netnod offer free value-added services like Internet Routing Registry (IRR), consumer broadband speed tests9, DNS

root name servers, country-code top-level domain (ccTLD) nameservers, as well as distribution of the official local time through NTP.

How does a route server work?

To handle the volume of BGP sessions, IXP replaces bilateral BGP sessions (two-way BGP sessions between two ASes) with a **multilateral BGP peering session** using a Route Server (RS). A Route Server (RS)

- Collects and shares routing information from its peers or participants that connects with
- Executes its own BGP decision process and also re-advertise the resulting information (I.e. best route selection) to all RS's peer routers.

A typical routing daemon maintains a **Routing Information Base (RIB)** which contains all BGP paths that it receives from its peers - the **Master RIB**. The router server also maintains **AS-specific RIBs** to keep track of the individual BGP sessions they maintain with each participant AS.

RSes maintain two types of route filters: a) **Import filters** are applied to ensure that each member AS only advertises routes that it should advertise, b) **Export filters** which are typically triggered by the IXP members themselves to restrict the set of other IXP member ASes that receive their routes.

Example steps:

1. In the first step, AS X advertises a prefix p1 to the RS which is added to the route server's AS X specific RIB.

2. The route server uses the peer-specific import filter, to check whether AS X is allowed to advertise p1. If it passes the filter, the prefix p1 is added to the Master RIB.

3. The route server applies the peer-specific export filter to check if AS X allows AS Z to receive p1, and if true it adds that route to the AS Z-specific RIB.

4. Now, RS advertises p1 to AS Z with AS X as the next hop.

Lesson 5: Router Design and Algorithms (Part 1)

What are the basic components of a router?

The main components of a router are:

- the input/output ports,
- the switching fabric,
- the routing processor.

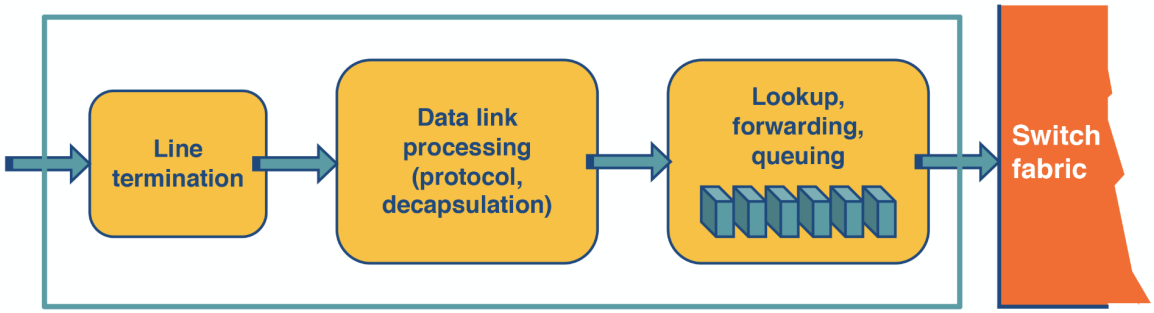
Explain the forwarding (or switching) function of a router.

This is the router's action to transfer a packet from an input link interface to the appropriate output link interface.
Forwarding takes place at very short timescales (typically a few nanoseconds), and is typically implemented in hardware.

The switching fabric moves the packets from input to output ports. What are the functionalities performed by the input and output ports?

Input ports:

1. The first function is to physically terminate the incoming links to the router.
2. Second, the data link processing unit decapsulates the packets.
3. Finally, the input ports perform the lookup function, they consult the forwarding table to ensure that each packet is forwarded to the appropriate output port through the switch fabric.

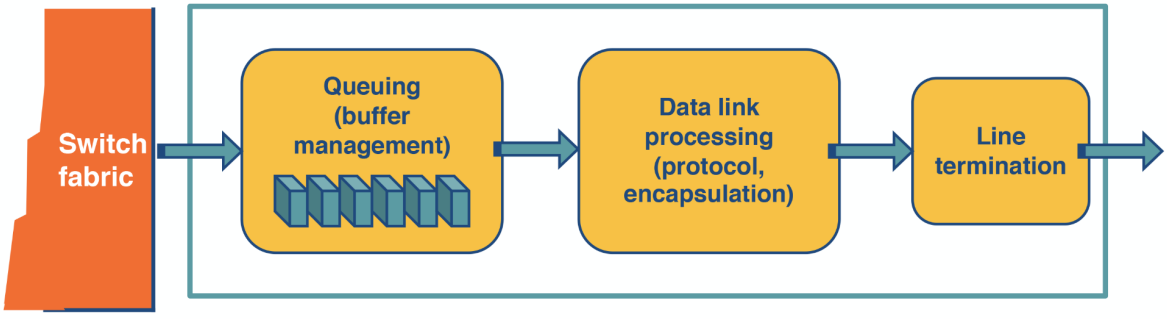


What's inside a router

Georgia Tech

Output ports:

1. An important function of the output ports is to receive and queue the packets which come from the switching fabric and then send them over to the outgoing link.



What's inside a router

Georgia Tech

What is the purpose of the router's control plane?

By control plane functions we refer to:

- Implementing the routing protocols,
- Maintaining the routing tables,
- Computing the forwarding table

All these functions are implemented in software in the routing processor, or these functions could be implemented by a remote controller.

What tasks occur in a router?

A router has input links and output links and **its main task is to switch a packet from an input link to the appropriate output link based on the destination address.**

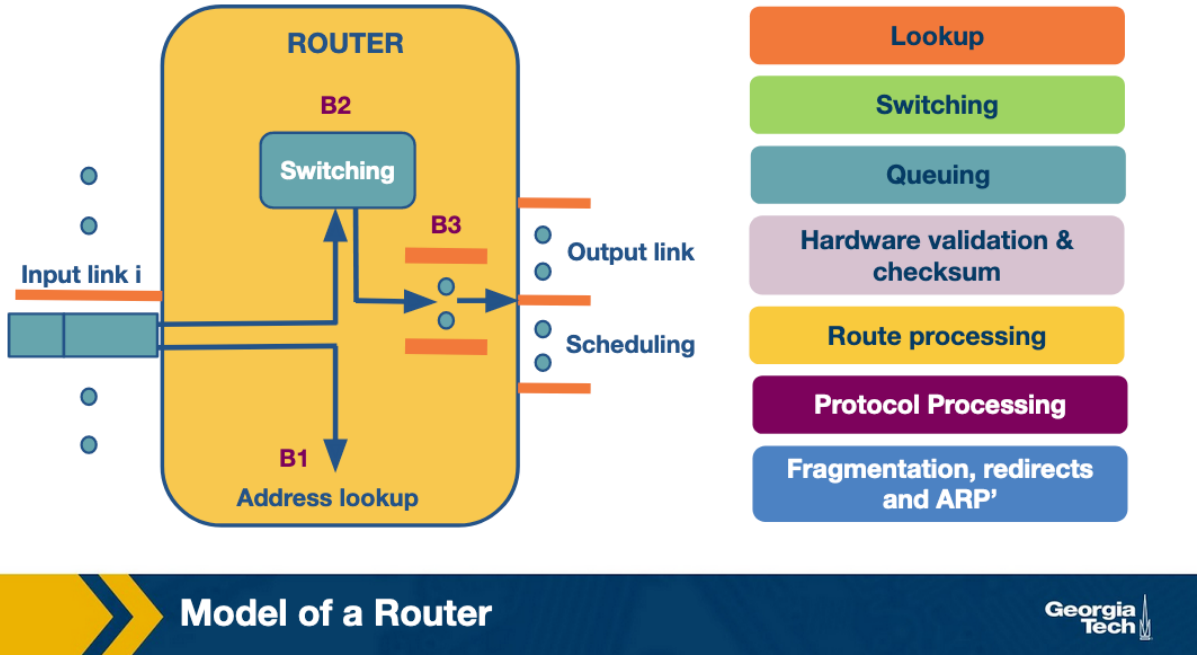
The most time-sensitive tasks: lookup, switching, and scheduling.

- **Lookup**: When a packet arrives at the input link, the router looks at the destination IP address and determines the output link by looking at the forwarding table (or Forwarding Information Base or FIB). The FIB provides a mapping between destination prefixes and output links.
- **Switching**: After lookup, the switching system takes over to transfer the packet from the input link to the output link. Modern fast routers use crossbar switches for this task. Though scheduling the switch (matching available inputs with outputs) is a difficult task because multiple inputs may want to send packets to the same output.
- **Queuing**: After the packet has been switched to a specific output, it will need to be queued (if the link is congested). The queue may be as simple as First-In-First-Out (FIFO) or it may be more complex (e.g. weighted fair queuing) to provide delay guarantees or fair bandwidth allocation.

Now, let's look at some less time-sensitive tasks that take place in the router.

- **Header validation and checksum**: The router checks the packet's version number, it decrements the time-to-live (TTL) field, and also it recalculates the header checksum.

- **Route processing**: The routers build their forwarding tables using routing protocols such as RIP, OSPF, and BGP. These protocols are implemented in the routing processors.
- **Protocol Processing**: The routers, in order to implement their functions, need to implement the following protocols: a) The simple network management protocol (SNMP) that provides a set of counters for remote inspection, b) TCP and UDP for remote communication with the router, c) Internet control message protocol (ICMP), for sending error messages, eg when time to live time is exceeded.



List and briefly describe each type of switching. Which, if any, can send multiple packets across the fabric in parallel?

The switching fabric moves the packets from input to output ports, and it makes the connections between the input and the output ports. There are three types of switching fabrics:

- **Memory**
    - Input/Output ports operate as I/O devices in an operating system, and they are controlled by the routing processor. When an input port receives a packet, it sends an interrupt to the routing processor and the packet is copied to the processor's memory. Then the processor

extracts the destination address and looks into the forward table to find the output port, and finally the packet is copied into that output's port buffer.

- **Bus**
    - In this case, the routing processor does not intervene as we saw the switching via memory. When an input port receives a new packet, it puts an internal header that designates the output port, and it sends the packet to the shared bus. Then all the output ports will receive the packet, but only the designated one will keep it. When the packet arrives at the designated output port, then the internal header is removed from the packet. Only one packet can cross the bus at a given time, and so the speed of the bus limits the speed of the router.
- **Crossbar** (interconnection network)
    - A crossbar switch is an interconnection network that connects N input ports to N output ports using 2N buses. Horizontal buses meet the vertical buses at crosspoints which are controlled by the switching fabric.
    - **Crossbar networks can carry multiple packets at the same time, as long as they are using different input and output ports.**

What are two fundamental problems involving routers, and what causes these problems?

The fundamental problems that a router faces revolve around:

- **Bandwidth and Internet population scaling**: These scaling issues are caused by:
    1. An increasing number of devices that connect to the Internet,
    2. Increasing volumes of network traffic due to new applications, and
    3. New technologies such as optical links that can accommodate higher volumes of traffic.
- **Services at high speeds**: New applications require services such as protection against delays in presence of congestion, and protection during attacks or failures. But offering these services at very high speeds is a challenge for routers.

What are the bottlenecks that routers face, and why do they occur?

- **Longest prefix matching**: As we have seen in previous topics, routers need to look up a packet's destination address to forward it. The increasing number of the Internet hosts and networks has made it impossible for routers to have explicit entries for all possible destinations. Instead routers group destinations into prefixes. But then, routers run into the problem of more complex algorithms for efficient longest prefix matching.
- **Service differentiation**. Routers are also able to offer service differentiation which means different quality of service (or security guarantees) to different packets. In turn, this requires the routers to classify packets based on more complex criteria that go beyond destination and they can include source or applications/services that the packet is associated with.
- **Switching limitations**. As we have seen, a fundamental operation of routers is to switch packets from input ports to output ports. A way to deal with high-speed traffic is to use parallelism by using crossbar switching. But at high speeds, this comes with its own problems and limitations (e.g. head of line blocking).
- **Bottlenecks about services**. Providing performance guarantees (quality of service) at high speeds is nontrivial. As is providing support for new services such as measurements and security guarantees.

Convert between different prefix notations (dot-decimal, slash, and masking).

- Dot decimal: e.g. of 16-bit prefix: 132.234
- Slash notation: Standard notation: A/L (where A=Address, L=Length) e.g.: 132.238.0.0/16
- Masking: We can use a mask instead of the prefix length. e.g.: The Prefix 123.234.0.0/16 is written as 123.234.0.0 with a mask 255.255.0.0

What is CIDR, and why was it introduced?

In the earlier days of the Internet, we used an IP addressing model based on classes (fixed length prefixes). With the rapid exhaustion of IP addresses, in 1993, the **Classless Internet Domain Routing (CIDR)** came into effect. CIDR essentially assigns IP addresses using arbitrary-length prefixes. CIDR has helped to decrease the router table size but at the same time it introduced us to a new problem: longest-matching-prefix lookup.

Name 4 takeaway observations around network traffic characteristics. Explain their consequences.

These challenges revolve around **lookup speed, memory, and update time**:

- Measurement studies on network traffic had shown a large number (in the order of hundred thousands, 250,000 according to a measurement study in the earlier days of the Internet) of concurrent flows of short duration. This already large number has only been increasing. This has a consequence that a caching solution would not work efficiently.
- The important element while performing any lookup operation is how fast it is done (lookup speed). A large part of the cost of computation for lookup is accessing memory.
- An unstable routing protocol may adversely impact the update time in the table: add, delete or replace a prefix. Inefficient routing protocols increase this value up to additional milliseconds.
- An important trade-off is memory usage. We have the option to use expensive fast memory (cache in software, SRAM in hardware) or cheaper but slower memory (e.g., DRAM, SDRAM).

Why do we need multibit tries?

Unibit trie requires a large number of memory accesses for lookup.

What is prefix expansion, and why is it needed?

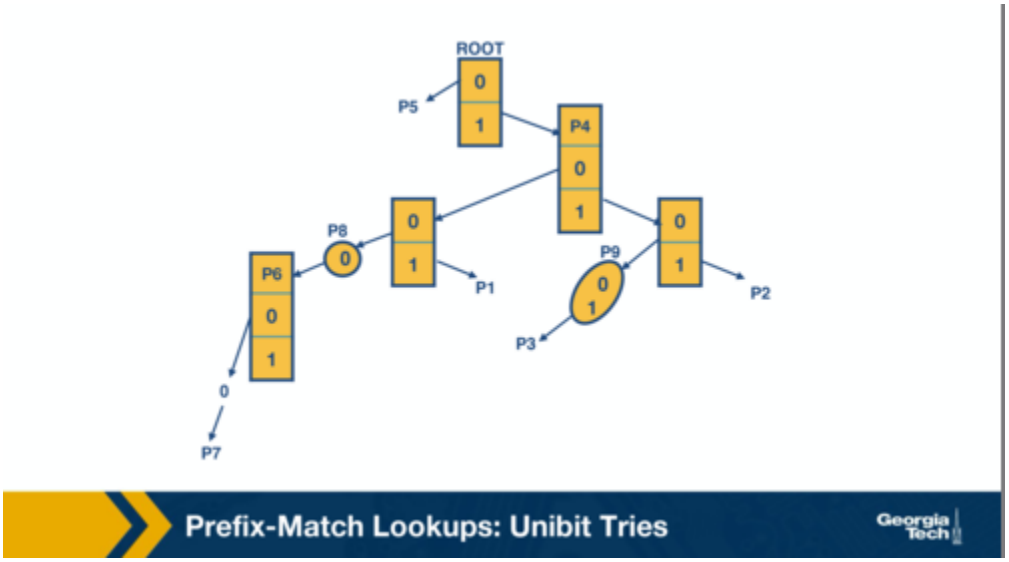
We expand a given prefix to more prefixes. We ensure that the expanded prefix is a multiple of the chosen **stride length**. At the same time we remove all lengths that are not multiples of the chosen stride length. We end up with a new database of prefixes, which may be larger (in terms of actual number of prefixes) but with fewer lengths. So, the expansion gives us **more speed with an increased cost of the database size**.

