18CSE453T Network Routing Algorithms

Unit II

Unit II - Content

Session 1 : *Basic Forwarding Functions, Routing table versus forwarding table*

Session 2 : Types of router, Elements of Router

Session 3 : Packet Flow, Packet Processing

Session 4 : *Shared CPU architecture, Shared forwarding Engine Architecture, Shared Nothing Architectures, Clustered Architectures*

Session 5 : *Impact of Addressing on lookup, Longest Prefix Matching*

Session 6 : *Naïve Algorithms, Binary Tries*

Session 7 : Multi-bit Tries, Compressing multi-bit strides

Session 8 : Search By Length Algorithms, Search By value approaches

Session 9 : Hardware Algorithms, Comparing Different Approaches

Session 1

Basic Forwarding Functions,
Routing table versus forwarding table

Router Architectures

- Traditionally, routers have been implemented purely with software running on a general purpose personal computer (PC) with a number of interfaces.
- Such a device can receive packets on one of its interfaces, perform routing functions, and send packets out on another of its interfaces.
- As the Internet grew over the years, the type and size of routers changed, since routers based on general-purpose PC architectures are limited by the performance of the central processor and memory.
- Fortunately, advances in silicon technology have made it possible to build hardware-based routers capable of handling high data rates.

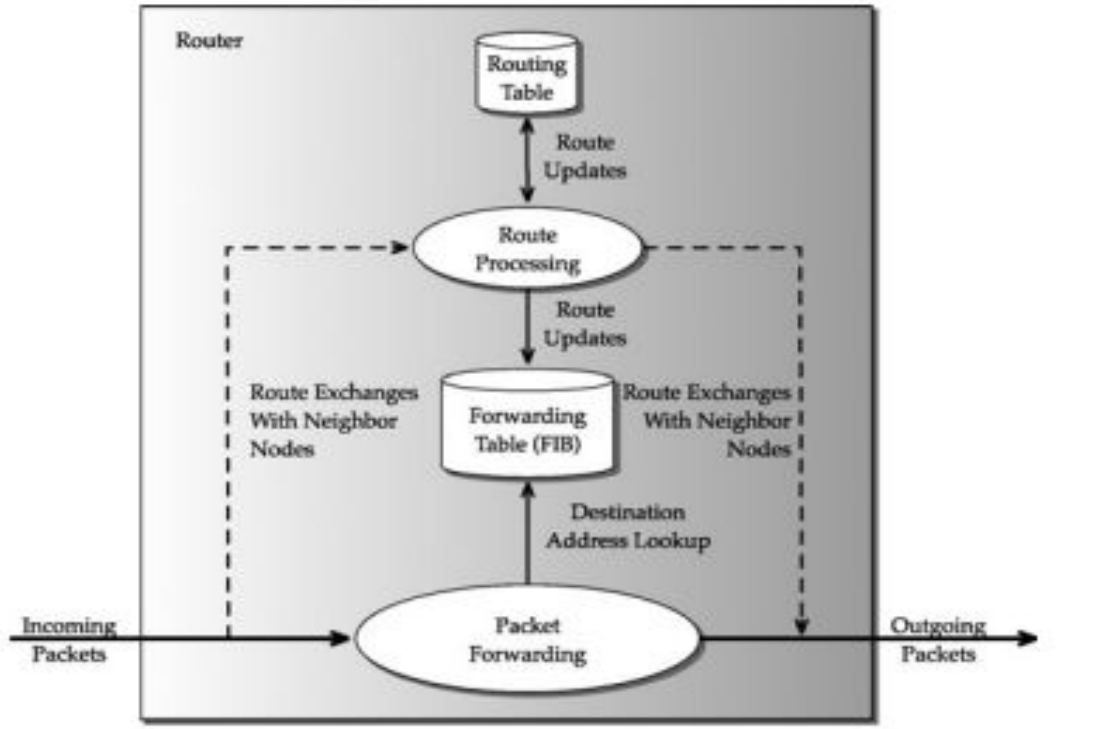
Functions of a Router

- Broadly speaking, a router must perform two fundamental tasks: routing and packet forwarding.
- Based on the information exchanged between neighboring routers using routing protocols, the routing process constructs a view of the network topology and computes the best paths.
- The network topology reflects network destinations that can be reached as identified through IP prefix-based network address blocks.
- The best paths are stored in a data structure called a forwarding table. The packet forwarding process moves a packet from an input interface (“ingress”) of a router to the appropriate output interface (“egress”) based on the information contained in the forwarding table.
- Since each packet arriving at the router needs to be forwarded, the performance of the forwarding process determines the overall performance of the router.

Functions of a Router – contd.,

- The functions of the packet forwarding process can be categorized into two subgroups: basic forwarding and complex forwarding.
- Basic forwarding defines the minimal set of functions a router should implement in order to transfer packets between interfaces.
- Complex forwarding functions represent the additional processing required by the routers, depending on their deployment environments and their usage.

Router Architecture



Router architecture: a functional view.

Basic Forwarding Functions

- For forwarding an IP packet from an incoming interface to an outgoing interface, a router needs to implement the following basic forwarding functions
- IP Header Validation:
  - Every IP packet arriving at a router needs to be validated. Such a test ensures that only well-formed packets are processed further while the rest are discarded.
  - This test also ensures that the version number of the protocol is correct, the header length of the packet is valid, and the computed header checksum of the packet is same as the value of the checksum field in the packet header.
- Packet Lifetime Control:
  - Routers must decrement the time-to-live (TTL) field in the IP packet header to prevent packets from getting caught in the routing loops forever.
  - If the TTL value is zero or negative, the packet is discarded; an ICMP message is generated and sent to the original sender.
- Checksum Recalculation:
  - Since the value of the TTL is modified, the header checksum needs to be updated.
  - Instead of computing the entire header checksum again, it is more efficient to compute it incrementally; after all, the TTL value is always decremented by 