When we expand our prefixes, there may be a collision, i.e. when an expanded prefix collides with an existing prefix. In that case the expanded prefix gets dropped.

Perform a prefix lookup given a list of pointers for unibit tries, fixed-length multibit ties, and variable-length multibit tries.

Unibit Tries:

- Begin the search for a longest prefix match by tracing the trie path
- Continue the search until it fails i.e, no match or empty pointer
- When search fails the last known successful prefix traced in the path is the match .



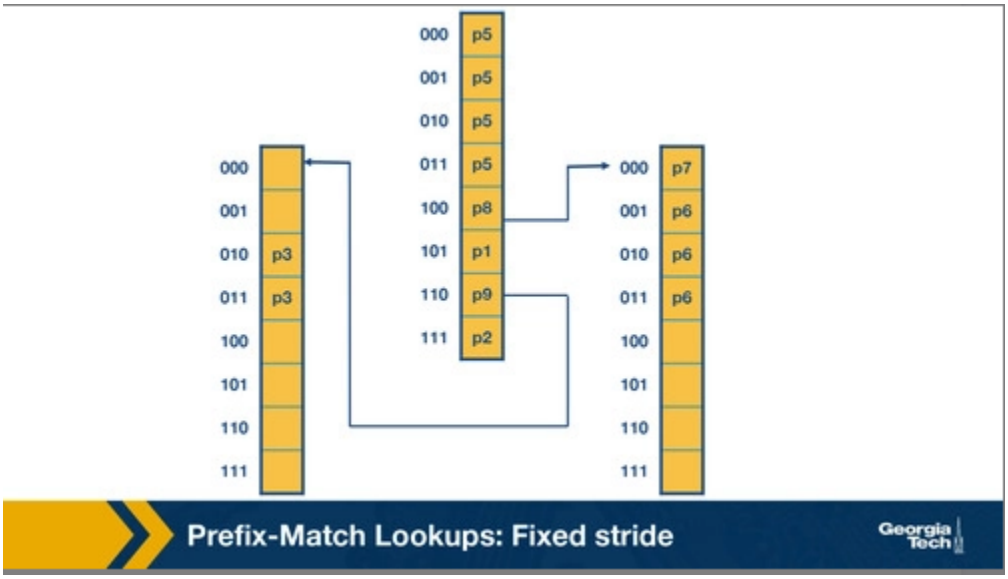
For example consider the above picture of unibit tries:

1. Assume that we are doing a longest prefix match for P1=101* (from our prefix database). We start at the root node and trace a 1-pointer to the right, then a 0-pointer to the left and then a 1-pointer to the right
2. For P7=100000*, we start at the root node and trace a 1-pointer to the right, then five 0-pointers the left

Fixed length Multi-bit tries:

- Every element in a trie represents two pieces of information: a pointer and a prefix value.
- The prefix search moves ahead with the present length in n-bits (3 in this case).
- When the path is traced by a pointer, we remember the last matched prefix (if any).

- Our search ends when an empty pointer is met. At that time, we return the last matched prefix as our final prefix match.



For example consider the above picture of fixed stride:

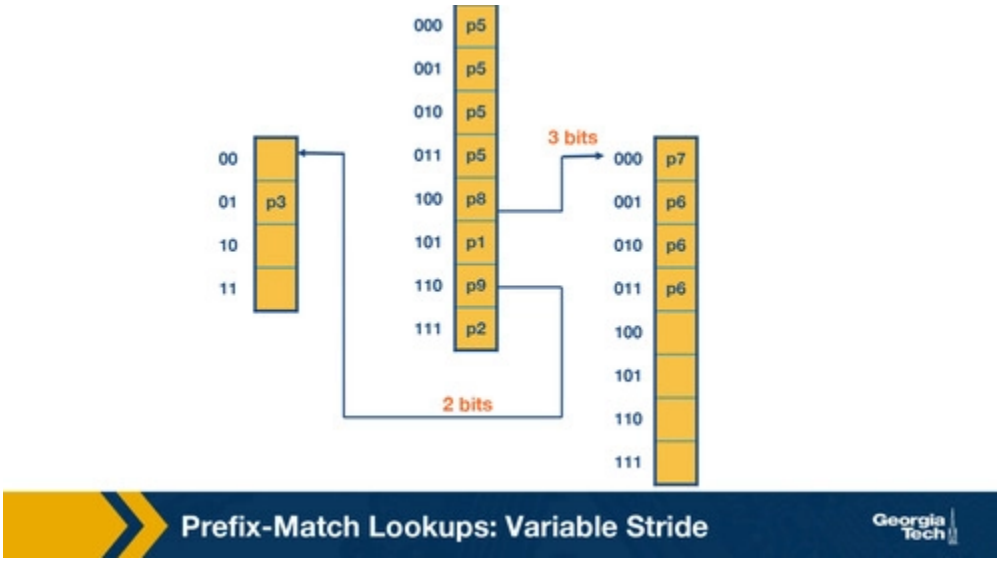
1.We consider an address A which starts with 001. The search for A starts with the 001 entry at the root node of the trie. Since there is no outgoing pointer, the search terminates here and returns P5.

2.Whereas if we search for 100000, the search would terminate with P7.

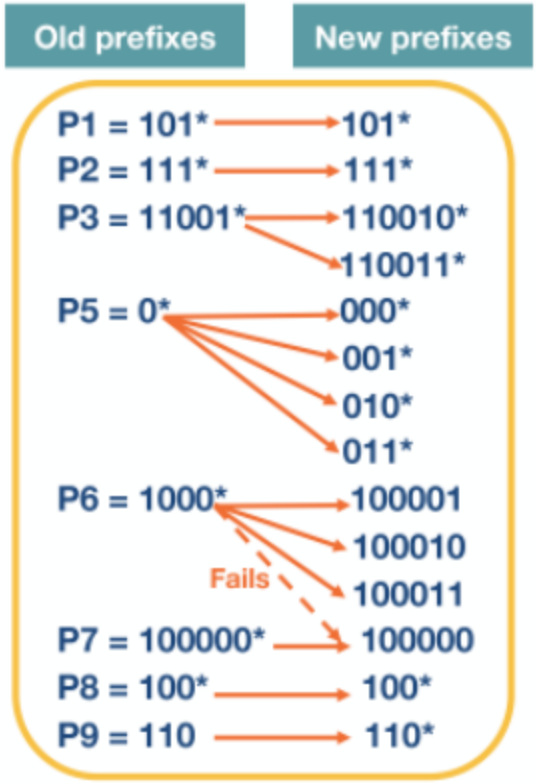
Variable length Multi-bit tries:

- We encode the stride of the trie node using a pointer to the node. The root node stays as is (in the previous scheme, fixed-length multibit ties).
- We note that the rightmost node still needs to examine 3 bits because of P7.
- But at the leftmost node need only to examine 2 bits, because P3 has 5 bits in total. So we can rewrite the leftmost node as in the figure below.

- So now we have 4 fewer entries than our fixed stride scheme. So by varying the strides we could make the prefix database smaller, and optimize for memory.



Perform a prefix expansion. How many prefix lengths do old prefixes have? What about new prefixes?



What are the benefits of variable-stride versus fixed-stride multibit tries?
By varying the strides we could make our prefix database smaller, and **optimize for memory**.

Some key points about fixed-stride trie:

1. Every element in a trie represents two pieces of information: a pointer and a prefix value.
2. The prefix search moves ahead with the preset length in n-bits (3 in this case)
3. When the path is traced by a pointer, we remember the last matched prefix (if any).
4. Our search ends when an empty pointer is met. At that time, we return the last matched prefix as our final prefix match.

Some key points about variable stride:

1. Every node can have a different number of bits to be explored
2. The optimizations to the stride length for each node are all done in pursuit of saving trie memory and the least memory access

3. An optimum variable stride is selected by using dynamic programming

Lesson 6: Router Design and Algorithms (Part 2) (Optional for Summer)

Why is packet classification needed?
What are three established variants of packet classification?
What are the simple solutions to the packet classification problem?
How does fast searching using set-pruning tries work?
What's the main problem with the set pruning tries?
What is the difference between the pruning approach and backtracking approach for packet classification with a trie?
What's the benefit of grid of tries approach?
Describe the"Take the Ticket"algorithm.
What is head-of-line problem?
How to avoid head-of-line problem using knockout scheme?
How to avoid head-of-line problem using parallel iterative matching?
DescribeFIFO with tail drop.
What are the reasons to make scheduling decisions more complex than FIFO?
DescribeBit-by-bit round Robin scheduling.
Bit-by-bit Round Robin provides fairness, what's the problem with this method?
Describe Deficit Round Robin (DRR).
What is a token bucket shaping?
In traffic scheduling, what is the difference between policing and shaping?
Howis a leaky bucket used for traffic policing and shaping?

Lesson 7: SDN (Part 1)

What spurred the development of Software Defined Networking (SDN)?
Software Defined Networking (SDN) arose as part of the process to make computer networks more programmable.
Computer networks are **very complex and especially difficult to manage** for two main reasons:

- Diversity of equipment on the network
- Proprietary technologies for the equipment

These made them **highly complex**, **slow to innovate**, and **drove up the costs** of running a network.
SDN divides the network into two planes (separation of tasks):

- control plane
- data plane.

What are the three phases in the history of SDN?

1. Active networks
2. Control and data plane separation
3. OpenFlow API and network operating systems

Summarize each phase in the history of SDN.

Active networks

- Intro
    - Slow and frustrating process to standardize protocols fostered the push for active networks trying to open up network control
    - Active networks with their network API went against the concept of keeping the core simple
    - 2 types of programming models in active networking:
        - Capsule model - carried in-band in data packets
        - Programmable router/switch model - established by out-of-band mechanisms
- Technology push - The pushes that encouraged active networking were:

    - Reduction in computation cost (more processing into the network).
    - Advancement in programming languages. (Java: platform portability, code execution safety, and VM (virtual machine) technology to protect the active node in case of misbehaving programs).
    - Advances in rapid code compilation and formal methods.
    - Funding from agencies such as DARPA (U.S. Defense Advanced Research Projects Agency) for a collection promoted interoperability among projects. There were no short-term use cases.
- Use pull - The use pulls for active networking were:
    - Network service provider frustration concerning the long timeline to develop and deploy new network services.
    - Third party interests to add value by implementing control at a more individualistic nature. This meant dynamically meeting the needs of specific applications or network conditions.
    - Researchers' interest in having a network that would support large-scale experimentation.
    - Unified control over middleboxes. Active networking envisioned unified control that could replace individually managing these boxes.
- Active networks contributions related to SDN:
    - Programmable functions in the network to lower the barrier to innovation.
        - While many early visions for SDN concentrated on increasing programmability of the control-plane, active networks focused on the programmability of the data-plane.
        - The concept of isolating experimental traffic from normal traffic has emerged from active networking and is heavily used in OpenFlow and other SDN technologies.
    - Network virtualization, and the ability to demultiplex to software programs based on packet headers.
    - The vision of a unified architecture for middlebox orchestration.
- Conclusion:
    - Did not see widespread deployment because it didn't solve a specific short-term problem and was too ambitious. It also did not focus on performance and security.

Control and data plane separation

- Intro
    - This phase was different from active networking in several ways:
        - It focused on spurring innovation by and for network administrators rather than end users and researchers.
        - It emphasized programmability in the control domain rather than the data domain.
        - It worked towards network-wide visibility and control rather than device-level configurations.
- Technology push - The technology pushes that encouraged control and data plane separation were:
    - Higher link speeds in backbone networks led vendors to implement packet forwarding directly in the hardware, thus separating it from the control-plane software.
    - Internet Service Providers (ISPs) found it hard to meet the increasing demands for greater reliability and new services (such as virtual private networks), and struggled to manage the increased size and scope of their networks.
    - Servers had substantially more memory and processing resources than those deployed one-two years prior. This meant that a single server could store all routing states and compute all routing decisions for a large ISP network. This also enabled simple backup replication strategies – thus, ensuring controller reliability.
    - Open source routing software lowered the barrier to creating prototype implementations of centralized routing controllers.
    - These pushes inspired two main innovations:
        - Open interface between control and data planes
        - Logically centralized control of the network
- Use pull
    - Selecting between network paths based on the current traffic load
    - Minimizing disruptions during planned routing changes
    - Redirecting/dropping suspected attack traffic
    - Allowing customer networks more control over traffic flow
    - Offering value-added services for virtual private network customers

- Control and data plane separation contributions related to SDN:
    - Logically centralized control using an open interface to the data plane.
    - Distributed state management.
- Conclusion
    - Did see widespread adoption because it had a more focused scope, and distinguished between control and data planes. This made it easier to focus on innovation in a specific plane.

<u>OpenFlow API and network operating systems</u>

- Intro
    - The basic working of an OpenFlow switch is as follows. Each switch contains a table of packet-handling rules. Each rule has a pattern, list of actions, set of counters and a priority. When an OpenFlow switch receives a packet, it determines the highest priority matching rule, performs the action aossociated with it and increments the counter.
- Technology push - OpenFlow was adopted in the industry, unlike its predecessors. This could be due to:
    - Before OpenFlow, switch chipset vendors had already started to allow programmers to control some forwarding behaviors.
    - This allowed more companies to build switches without having to design and fabricate their own data plane.
    - Early OpenFlow versions built on technology that the switches already supported. This meant that enabling OpenFlow initially was as simple as performing a firmware upgrade!
- Use pull:
    - OpenFlow came up to meet the need of conducting large scale experimentation on network architectures.
    - OpenFlow was useful in data-center networks – there was a need to manage network traffic at large scales.
    - Companies started investing more in programmers to write control programs, and less in proprietary switches that could not support new features easily. This allowed many smaller players to become competitive in the market by supporting capabilities like OpenFlow.
- Some key effects that OpenFlow had were:
    - Generalizing network devices and functions.

    - The vision of a network operating system.
    - Distributed state management techniques.

What is the function of the control and data planes?
The control plane contains the logic that controls the forwarding behavior of routers such as routing protocols and network middlebox configurations.
The data plane performs the actual forwarding as dictated by the control plane. For example, IP forwarding and Layer 2 switching are functions of the data plane.

Why separate the control from the data plane?
The reasons we separate the two are:

- Independent evolution and development
    - Routers only focus on routing.
    - Improvement in routing algorithms can take place without affecting any of the existing routers.
    - By limiting the interplay between these two functions, we can develop them more easily.
- Control from high-level software program
    - In SDN, we use software to compute the forwarding tables. Thus, we can easily use higher-order programs to control the routers' behavior.
    - The decoupling of functions makes debugging and checking the behavior of the network easier.

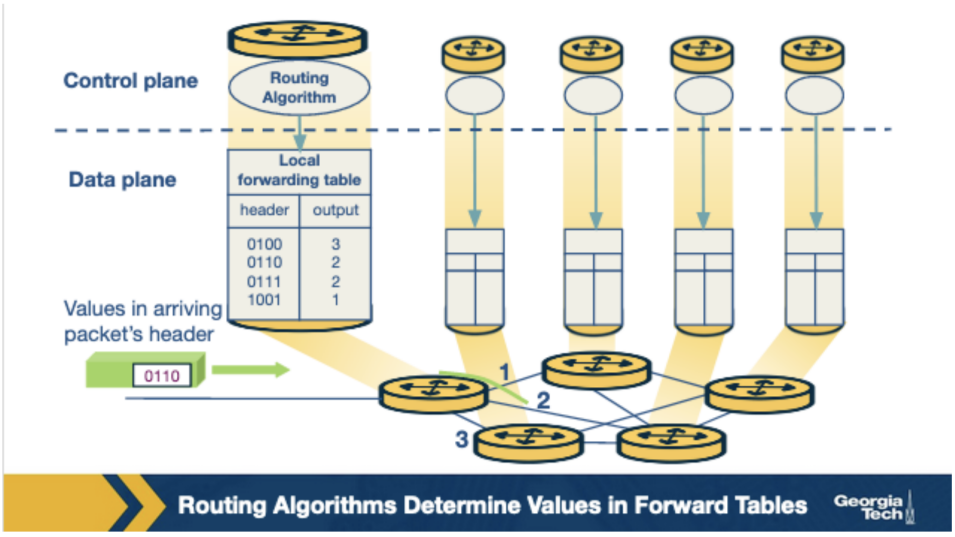
Why did the SDN lead to opportunities in various areas such as data centers, routing, enterprise networks, and research networks?
Separation of the control and data planes supports the independent evolution and development of both. Thus, the software aspect of the network can evolve independent of the hardware aspect. Since both control and forwarding behavior are separate, this enables us to use higher-level software programs for control. This makes it easier to debug and check the network's behavior.4

What is the relationship between forwarding and routing?

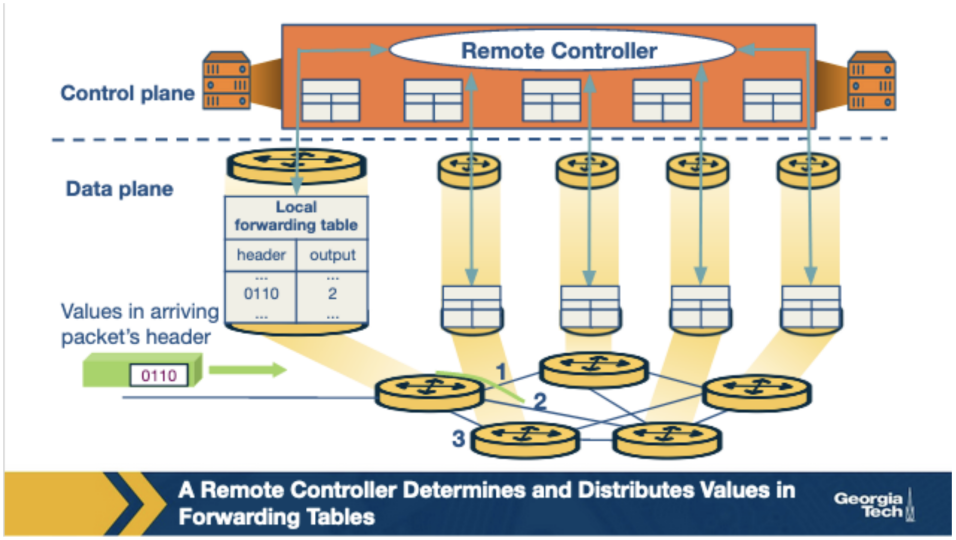
1. Forwarding
    a. Is the process inside a router of determining through which output link to send the packet it received at its input link.
    b. It could actually block the packet from exiting the router, if it is suspected to have been sent by a malicious router.
    c. It could also duplicate the packet and send it along multiple output links.
    d. Forwarding usually takes place in nanoseconds and is implemented in the hardware.
    e. Forwarding is a function of the **data plane**.
    f. A router looks at the header of an incoming packet and consults the forwarding table, to determine the outgoing link to send the packet to.
2. Routing
    a. Involves determining the path from the sender to the receiver across the network.
    b. Routers rely on routing algorithms for this purpose.
    c. It is an end-to-end process for networks.
    d. It usually takes place in seconds and is implemented in software.
    e. Routing is a function of the **control plane**.

What is the difference between a traditional and SDN approach in terms of coupling of control and data plane?

In the traditional approach, the **routing algorithms (control plane)** and **forwarding function (data plane)** are closely coupled. The router runs and participates in the routing algorithms. From there it is able to construct the forwarding table which consults it for the forwarding function.



In the SDN approach, there is a **remote controller** that computes and distributes the forwarding tables to be used by every router. This controller is physically separate from the router. We have a separation of the functionalities. The routers are solely responsible for forwarding, and the remote controllers are solely responsible for computing and distributing the forwarding tables. The controller is implemented in software, and therefore we say the network is software-defined.

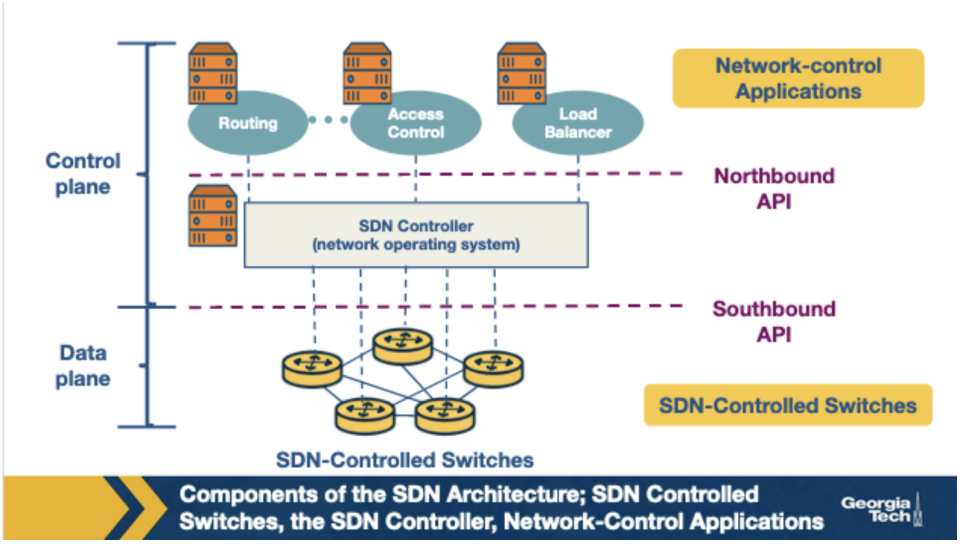


What are the main components of an SDN network and their responsibilities?

1. **SDN-controlled network elements**

a. The SDN-controlled network elements, sometimes called the infrastructure layer, is responsible for the **forwarding of traffic** in a network based on the rules computed by the SDN control plane.

2. **SDN controller**
    a. The SDN controller is a logically centralized entity that acts as an **interface** between the network elements and the network-control applications.
3. **Network-control applications**
    a. The network-control applications are programs that **manage the underlying network** by collecting information about the network elements with the help of an SDN controller.



What are the four defining features in an SDN architecture?

1) **Flow-based forwarding**: The rules for forwarding packets in the SDN-controlled switches can be computed based on any number of header field values in various layers such as the transport-layer, network-layer and link-layer. This differs from the traditional approach where only the destination IP address determines the forwarding of a packet.

2) **Separation of data plane and control plane**: The SDN-controlled switches operate on the data plane and they only execute the rules in the flow tables. Those rules are computed, installed, and managed by software that runs on separate servers.

3) **Network control functions**: The **SDN control plane**, (running on multiple servers for increased performance and availability) consists of two components: **the controller** and the **network applications**. The controller maintains up-to-date network state information about the network devices and elements (for example, hosts, switches, links) and provides it to the network-control applications. This information, in turn, is used by the applications to monitor and control the network devices.

4) **A programmable network**: The network-control applications act as the "brain" of the SDN control plane by managing the network. Example applications can include network management, traffic engineering, security, automation, analytics, etc. For example, we can have an application that determines the end-to-end path between sources and destinations in the network using Dijkstra's algorithm.

What are the three layers of an SDN controller?

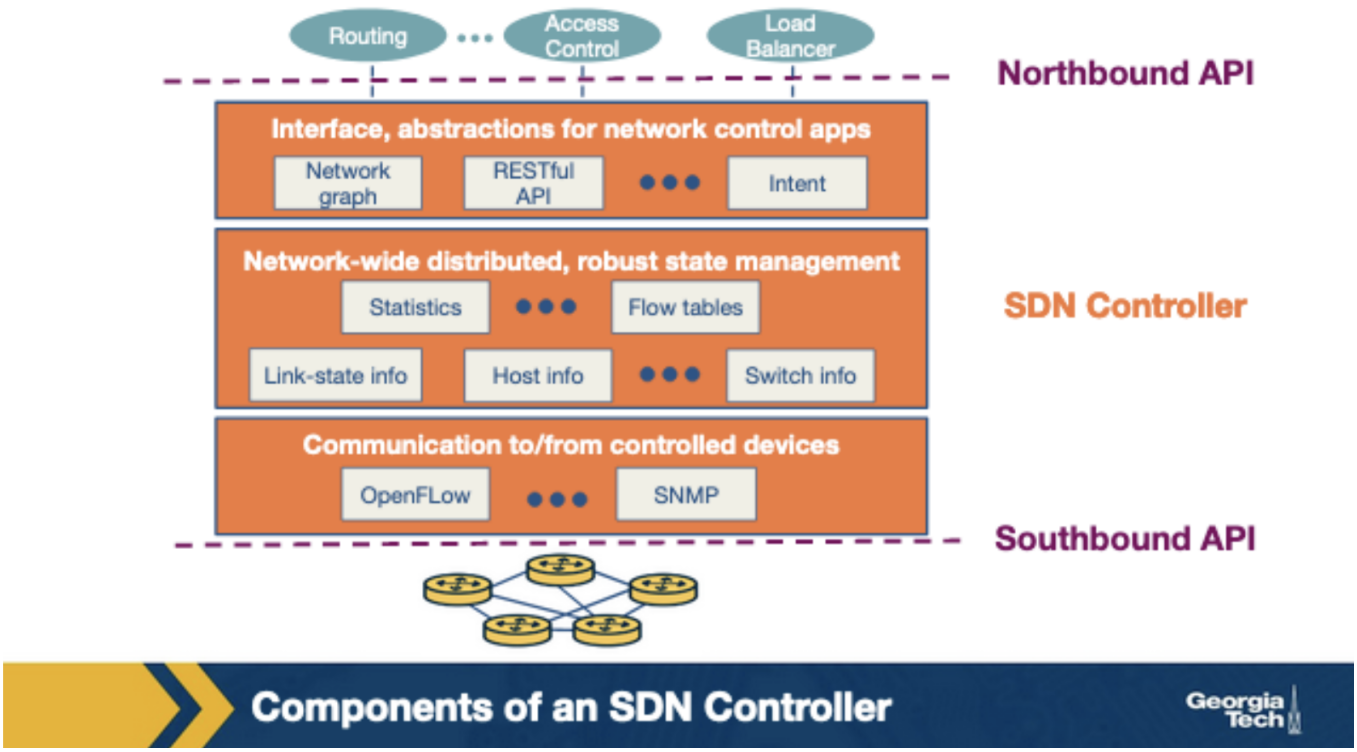
The SDN controller is a part of the SDN control plane and acts as an interface between the network elements and the network-control applications.

The SDN controller, although viewed as a monolithic service by external devices and applications, **is implemented by distributed servers to achieve fault tolerance, high availability and efficiency**. Despite the issues of synchronization across servers, many modern controllers such as *OpenDayLight* and *ONOS* have solved it and prefer distributed controllers to provide highly scalable services.

An SDN controller can be broadly split into three layers:

1. **Communication layer**:
    a. communicating between the controller and the network elements
    b. Controller's "southbound" interface
    c. OpenFlow is an example of this protocol
2. **Network-wide state-management layer**:
    a. stores information of network-state (hosts, links, switches, flow tables of the switches, etc.)

3. **Interface to the network-control application layer**:
    a. communicating between controller and applications
    b. Controller's "northbound" interface
    c. Network-control applications can read/write network state and flow tables in the controller's state-management layer.



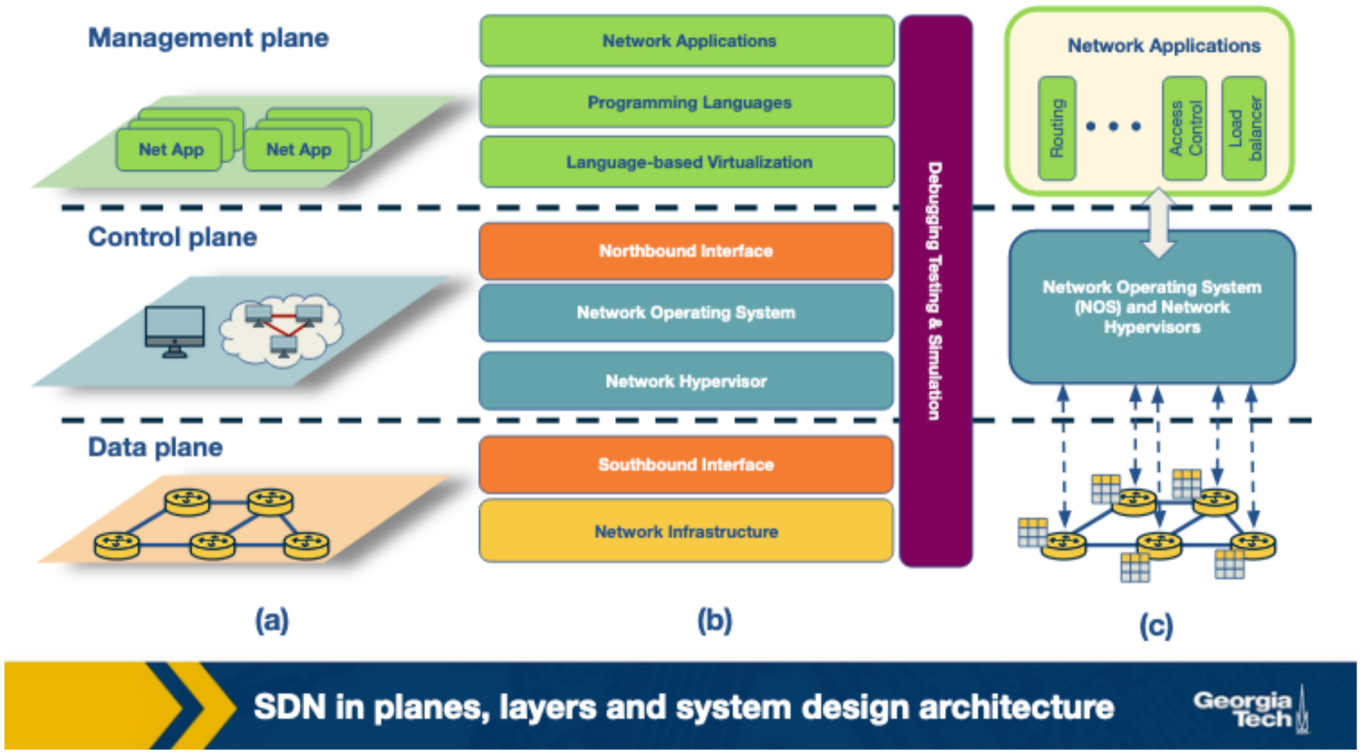
Lesson 8: SDN (Part 2)

Traditionally viewed, computer networks have three planes of functionality, which are all abstract logical concepts:

1. **Data plane**: These are functions and processes that forward data in the form of packets or frames.
2. **Control plane**: These refer to functions and processes that determine which path to use by using protocols to populate forwarding tables of data plane elements.
3. **Management plane**: These are services that are used to monitor and configure the control functionality, e.g. SNMP-based tools.

In short, say if a network policy is defined in the management plane, the control plane enforces the policy and the data plane executes the policy by forwarding the data accordingly.

Describe the three perspectives of the SDN landscape.
Three perspectives of the SDN landscape:
(a) a plane-oriented view,
(b) the SDN layers, and
(c) a system design perspective.



Describe the responsibility of each layer in the SDN layer perspective.

1. **Infrastructure**: routers, switches and other middlebox hardware, that are merely forwarding elements that do a simple forwarding task, and any logic to operate them is directed from the centralized control system.
2. **Southbound interfaces**: Separates control and data plane functionality. These APIs are tightly coupled with the forwarding elements of the underlying physical or virtual infrastructure. Has a widely acceptable norm (OpenFlow).
3. **Network virtualization**: The network infrastructure needs to provide support for arbitrary network topologies and addressing schemes. New advancements in SDN network virtualization such as VxLAN, NVGRE, FlowVisor, FlowN, NVP are promising.
4. **Network operating systems (NOS)**: A logically centralized controller that provides abstractions, essential services and common APIs to developers.
5. **Northbound interfaces**: A standard for Northbound interface is still an open problem, as are its use cases. Northbound interfaces are supposed to be a mostly software ecosystem. Another key requirement is the abstraction that guarantees programming language and controller independence.

6. **Language-based virtualization**: An important characteristic of virtualization is the ability to express modularity and allow different levels of abstraction. For example, using virtualization we can view a single physical device in different ways.
7. **Network programming languages**: Network programmability can be achieved using low-level or high-level programming languages. Using low-level languages, it is difficult to write modular code, reuse it and it generally leads to more error-prone development. High level programming languages in SDNs provide abstractions, make development more modular, code more reusable in the control plane, do away with device specific and low-level configurations, and generally allow faster development.
8. **Network applications**: These are the functionalities that implement the control plane logic and translate to commands in the data plane. SDNs can be deployed on traditional networks. Due to this, there is a wide variety of network applications such as routing, load balancing, security enforcement, end-to-end QoS enforcement, power consumption reduction, network virtualization, mobility management, etc.

Describe a pipeline of flow tables in OpenFlow.

A model derived from OpenFlow is currently the most widely accepted design of SDN data plane devices. It is based on a pipeline of flow tables where each entry of a flow table has three parts:

a) a matching rule,
b) actions to be executed on matching packets, and
c) counters that keep statistics of matching packets.

In an OpenFlow device, when a packet arrives, the lookup process starts in the first table and ends either with a match in one of the tables of the pipeline or with a miss (when no rule is found for that packet). Some possible actions for the packet include:

1. Forward the packet to outgoing port
2. Encapsulate the packet and forward it to controller
3. Drop the packet
4. Send the packet to normal processing pipeline

5. Send the packet to next flow table

What's the main purpose of southbound interfaces?
The Southbound interfaces or APIs are the separating medium between the control plane and data plane functionality.

What are three information sources provided by OpenFlow protocol?
There are three information sources provided by OpenFlow protocol to the Network Operating System (NOS):

1. **Event-based messages** that are sent by forwarding devices to controller when there is a link or port change
2. **Flow statistics** are generated by forwarding devices and collected by controller
3. **Packet messages** are sent by forwarding devices to controller when they do not know what to do with a new incoming flow

What are the core functions of an SDN controller?
Some base network service functions all controllers should provide include: topology, statistics, notifications, device management, along with shortest path forwarding and security mechanisms are essential network control functionalities that network applications may use in building its logic.

What are the differences between centralized and distributed architectures of SDN controllers?
Centralized controllers: In this architecture, we typically see a single entity that manages all forwarding devices in the network, which is a single point of failure and may have scaling issues. Also, a single controller may not be enough to handle a large number of data plane elements. Some enterprise class networks and data centers use such architectures, such as Maestro, Beacon, NOX-MT.

Distributed controllers: Unlike single controller architectures that cannot scale in practice, a distributed network operating system (controller) can be scaled to meet

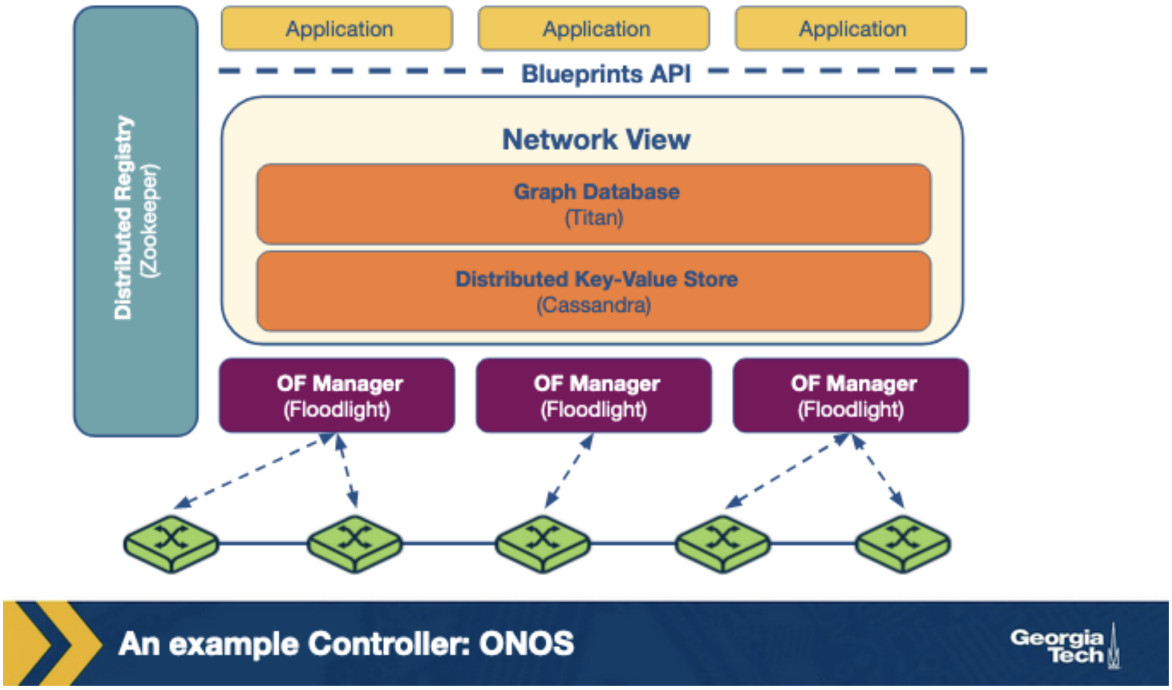
the requirements of potentially any environment - small or large networks. Distribution can occur in two ways: it can be a centralized cluster of nodes or physically distributed set of elements. Typically, a cloud provider that runs across multiple data centers interconnected by a WAN may require a hybrid approach to distribution - clusters of controllers inside each data center and distributed controller nodes in different sites. Properties of distributed controllers:

1. Weak consistency semantics
2. Fault tolerance

When would a distributed controller be preferred to a centralized controller?
When you are interested in:

1. Fault tolerance (avoid single point of failure)
2. Scalability (horizontal scaling)
3. Performance

Describe the purpose of each component of ONOS (Open Networking Operating System) is a distributed SDN control platform.



ONOS (Open Networking Operating System) is a distributed SDN control platform. It aims to provide a global view of the network to the applications, scale-out performance and fault tolerance.

There are several ONOS instances running in a cluster. The management and sharing of the network state across these instances is achieved by maintaining a global network view. This view is built by using the network topology and state information (port, link and host information, etc) that is discovered by each instance.

To make forwarding and policy decisions, the applications consume information from the view and then update these decisions back to the view. The corresponding *OpenFlow managers* receive the changes the applications make to the view, and the appropriate *switches* are programmed.

Titan, a graph database and a distributed key value store **Cassandra** are used to implement the view. The applications interact with the network view using the **Blueprints graph API**.

The distributed architecture of ONOS offers scale-out performance and fault tolerance. Each ONOS instance serves as the master OpenFlow controller for a group of switches. The propagation of state changes between a switch and the network view is handled solely by the master instance of that switch. The workload can be distributed by adding more instances to the ONOS cluster in case the data plane increases in capacity or the demand in the control plane goes up.

Zookeeper is used to maintain the mastership between the switch and the controller.

How does ONOS achieve fault tolerance?
To achieve fault tolerance, ONOS redistributes the work of a failed instance to other remaining instances. Each switch in the network connects to multiple ONOS instances with only one instance acting as its master. Each ONOS instance acts

as a master for a subset of switches. Upon failure of an ONOS instance, an election is held on a consensus basis to choose a master for each of the switches that were controlled by the failed instance. For each switch, a master is selected among the remaining instances with which the switch had established connection. At the end of election for all switches, each switch would have at most one new master instance.

What is P4?

P4 (Programming Protocol-independent Packet Processors) is a high-level programming language to configure switches which works in conjunction with SDN control protocols.

What are the primary goals of P4?

The following are the primary goals of P4:

- **Reconfigurability**: The way parsing and processing of packets takes place in the switches should be modifiable by the controller.
- **Protocol independence**: To enable the switches to be independent of any particular protocol, the controller defines a packet parser and a set of tables mapping matches and their actions. The packet parser extracts the header fields which are then passed on to the match+action tables to be processed.
- **Target independence**: The packet processing programs should be programmed independent of the underlying target devices. These generalized programs written in P4 should be converted into target-dependent programs by a compiler which are then used to configure the switch.

What are the two main operations of the P4 forwarding model?

The switches using P4 use a programmable parser and a set of match+action tables to forward packets. The tables can be accessed in multiple stages in a series or parallel manner. This contrasts with OpenFlow, which supports only fixed parsers based on predetermined header fields and only a series combination of match+actions tables.

The following are the two main operations of the P4 forwarding model:

(i) **Configure**: These sets of operations are used to program the parser. They specify the header fields to be processed in each match+action stage and also define the order of these stages.
(ii) **Populate**: The entries in the match+action tables specified during configuration may be altered using the populate operations. It allows addition and deletion of the entries in the tables.

In short, configuration determines the packet processing and the supported protocols in a switch whereas population decides the policies to be applied to the packets.

What are the applications of SDN? Provide examples of each application.

1. **Traffic Engineering**

This is one of the major areas of interest for SDN applications with main focus on optimizing the traffic flow so as to minimize power consumption, judiciously use network resources, perform load balancing, etc. With the help of optimization algorithms and monitoring the data plane load via southbound interfaces, the power consumption can be reduced drastically while still maintaining the desired goals of performance. Another use case of SDN applications is to automate the management of router configuration to reduce the growth in routing tables due to duplication of data. Large scale service providers also use SDN for traffic optimization to scale dynamically.

2. **Mobility and Wireless**

The existing wireless networks face various challenges in its control plane including management of the limited spectrum, allocation of radio resources and load-balancing. The deployment and management of various wireless networks (WLANS, cellular networks) is made easier using SDN. SDN-based wireless networks offer a variety of features including on-demand virtual access points (VAPs), usage of spectrum dynamically, sharing of wireless infrastructure, etc.

3. **Measurement and Monitoring**

The first class of applications in this domain aims to add features to other networking services. For example, new functions can be added easily to

measurement systems such as BISmark in an SDN-based broadband connection, which enables the system to respond to changes in network conditions. A second class of these applications aim to improve the existing features of SDNs using OpenFlow such as reducing the load on the control plane arising from collection of data plane statistics using various sampling and estimation techniques.

4. **Security and Dependability**

The applications in this area focus majorly on improving the security of networks. One approach of using SDN to enhance security is to impose security policies on the entry point to the network. Another approach is to use programmable devices to enforce security policies on a wider network. DDoS detection, an SDN application identifies and mitigates DDoS flooding attacks by leveraging the timely information collected from the network. Furthermore, SDN has also been used to detect any anomalies in the traffic, to randomly mutate the IP addresses of hosts to fake dynamic IPs to the attackers (OF-RHM), and monitoring the cloud infrastructures (CloudWatcher).

With regards to improving the security of SDN itself, there have been simple approaches like rule prioritizations for applications. However, there's still significant room for research and improvement in this area.

5. **Data Center Networking**

Data Center networking can be revolutionized by the use of SDN which aims to offer services such as live migration of networks, troubleshooting, real-time monitoring of networks among various other features. SDN applications can also help detect anomalous behavior in data centers by defining different models and building application signatures from observing the information collected from network devices in the data center. Any deviation from the signature history can be identified and appropriate measures can be taken. SDN also helps in performing dynamic reconfigurations of virtual networks involved in a live virtual network migration, which is an important feature of virtual networks in the cloud. LIME is one such SDN application which aims to provide live migration and FlowDiff is an application which detects abnormalities.

Which BGP limitations can be addressed by using SDN?

1) **Routing only on destination IP prefix** - The routing is decided based on the destination prefix IP of the incoming packet. There's no flexibility to customize rules for example based on the traffic application or the source/destination network.
2) **Networks have little control over end-to-end paths** - Networks can only select paths advertised by direct neighbors. Networks cannot directly control preferred paths but instead have to rely on indirect mechanisms such as "AS Path prepending".

What's the purpose of SDX?

SDX is an SDN-based architecture. It was proposed to implement multiple applications including:

- Application specific peering - Custom peering rules can be installed for certain applications, such as high-bandwidth video applications like Netflix or YouTube which constitute a significant amount of traffic volume.
- Traffic engineering - Controlling the inbound traffic based on source IP or port numbers by setting forwarding rules.
- Traffic load balancing - The destination IP address can be rewritten based on any field in the packet header to balance the load.
- Traffic redirection through middleboxes - Targeted subsets of traffic can be redirected to middleboxes.

Describe the SDX architecture.

In the SDX architecture, each AS has the illusion of its own virtual SDN switch that connects its border router to every other participant AS. For example, AS A has a virtual switch connecting to the virtual switches of ASes B and C.
Each AS can define forwarding policies as if it is the only participant at the SDX, without influencing how other participants forward packets on their own virtual switches. Each AS can have its own SDN applications for dropping, modifying, or

forwarding their traffic. The policies can also be different based on the direction of the traffic (inbound or outbound). An inbound policy is applied on the traffic coming from other SDX participants on a virtual switch. An outbound policy is applied to traffic from the participant's virtual switch port towards other participants. The SDX is responsible for combining the policies from multiple participants into a single policy for the physical switch.

What are the applications of SDX in the domain of wide area traffic delivery?

1. Application specific peering

ISPs prefer dedicated ASes to handle the high volume of traffic flowing from high bandwidth applications such as YouTube, Netflix. This can be achieved by identifying a particular application's traffic using packet classifiers and directing the traffic in a different path. However this involves configuring additional and appropriate rules in the edge routers of the ISP. This overhead can be eliminated by configuring custom rules for flows matching a certain criteria at the SDX.

2. Inbound traffic engineering

An SDN enabled switch can be installed with forwarding rules based on the source IP address and source port of the packets, thereby enabling an AS to control how the traffic enters its network. This is in contrast with BGP which performs routing based solely on the destination address of a packet. Although there are workarounds such as using AS path prepending and selective advertisements to control the inbound traffic using BGP, they come with certain limitations. An AS's local preference takes a higher priority for the outgoing traffic and the selective advertisements can lead to pollution of the global routing tables.

3. Wide-area server load balancing

The existing approach of load balancing across multiple servers of a service involves a client's local DNS server issuing a request to the service's DNS server. As a response, the service DNS returns the IP address of a server such that it balances the load in its system. This involves DNS caching which can lead to slower responses in case of a failure. A more efficient approach to load balancing can be achieved with the help of SDX, as it supports modification of the packet headers. A single anycast IP can be assigned to a service, and the destination IP

addresses of packets can be modified at the exchange point to the desired backend server based on the request load.

4. Redirection through middle boxes

SDX can be used to address the challenges in existing approaches to using middleboxes (firewalls, load balancers, etc). The placement of middleboxes are usually targeted at important junctions, such as the boundary of the enterprise networks with their upstream ISPs. To avoid the high expenses involved in placing middleboxes at every location in case of geographically large ISPs, the traffic is directed through a fixed set of middleboxes by the ISPs. This is done by manipulating routing protocols such as internal BGP to essentially hijack a subset of traffic and send it to a middlebox. This approach could result in unnecessary additional traffic being redirected, and is also limited by the fixed set of middleboxes. To overcome these issues, an SDX can identify and redirect the desired traffic through a sequence of middleboxes.

Lesson 9: Internet Security

What are the properties of secure communication?

1. **Confidentiality:** A message should only be available to the sender and receiver.
2. **Integrity:** The message should not be altered while in transit.
3. **Authentication**: The sender and receiver should provide proof that they are who they say they are.
4. **Availability**: The communication channel must be reliable, or all of the above is irrelevant.

How does Round Robin DNS (RRDNS) work?

A DNS server returns different permutations of the same list of DNS records. This is meant to distribute traffic evenly across the destination IPs.

How does DNS-based content delivery work?

The DNS server returns an IP that represents the “nearest edge server” to the client. The nearest server is calculated based on network topology and current link characteristics relative to where the client is located. TTL (time to live) is lower than that in RRDNS.

How do Fast-Flux Service Networks work?

The DNS records are short lived (shorter than RRDNS & DNS-based CDN). Typically the set of records returned are only a small percentage of the IP addresses available for that domain. These IP addresses belong to compromised machines act as proxies between the incoming request and control node/mothership, forming a resilient, robust, one-hop overlay network.

What are the main data sources to identify hosts that likely belong to rogue networks, used by FIRE (Finding Rogue networks system)?

Botnet command and control providers: Networks where bot masters are known to reside

Drive-by-download hosting providers: Networks which host malicious websites that download a file without user authorization when the victim visits a web page.

Phish housing providers: Networks where impersonation sites tend to proliferate

All networks will have the above. The difference is that legitimate networks will take these down within a few days while rogue networks will leave them up for long periods. The FIRE approach is to identify the most malicious networks as those which have the highest ratio of malicious IP addresses as compared to the total owned IP addresses of that AS.

The design of ASwatch is based on monitoring global BGP routing activity to learn the control plane behavior of a network. Describe 2 phases of this system.

Training Phase: The system learns from known sets of legitimate and malicious ASes. This information is used to train a machine learning algorithm

- **Rewiring Activity:** ASes that change routes often are more likely to be malicious.
- **IP Space Fragmentation and Churn:** Malicious ASes are more likely to advertise only a fraction of their IP space. This is to prevent having their entire IP space from being shut down at once.
- **BGP Routing Dynamics:** Malicious ASes tend to advertise IP prefixes for short amounts of time relative to a legitimate AS.

Operational Phase: The system uses the machine learning algorithm from the training phase to classify unknown ASes. Consistent low scores for an AS will identify it as malicious.

What are 3 classes of features used to determine the likelihood of a security breach within an organization?

1. **Mismanagement symptoms** – If there are misconfigurations in an organization's network, it indicates that there may not be policies in place to prevent such attacks or may not have the technological capability to detect these failures. This increases the likelihood of a breach. The features used are:
    - **Open Recursive Resolvers** – misconfigured open DNS resolvers
    - **DNS Source Port Randomization** – many servers still do not implement this

- **BGP Misconfiguration** – short-lived routes can cause unnecessary updates to the global routing table
- **Untrusted HTTPS Certificates** – can detect the validity of a certificate by TLS handshake
- **Open SMTP Mail Relays** – servers should filter messages so that only those in the same domain can send mails/messages.

2. **Malicious Activities** – Another factor to consider is the level of malicious activities that are seen to originate from the organization's network and infrastructure. We can determine this using spam traps,
darknet monitors, DNS monitors, etc. We create a reputation blacklist of the IP addresses that are involved in some malicious activities. There are ji3 such types of malicious activities:ji

- **Capturing spam activity** – for example, CBL, SBL, SpamCop
- **Capturing phishing and malware activities** – for example, PhishTank, SURBL
- **Capturing scanning activity** – for example, Dshield, OpenBL

3. **Security Incident Reports** – Data based on actual security incidents gives us the ground truth on which to train our machine learning model on. The system uses 3 collections of such reports to ensure a wider coverage area:

- **VERIS Community Database** – This is a public effort to collect cyber security incidents in a common format. It is maintained by the Verizon RISK team. It contains more than 5000 incident reports.
- **Hackmageddon** – This is an independently maintained blog that aggregates security incidents on a monthly basis.
- **The Web Hacking Incidents Database** – This is an actively maintained repository for cyber security incidents.

(BGP hijacking) What is the classification by affected prefix?
A malicious AS will advertise an existing prefix (Exact prefix hijacking) or sub-prefix (more specific) (Sub-prefix hijacking) in order to steal traffic from the legitimate AS. Another approach is to advertise an address that belongs to a different AS, but is not being advertised (Squatting).

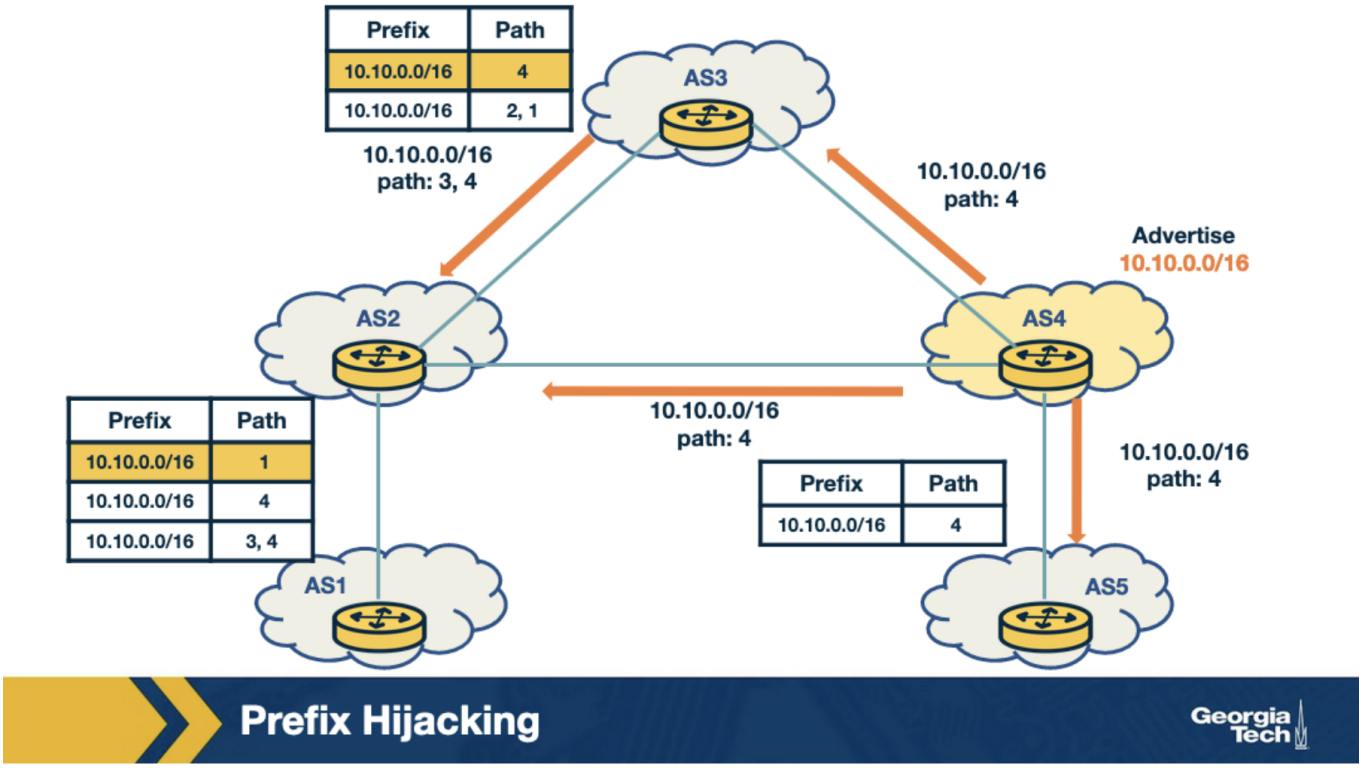
(BGP hijacking) What is the classification by AS-Path announcement?
Here a malicious AS will advertise that it either contains the destination or has a shorter path to the destination than it actually does. This is meant to trick neighbor ASes to route traffic through the bad AS.

(BGP hijacking) What is the classification by data plane traffic manipulation?
The malicious AS tries to hijack and manipulate the network traffic. It can blackhole packets, sniff packets in route (man in the middle attack) or impersonate the sender.

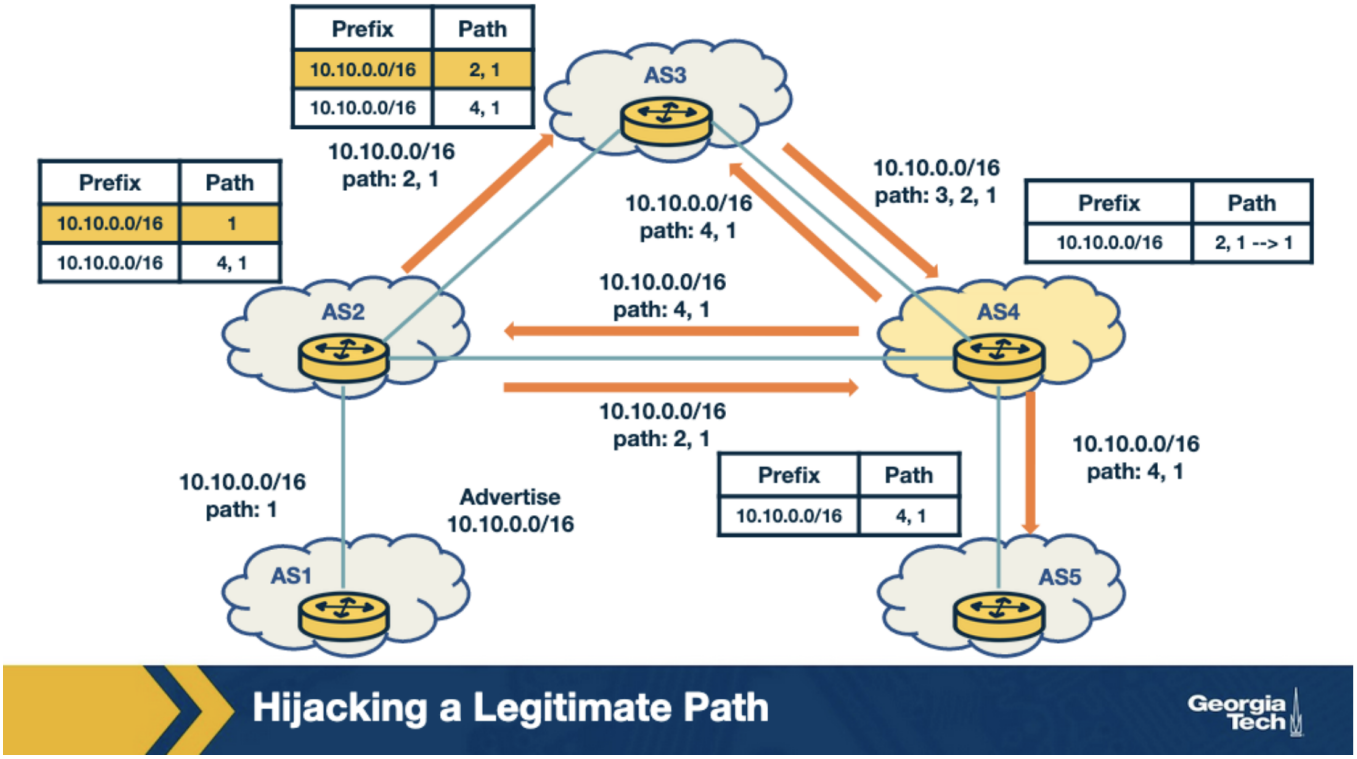
What are the causes or motivations behind BGP attacks?

- Human error / misconfiguration.
- Targeted attack or theft of information.
- High Impact Attack causing disruption of services.

Explain the scenario of prefix hijacking.



Explain the scenario of hijacking a path.



The key observation here is that the attacker does not need not to announce a new prefix, but rather it manipulates an advertisement before propagating it.

prefix

What are the key ideas behind ARTEMIS?

ARTEMIS is a system designed to allow an AS to safeguard its prefixes.

- **Configuration File:** This contains all prefixes owned by the network. It is populated by the network operator.
- **Mechanism for receiving BGP updates:** receiving updates from local routers and monitoring systems

The ARTEMIS system also allows the network operator to choose between a) accuracy and speed, and b) false negatives which are inconsequential (less impact on control plane) for less false positives.

What are the two automated techniques used by ARTEMIS to protect against BGP hijacking?

Prefix deaggregation: Broadcast out more specific prefixes than the attacker to get traffic back

Mitigation with Multiple Origin AS:

- The network notifies a Third parties of the hijacked prefix(es)
- The third party announces the prefixes from their location
- The traffic for the prefixes is attracted to the third party organization, which then scrubs it and tunnels it to the legitimate AS

What are two findings from ARTEMIS?

- Having one third party to do BGP announcements for you is highly effective (BGP announcements)
- Prefix filtering is much less effective than BGP announcement

Explain the structure of a DDoS attack.

A Distributed Denial of Service (DDoS) attack is an attempt to compromise a server or network resources with a flood of traffic. To achieve this, the attacker first compromises and deploys flooding servers (slaves).
Later, when initiating an attack, the attacker instructs these flooding servers to send a high volume of traffic to the victim. This results in the victim host either becoming unreachable or in exhaustion of its bandwidth.

What is spoofing, and how is it related to DDoS attacks?

IP spoofing is the act of setting a false IP address in the source field of a packet with the purpose of impersonating a legitimate server. In DDoS attacks, this can happen in two forms. In the first form, the source IP address is spoofed, resulting in the response of the server sent to some other client instead of the attacker's machine. This results in wastage of

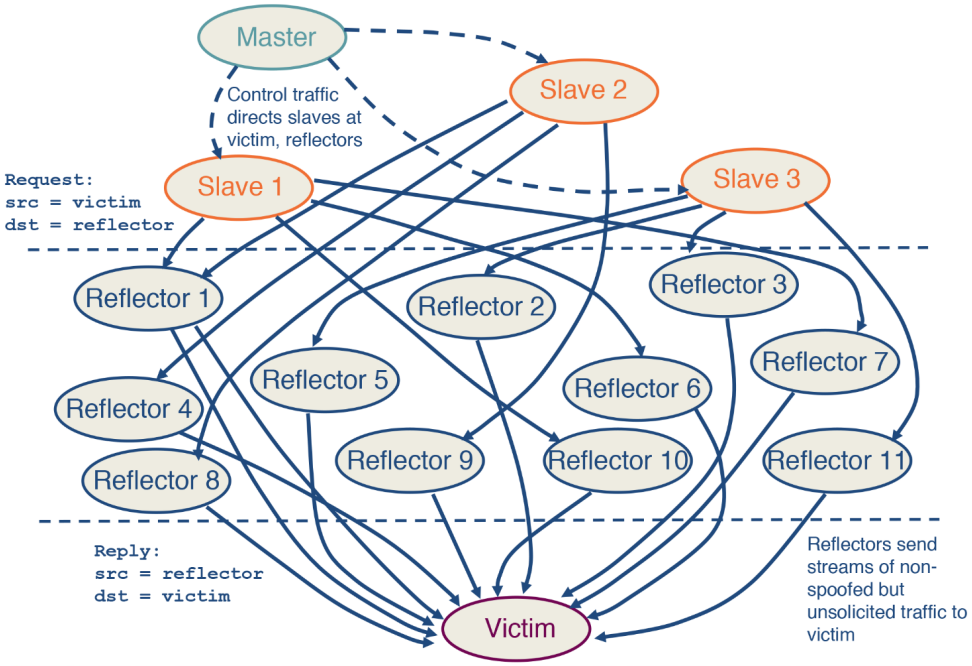
network resources and the client resources while also causing denial of

service to legitimate users. In the second type of attack, the attacker sets the same IP address in both the source and destination IP fields. This results in the server sending the replies to itself, causing it to crash.

Describe a Reflection and Amplification attack.

In a reflection attack, the attackers use a set of reflectors to initiate an attack on the victim. A reflector is any server that sends a response to a request. For example, any web server or a DNS server would return a SYN ACK in response to a SYN packet as part of TCP handshake. Other examples include query responses sent by a server or Host Unreachable responses to a particular IP.

Here, the master directs the slaves to send spoofed requests to a very large number of reflectors, usually in the range of 1 million. The slaves set the source address of the packets to the victim's IP address, thereby redirecting the response of the reflectors to the victim. Thus, the victim receives responses from millions of reflectors resulting in exhaustion of its bandwidth. In addition, the resources of the victim are wasted in processing these responses, making it unable to respond to legitimate requests. This forms the basis of a reflection attack. Let's consider the below figure.



Using Reflectors to Render a DDOS Attack Much More Diffuse

Georgia Tech

The master commands the three slaves to send spoofed requests to the reflectors, which in turn sends traffic to the victim. This is in contrast with the conventional DDoS attack we saw in the previous section, where the slaves directly send traffic to the victim. Note that the victim can easily identify the

reflectors from the response packets. However, the reflector cannot identify the slave sending the spoofed requests.

If the requests are chosen in such a way that the **reflectors send large responses** to the victim, it is a reflection and **amplification attack**. Not only would the victim receive traffic from millions of servers, the response sent would be large in size, making it further difficult for the victim to handle it.

What are the defenses against DDoS attacks?

Traffic Scrubbing Services

A scrubbing service diverts the incoming traffic to a specialized server, where the traffic is "scrubbed" into either clean or unwanted traffic. The clean traffic is then sent to its original destination. Although this method offers fine-grained filtering of the packets, there are monetary costs required for an in-time subscription, setup and other recurring costs. The other limitations include reduced effectiveness due to per packet processing and challenges in handling Tbps level attacks. There's also a possibility of decreased performance as the traffic may be rerouted and becoming susceptible to evasion attacks.

ACL Filters

Access Control List filters are deployed by ISPs or IXPs at their AS border routers to filter out unwanted traffic. These filters, whose implementation depends on the vendor-specific hardware, are effective when the hardware is homogeneous and the deployment of the filters can be automated. The drawbacks of these filters include limited scalability and since the filtering does not occur at the ingress points, it can exhaust the bandwidth to a neighboring AS.

BGP Flowspec

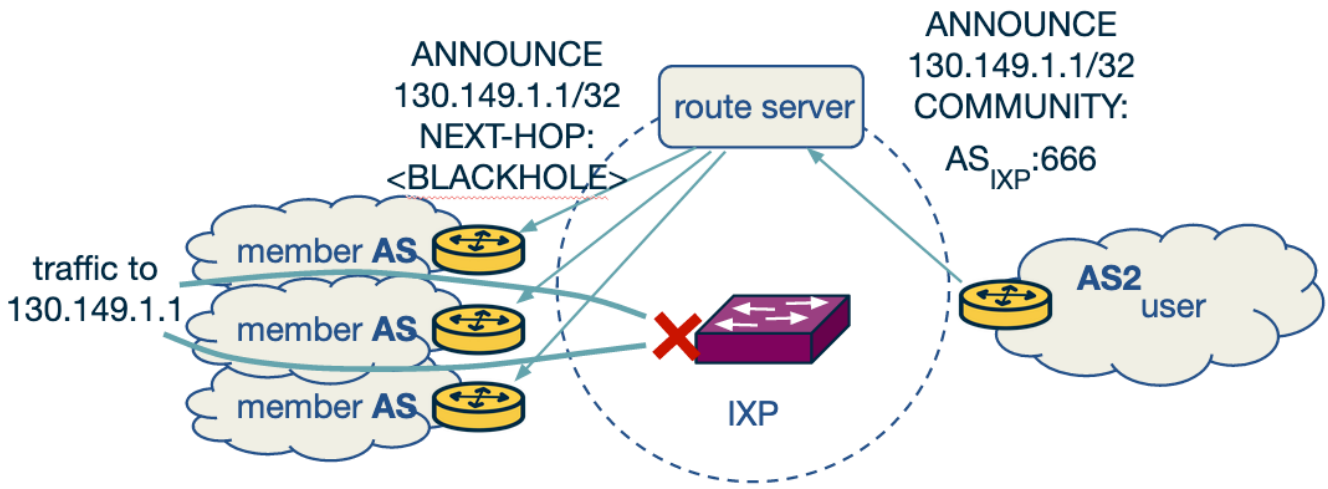
The flow specification feature of BGP, called Flowspec, helps to mitigate DDoS attacks by supporting the deployment and propagation of fine-grained filters across AS domain borders. It can be designed to match a specific flow or be based on packet attributes like length and fragment. It can also be based on the drop rate limit. Although flowspec has been effective in an intra-domain

environment, it is not so popular in inter-domain environments as it depends on trust and cooperation among competitive networks.

Explain provider-based blackholing.
The premise of this approach is that the traffic is stopped closer to the source of the attack and before it reaches the targeted victim. For a high volume attack, it proves to be an effective strategy when compared to other mitigation options.

This technique is implemented with the help of the upstream provider. The victim AS uses BGP to communicate the attacked destination prefix to its upstream AS, which then drops the attack traffic towards this prefix. Then either the provider will advertise a more specific prefix and modify the next-hop address that will divert the attack traffic to a null interface. The blackhole messages are tagged with a specific BGP blackhole community attribute, usually publicly available, to differentiate it from the regular routing updates.

Explain IXP blackholing.
In a similar manner, at IXPs, if the AS is a member of an IXP infrastructure and it is under attack, it sends the blackholing messages to the IXP route server when a member connects to the route server. The route server then announces the message to all the connected IXP member ASes, which then drops the traffic towards the blackholed prefix. The null interface to which the traffic should be sent is specified by the IXP. The blackholing message sent to the IXP should contain the IXP blackhole community.



(b) IXP blackholing

What is BGP Blackholing?

What is one of the major drawbacks of BGP blackholing?

One of the major drawbacks of BGP blackholing is that the destination under attack becomes unreachable since all the traffic including the legitimate traffic is dropped.

Generally, the mitigation is ineffective if a large number of upstream peers (or IXP peers) do not accept the blackholing announcements.

Lesson 10:Internet Surveillance and Censorship

What is DNS censorship?

DNS censorship is a large-scale network traffic filtering strategy opted by a network to enforce control and censorship over Internet infrastructure to suppress material which they deem as objectionable.

What are the properties of GFW (Great Firewall of China)?

1. **Locality of GFW nodes**: There are two differing notions on whether the GFW nodes are present only at the edge ISPs or whether they are also present in non-bordering Chinese ASes. The majority view is that *censorship nodes are present at the edge*.
2. **Centralized management**: Since the blocklists obtained from two distinct GFW locations are the same, there is a high possibility of a central management (GFW Manager) entity that orchestrates blocklists.
3. **Load balancing**: GFW load balances between processes based on source `and destination IP address. The processes are clustered together to collectively send injected DNS responses.

How does DNS injection work?

Basically a fake DNS A record response is sent back.

DNS injection is one of the most common censorship techniques employed by the GFW. The GFW uses a ruleset to determine when to inject DNS replies to censor network traffic.

What are the three steps involved in DNS injection?

1. DNS probe is sent to the open DNS resolvers
2. The probe is checked against the blocklist of domains and keywords
3. For domain level blocking, a fake DNS A record response is sent back. There are two levels of blocking domains: the first one is by directly blocking the domain, and the second one is by blocking it based on keywords present in the domain

List five DNS censorship techniques and briefly describe their working principles.

1. **Packet Dropping**: All network traffic going to a set of specific IP addresses is discarded. The censor identifies undesirable traffic and chooses to not properly forward any packets it sees associated with the traversing undesirable traffic instead of following a normal routing protocol.
    a. Strengths
        i. Easy to implement
        ii. Low cost
    b. Weaknesses
        i. Maintenance of blocklist - It is challenging to stay up to date with the list of IP addresses to block
        ii. Over blocking - If two websites share the same IP address and the intention is to only block one of them, there's a risk of blocking both
2. **DNS Poisoning**: When a DNS receives a query for resolving hostname to IP address - if there is no answer returned or an incorrect answer is sent to redirect or mislead the user request, this scenario is called DNS Poisoning.
    a. Strength
        i. No overblocking: Since there is an extra layer of hostname translation, access to specific hostnames can be blocked versus blanket IP address blocking.
    b. Weakness
        i. **Blocks the entire domain**. It is not possible to allow email contact while blocking the website.
3. **Content Inspection**:
    a. Proxy-based content inspection: This censorship technique is more sophisticated, in that it allows for **all network traffic to pass through a proxy where the traffic is examined for content**, and the proxy rejects requests that serve objectionable content.
        i. Strengths

1. Precise censorship: A very precise level of censorship can be achieved, down to the level of single web pages or even objects within the web page.
2. Flexible: Works well with hybrid security systems. E.g., with a combination of other censorship techniques like packet dropping and DNS poisoning.

ii. Weakness

1. Not scalable: They are expensive to implement on a large scale network as the processing overhead is large (through a proxy)

b. Intrusion detection system (IDS) based content inspection: An alternative approach is to use parts of an IDS to inspect network traffic. An IDS is easier and more cost effective to implement than a proxy based system as it is more responsive than reactive in nature, in that it informs the firewall rules for future censorship.

4. **Blocking with Resets**: The GFW employs this technique where it sends a TCP reset (RST) to block individual connections that contain requests with objectionable content.
    a. The RST packet is sent following a specific request (and not immediately after the handshake is complete).
    b. Content based.
5. **Immediate Reset of Connections**: Censorship systems like GFW have blocking rules in addition to inspecting content, to suspend traffic coming from a source immediately, for a short period of time.
    a. It achieves this by sending a RST request right after the handshake is complete, even before the client makes any request.
    b. Identity based.

Which DNS censorship technique is susceptible to overblocking?

Packet Dropping (IP filtering), because different domains might be associated with the same IP and we could be blocking several domains that share the same IP.

What are the strengths and weaknesses of "packet dropping" DNS censorship technique?

Strengths: Simple to implement, low cost
Weaknesses: Prone to overblocking, hard to maintain IP list to block

What are the strengths and weaknesses of "DNS poisoning" DNS censorship technique?
Strengths: no overblocking (blocks specific hostnames rather than blanket IP addresses)
Weaknesses: Blocks the entire domain, cannot block specific requests

What are the strengths and weaknesses of "content inspection" DNS censorship technique?
Strengths: Can be very targeted, Can filter domains or specific terms in the url
Weaknesses: Expensive to implement, has a impact on performance as it generates a bottleneck

What are the strengths and weaknesses of "blocking with resets" DNS censorship technique?
Strengths: Blocks individual connections that contain objectionable content
Weaknesses:

What are the strengths and weaknesses of "immediate reset of connections" DNS censorship technique?
Strengths: Blocks all traffic from a specific source immediately for a period of time
Weaknesses:

Our understanding of censorship around the world is relatively limited. Why is it the case? What are the challenges?

1. **Diverse Measurements**: We need widespread longitudinal measurements (spanning different geographic regions, ISPs, countries, and regions within

a country) to understand global Internet manipulation (organizations may implement censorship at multiple layers of the Internet protocol stack and using different techniques) and the heterogeneity of DNS manipulation, across countries, resolvers, and domains.

2. **Need for Scale**: There is a need for methods and tools that are independent of human intervention and participation. Relying on volunteers to measure Internet censorship is unlikely to reach the scale required.
3. **Identifying the intent to restrict content access**: While identifying inconsistent or anomalous DNS responses can help to detect a variety of underlying causes such as misconfigurations, identifying DNS manipulation is different and it requires that we detect the intent to block access to content. It poses its own challenges. So we need to rely on identifying multiple indications to infer DNS manipulation.
4. **Ethics and Minimizing Risks**: Obviously, there are risks associated with involving citizens in censorship measurement studies, based on how different countries may penalize access to censored material. Therefore, it is safer to stay away from using DNS resolvers or DNS forwarders in the home networks of individual users. Instead, it is safer to rely on open DNS resolvers that are hosted in Internet infrastructure, for example, within Internet service providers or cloud hosting providers.

What are the limitations of main censorship detection systems?

Global censorship measurement tools were created by efforts to measure censorship by running experiments from diverse vantage points. For example, CensMon used PlanetLab nodes in different countries. However, many such methods are no longer in use.

One of the most common systems/approaches is the OpenNet Initiative where volunteers perform measurements on their home networks at different times since the past decade. Relying on volunteer efforts makes continuous and diverse measurements very difficult.

In addition, Augur is a new system created to perform longitudinal global measurements using TCP/IP side channels. However, this system focuses on identifying IP-based disruptions as opposed to DNS-based manipulations.

What kind of disruptions does Augur focus on identifying?
Augur performs longitudinal global measurements using TCP/IP side channels. The system focuses on identifying IP-based disruptions as opposed to DNS-based manipulations.

How does Iris counter the issue of lack of diversity while studying DNS manipulation? What are the steps associated with the proposed process?
In order to counter the lack of diversity, Iris uses a few thousand open DNS resolvers located all over the globe, that are part of the Internet infrastructure. There are two main steps associated with this process:

1. Scanning the Internet's IPv4 space for open DNS resolvers
2. Identifying Infrastructure DNS Resolvers

What are the steps involved in the global measurement process using DNS resolvers?
The steps involved in this measurement process are:

1. **Performing global DNS queries** – Iris queries thousands of domains across thousands of open DNS resolvers.
2. **Annotating DNS responses with auxiliary information** – To enable the classification, Iris annotates the IP addresses with additional information such as their geo-location, AS, port 80 HTTP responses, etc. This information is available from the Censys dataset.
3. **Additional PTR and TLS scanning** – One IP address could host several websites via virtual hosting. So, when Censys retrieves certificates from port 443, it could differ from one retrieved via TLS's Server Name Indication (SNI) extension. This results in discrepancies that could cause Iris to label virtual hosting as DNS inconsistencies. To avoid this, Iris adds PTR and SNI certificates.

What metrics does Iris use to identify DNS manipulation once data annotation is complete? Describe the metrics. Under what condition, do we declare the response as being manipulated?
After annotating the dataset, techniques are performed to clean the dataset, and identify whether DNS manipulation is taking place or not. Iris uses two types of metrics to identify this manipulation:

1. **Consistency Metrics**: Domain access should have some consistency, in terms of network properties, infrastructure or content, even when accessed from different global vantage points. Using one of the domains Iris controls gives a set of high-confidence consistency baselines. Some consistency metrics used are IP address, Autonomous System, HTTP Content, HTTPS Certificate, PTRs for CDN.

2. **Independent Verifiability Metrics**: In addition to the consistency metrics, they also use metrics that could be externally verified using external data sources. Some of the independent verifiability metrics used are: HTTPS certificate (whether the IP address presents a valid, browser trusted certificate for the correct domain name when queried without SNI) and HTTPS Certificate with SNI.

How to identify DNS manipulation via machine learning with Iris?
If **any** consistency metric or independent verifiability metric is satisfied, the response is correct. Otherwise, the response is classified as manipulated.

How is it possible to achieve connectivity disruption using routing disruption approach?
The highest level of Internet censorship is to completely block access to the Internet. A more subtle approach is to use software to interrupt the routing or packet forwarding mechanisms.

Routing disruption: A routing mechanism decides which part of the network can be reachable. Routers use **BGP** to communicate updates to other routers in the

network. The routers share which destinations it can reach and continuously update its forwarding tables to select the best path for an incoming packet.
If this communication is disrupted or disabled on critical routers, it could result in unreachability of the large parts of a network.
Using this approach can be easily detectable, as it involves withdrawing previously advertised prefixes must be withdrawn or re-advertising them with different properties and therefore modifying the global routing state of the network, which is the control plane.

How is it possible to achieve connectivity disruption using packet filtering approach?

Packet filtering: It is typically used as a security mechanism in firewalls and switches. But to disrupt a network's connectivity, packet filtering can be used to block packets matching a certain criteria disrupting the normal forwarding action. This approach can be harder to detect and might require active probing of the forwarding path or monitoring traffic of the impacted network.

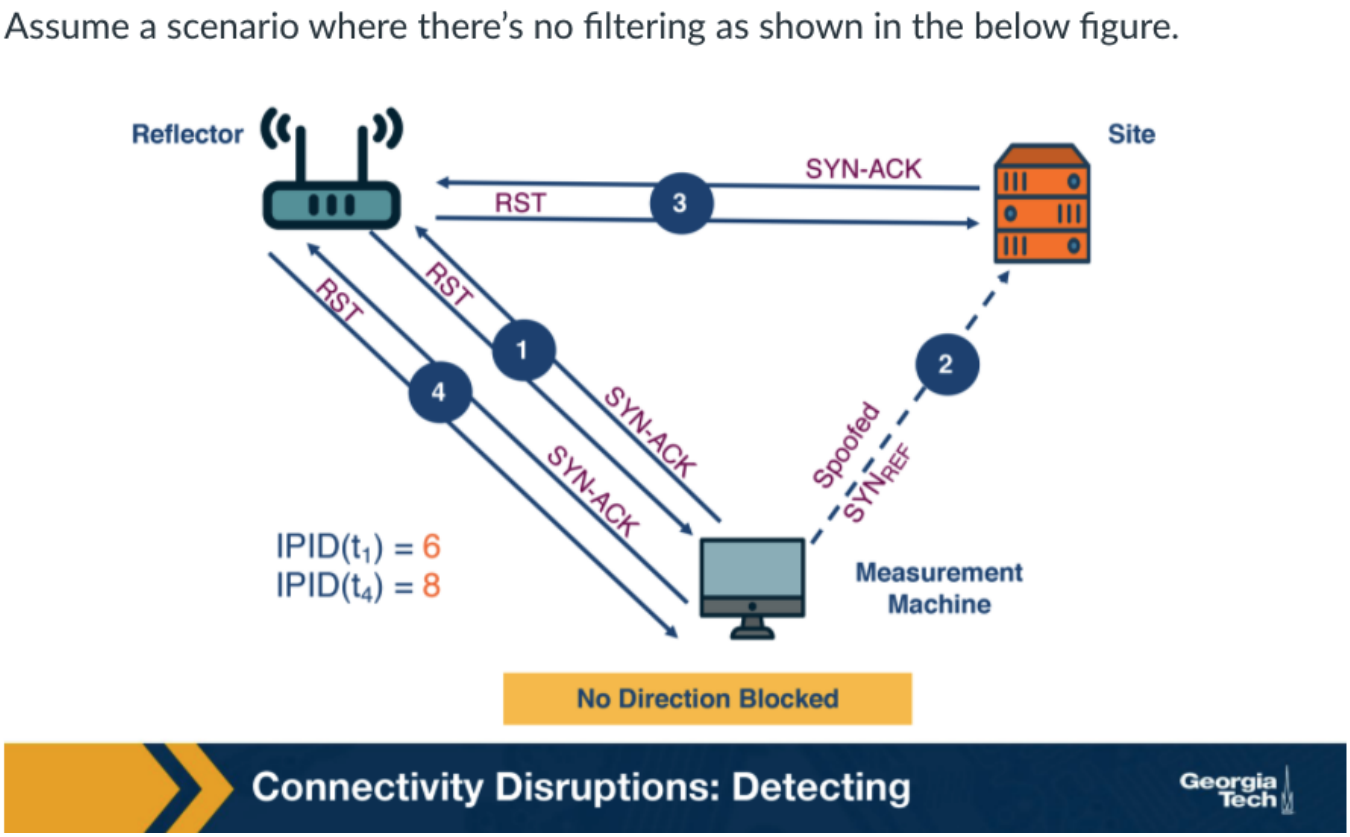
Explain a scenario of connectivity disruption detection in case when no filtering occurs.

Augur uses a measurement machine to detect filtering between hosts.
The measurement machine keeps track of the **IP ID** of the reflector, which normally uses a monotonically increasing global counter, to determine if and how many packets are generated by a host.

Probing is used to monitor the IP ID of a host over time, it consists in the measurement machine sending a TCP SYN-ACK packet to the reflector and receiving a TCP RST packet which contains the IP ID.

Perturbation is used to try and force the reflector host to generate a response packet. It is achieved by sending a spoofed (with the reflector's IP address) TCP SYN packet to the site host, which in turn sends a TCP SYN-ACK response packet to the reflector. If the reflector receives the packet, it will send a TCP RST response back to the site.

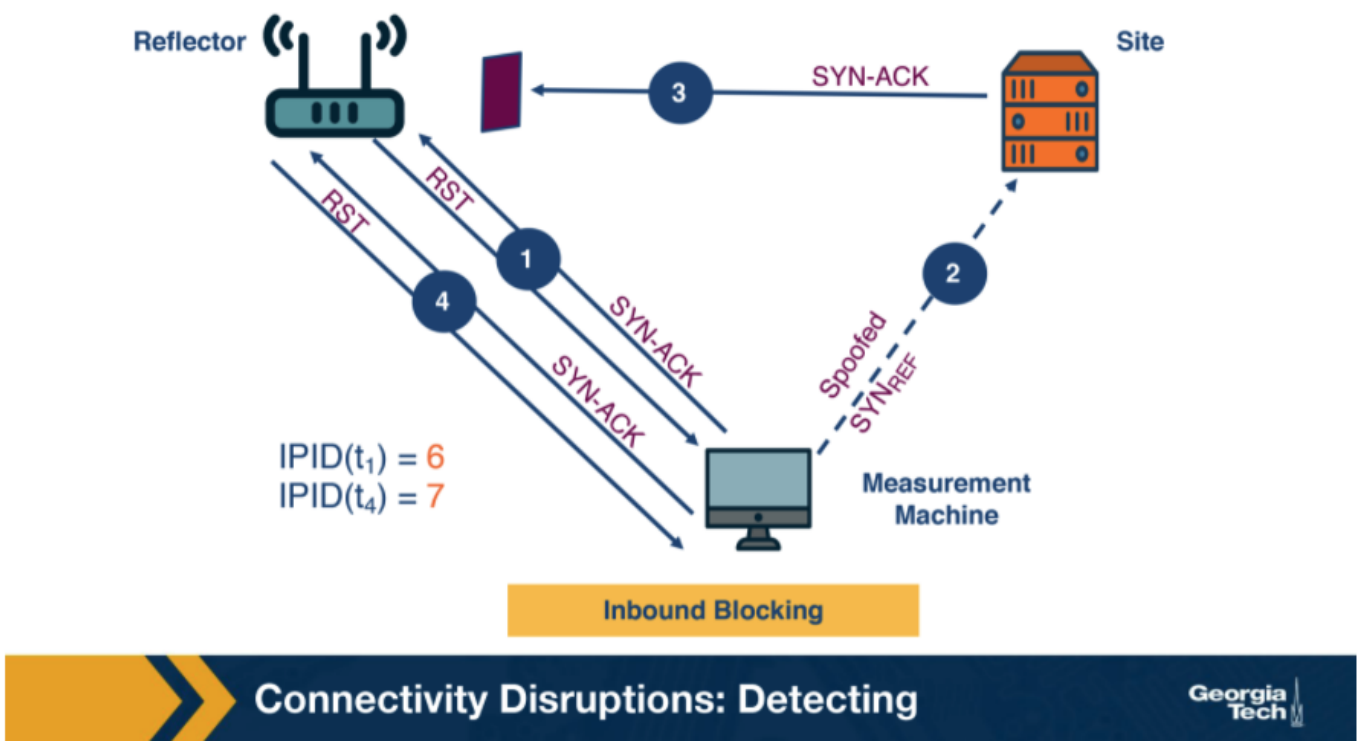
If the connection is **not blocked** in any way, when the measurement machine probes the reflector again, it should see a **difference of 2 between the IP IDs.**



Explain a scenario of connectivity disruption detection in case of the inbound blocking.

If the path between the site and the reflector is blocked, this is called **Inbound Blocking**. We know this is happening because the reflector never received the TCP SYN-ACK response from the site and doesn't increase its IP ID.

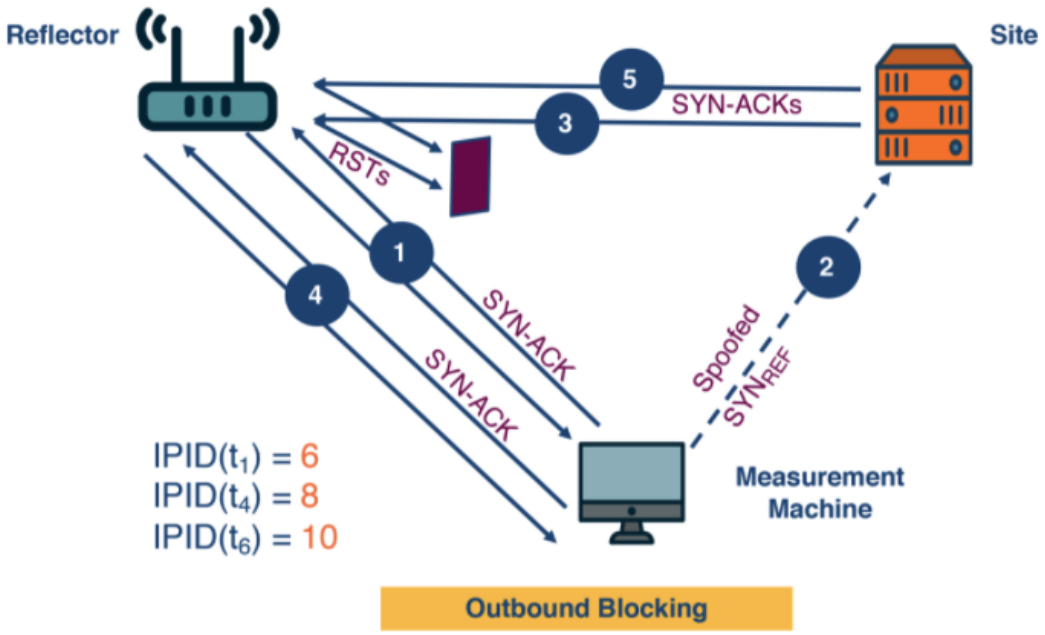
So when the measurement machine probes the reflector, **the IP ID only increases by 1**.



Explain a scenario of connectivity disruption detection in case of the outbound blocking.

Finally, if the path between the reflector and the site is blocked, it is called **Outbound blocking**. We know this happens because the reflector receives the TCP SYN-ACK request coming from the site, but the reflector's TCP RST response never reaches the site, so the site will keep sending the TCP SYN-ACK response periodically (depending on its configuration).

When the measurement machine probes the reflector, **the IP ID has increased more than 2**.



Connectivity Disruptions: Detecting

Georgia Tech

Lesson 11: Applications (Video)

Compare the bitrate for video, photos, and audio.
Video has the highest bitrate (2 Mbps) by a large margin, while audio (128 kbps) has the lowest bitrate. Photos (320 kbps) are in between video and audio.

What are the characteristics of streaming stored video?
It's **interactive** (users can pause, rewind or fast forward), and should play **continuously**. Video is usually stored in a CDN and can be shared using client + server model or p2p.

What are the characteristics of streaming live audio and video?
These are more **delay sensitive** than stored video, but not as delay sensitive as conversation voice and video. Live video and audio streams tend to have lots of **simultaneous users**. Generally, up to 10 seconds delay is ok. Not interactive.

What are the characteristics of conversational voice and video over IP?
Conversational streaming usually involves 2 or more clients and is **highly delay sensitive**. Delays under 150ms are unnoticeable, and delays over 400ms can be frustrating to users. Conversational applications are **loss tolerant** as long as the lost information is not too concentrated.

How does the encoding of analog audio work (in simple terms)?
Analog is a continuous wave. For digital, thousands of samples are taken per second and then rounded to some discrete value to best approximate the analog wave. This is known as **quantization**.

What are the three major categories of VoIP encoding schemes?
narrowband, broadband, and multimode.

What are the functions that signaling protocols are responsible for?
1) User location - Identifying where the recipient/callee of the signal is
2) Session establishment - for recipient/callee (handling acceptance / rejection / redirection)

3) Session negotiation (synchronizing participants on some set of session properties)
4) Call participation management (handling endpoints joining or leaving an existing session)

What are three QoS VoIP metrics?

1. End to end delay
2. Jitter
3. Packet Loss

What kind of delays are included in "end-to-end delay"?

- encoding
- Converting data into packets
- Network delay
- Playback delay (from recipient's buffer)
- decoding

How does "delay jitter" occur?

Jitter occurs because different packets experience different levels of delay.

What are the mitigation techniques for delay jitter?

- Dropping packets that are too old
- Using a "jitter buffer", which reduces the number of packets dropped but increases end to end delay

Compare the three major methods for dealing with packet loss in VoIP protocols.

See below

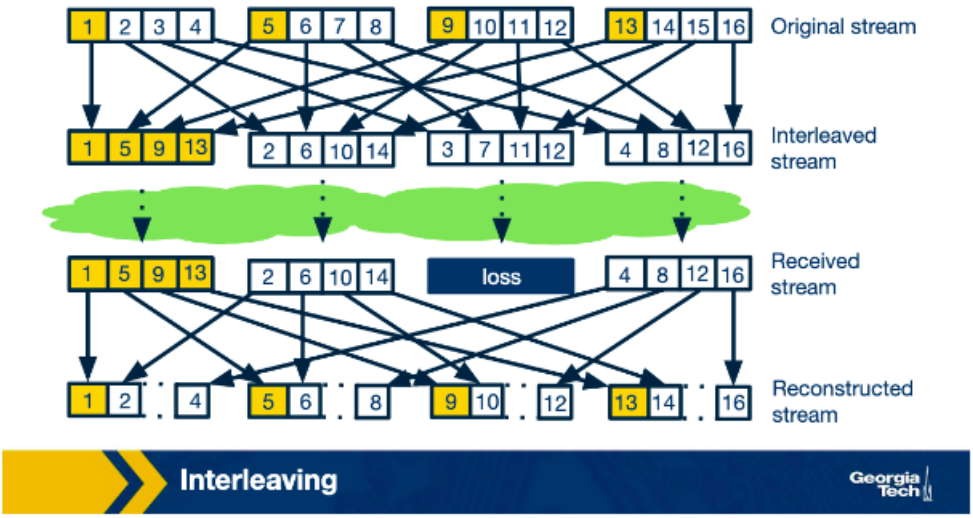
How does FEC (Forward Error Correction) deal with the packet loss in VoIP?

Redundant data is transmitted to the recipient. There a XOR is applied to the data which essentially fills the holes created by the lost packet. Redundant data is usually lower quality than the original.

What are the tradeoffs of FEC?

The biggest trade off is the increased bandwidth consumption. The receiver also needs a larger buffer to handle the duplicate data, which leads to playback delay.

How does interleaving deal with the packet loss in VoIP/streaming stored audio? What are the tradeoffs of interleaving?
Interleaving spreads playback information across multiple chunks. This way, if one chunk is lost, there is still enough information in the other chunks to reconstruct the output. The idea is that many smaller audio gaps are preferable to one large audio gap.
The receiving side has to wait longer to receive consecutive chunks of audio, and that increases latency.



How does error concealment technique deal with the packet loss in VoIP?
By replacing a lost packet with either a copy of a prior packet or an approximated packet (using interpolation with the packets before and after the lost packet)

What developments lead to the popularity of consuming media content over the Internet?

- Increase in bandwidth of network core and last mile access links
- Better video compression
- Digital Rights management culture has encourages content providers to put content on the Internet

Provide a high-level overview of adaptive video streaming.
Video is available in multiple bitrates. The client will select a bitrate for each fetch depending on the network conditions

(Optional) What are two ways to achieve efficient video compression?

- within an image (spatial redundancy) -- pixels that are nearby in a picture tend to be similar.
- across images (temporal redundancy) -- in a continuous scene, consecutive pictures are similar.

(Optional) What are the four steps of JPEG compression?

1. Decompose image into 3 matrices Y, Cb and Cr (color component (chrominance or Cb, Cr) and brightness component (luminance or Y)) from RGB
2. Divide the image into sub-images, apply Discrete Cosine Transformation to transform it into the frequency domain.
3. Compress the matrix of the coefficients using a predefined Quantization table
4. Lossless encoding to store the coefficients.

(Optional) Explain video compression and temporal redundancy using I-, B-, and P-frames.

(Optional) Why is video compression unable to use P-frames all the time?

(Optional) What is the difference between constant bitrate encoding and variable bitrate encoding (CBR vs VBR)?

Which protocol is preferred for video content delivery -UDP or TCP? Why?
TCP. Reliability and congestion control are the 2 factors that lead to TCP winning out. Reliability is important because lost data will likely lead to decoding failure. Congestion control is key to preventing re-buffering

What was the original vision of the application-level protocol for video content delivery and why was HTTP chosen eventually?
Specialized stateful servers (intelligence on the server). HTTP was used because of existing CDN infrastructure, which uses HTTP. It also made bypassing middleboxes and firewalls easier since they already know HTTP.

Summarize how progressive download works.
Requests (HTTP GET) are made for content of varying bit rates (depending on network intelligence). Servers send over information as fast as possible.

Progressive download starts in a “Buffer filling” state where it continuously makes requests until the buffer is full and a “Steady state”, where requests are made as space becomes available.

How to handle network and user device diversity?
Use multiple bitrates to accommodate varying network conditions and screen sizes. The client can then pick the best bitrate.
At the beginning of every video session, the client first downloads a manifest file that contains all the metadata information about the video content (ex. bitrates) and the associated URLs.

How does the bitrate adaptation work in DASH (Dynamic Streaming over HTTP)?
A video is divided into chunks that can be downloaded at different bitrates. The client adapts the bitrate based on its estimation of network conditions.

What are the goals of bitrate adaptation?
The goal of a good bitrate adaptation algorithm then is to maximize the overall user Quality of experience (QoE).
A good quality of experience (QoE) is usually characterized by:

1. Low or zero rebuffering
2. High quality video
3. Low variation in video quality
4. Low startup latency

What are the different signals that can serve as an input to a bitrate adaptation algorithm?

Network Throughput: the speed of the network. The bitrate should be less than the network throughput

Video Buffer: The amount of video in the buffer. If the buffer is almost full, the client can afford a longer wait for a higher bitrate video. If it's almost empty, the client must go with a lower bitrate to avoid rebuffering.

Explain buffer-filling rate and buffer-depletion rate calculation.

The **buffer fill rate** is the network bandwidth divided by the chunk bitrate. It's the number of chunks that can be downloaded per second. If each chunk is 1 second of video, then it's the number of seconds of video that can be downloaded in 1 second.

The **buffer depletion rate** is the number of chunks consumed in 1 second of playback. If each chunk is 1 second, this is simply 1

What steps does a simple rate-based adaptation algorithm perform?

Estimation: The algorithm looks at previous chunks that were downloaded and performs a smoothing filter (moving average or harmonic mean) to estimate the future bandwidth.

Quantization: select the maximum bitrate that is smaller than the output from the estimation step. There is adjusted using a factor for the following reasons:

- A more conservative bitrate is less likely to cause rebuffering
- The selected bitrate may be higher than the maximum offered
- Account for transport layer overhead

Explain the problem of bandwidth over-estimation with rate-based adaptation.

The bandwidth could drop significantly. If a client is downloading at a high bitrate, the drop in bandwidth could make the next chunk take a long time to download. If the time to download is longer than the buffer size, then re-buffering will occur

Explain The problem of bandwidth under-estimation with rate-based adaptation.

TCP converges to an equal sharing of network resources when all connections are taking as much bandwidth as possible. The on/off behavior of DASH leads to some clients having a lower share. Rate-based adaptation makes this worse because selecting smaller bit rates can lead to further reduction in the bandwidth share.

Lesson 12: Applications (CDNs and Overlay Networks)

What is the drawback to using the traditional approach of having a single, publicly accessible web server?
What is a CDN?
What are the six major challenges that Internet applications face?
What are the major shifts that have impacted the evolution of the Internet ecosystem?
Compare the“enter deep” and “bring home” approach of CDN server placement.
What is the role of DNS in the way CDN operates?
What are the two main steps in CDN server selection?
What is the simplest approach to select a cluster? What are the limitations of this approach?
What metrics could be considered when using measurements to select a cluster?
How are the metrics for cluster selection obtained?
Explain the distributed system that uses a 2-layered system. What are the challenges of this system?
What are the strategies for server selection? What are the limitations of these strategies?
What is consistent hashing? How does it work?
Why would a centralized design with a single DNS server not work?
What are the main steps that a host takes to use DNS?
What are the services offered by DNS, apart from hostname resolution?
What is the structure of DNS hierarchy?Why does DNS use a hierarchical scheme?
What is the difference between iterative and recursive DNS queries?
What is DNS caching?
What is a DNS resource record?
What are the most common types of resource records?
Describe the DNS message format.
What is IP Anycast?
What is HTTP Redirection?

GLOSSARY

Lesson 1

DARPA – Defense Advanced Research Projects Agency
NCP – Network Control Protocol
ARPANET – Advanced Research Projects Agency Network
TCP – Transmission Control Protocol
UDP – User Datagram Protocol
DNS – Domain Name System
API – Application Programming Interfaces
ISO – International Organization for Standardization
OSI – Open Systems Interconnection
HTTP – Hyper Text Transfer Protocol
SMTP – Simple Mail Transfer Protocol
FTP – File Transfer Protocol
PPP – Point to Point Protocol
NAT – Network Address Translation
STUN – Session Traversal Utilities for NAT
IPX – Internetwork Packet Exchange
AIP – Accountable Internet Protocol
MAC address – Media Access Control address
LAN – Local Area Network

Lesson 2

DCTCP – Data Center TCP
RTT – Round Trip Time
ACK – Acknowledgment
SYN/ACK – Synchronize Acknowledge
ARQ – Automatic Repeat Request
ICMP – Internet Control Message Protocol

ECN – Explicit Congestion Notification
QCN – Quantized Congestion Notification
TTL – Time To Live
AIMD – Additive Increase Multiplicative Decrease
MSS – Maximum Segment Size

Lesson 3

RIP – Routing Information Protocol
OSPF – Open Shortest Path First
BGP – Border Gateway Protocol
IGP – Interior Gateway Protocol
DV – Distance Vector
AS – Autonomous System
LSA – Link Statement Advertisements
FIB – Forwarding Information Base
SPF – Shortest Path First
SNMP – Simple Network Management Protocol
MIB – Management Information Bases

Lesson 4

ISP – Internet Service Provider
IXP – Internet Exchange Points
CDN – Content Delivery Networks
eBGP – external BGP
iBGP – internal BGP
MED – Multi-Exit Discriminator
DDoS Attack – Distributed Denial-of-Service Attack
ccTLD – country-code Top-Level Domain
NTP – Network Time Protocol

RS – Route Server
RIB – Routing Information Base
VP – Vantage Point

Lesson 5

SDN – Software Defined Networking
FIB – Forwarding Information Base
FIFO – First-In-First-Out
CIDR – Classless Internet Domain Routing

Lesson 6

MPLS – Multiprotocol Label Switching
HOL – Head-of-line
DRR – Deficit Round Robin

Lesson 7

IDSs – Intrusion Detection Systems
API – Application Programming Interfaces
IETF – Internet Engineering Task Force
ATM – Asynchronous Transfer Mode
NFV – Network Function Virtualization
REST – Representational State Transfer
ONOS – Open Network Operating System
MD-SAL – Model Driven Service Abstraction Layer
CRUD – Create, Read, Update, Delete

Lesson 8

ForCES – Forwarding and Control Element Separation
OVSDB – Open vSwitch Database Management Protocol
POF – Protocol Oblivious Forwarding
VLAN – Virtual Local Area Network
MLPS – Multiprotocol Label Switching
VxLAN – Virtual Extensible Local Area Network
NVGRE – Network Virtualization using Generic Routing Encapsulation
NVP – Network Virtualization Platform
NOS – Network Operating System
NDMs – Negotiable Datapath Models
WAN – Wide Area Network
P4 – Programming Protocol-independent Packet Processors
NPU – Network Processor
TDGs – Table Dependency Graphs
LVAP – Light Virtual Access Points
OF-RHM – OpenFlow Random Host Mutation
SDX – Software Defined Everything

Lesson 9

RRDNS – DNS Round Robin
MOAS – Multiple Origin AS
FFSN – Fast-Flux Service Networks
FIRE – Finding Rogue Networks
C&C – Command and Control
TLS Handshake – Transport Layer Security Handshake
RF – Random Forest
SVM – Support Vector Machine
NANOG – North American Network Operators' Group

ACL – Access Control List

Lesson 10

GFW – Great Firewall of China
IDS – Intrusion Detection System
PTR – Pointer Record
TLS – Transport Layer Security
SNI – Server Name Indication
RIRs – Regional Internet Registries
RIS – Routing Information Service
IBR – Internet Background Radiation

Lesson 11

VoIP – Voice over IP
PCM – Pulse Code Modulation
SIP – Session Initiation Protocol
FEC – Forward Error Correction
XOR – exclusive OR
MPEG – Moving Picture Experts Group
JPEG – Joint Photographics Experts Group
DRM – Digital Rights Management
RGB – Red Green Blue
GoP – Group of Pictures
VBR – Variable bitrate
CBR – Constant bitrate
DASH – Dynamic Streaming over HTTP

Lesson 12

LDNS – Local DNS server
TLD Servers – Top Level Domain Servers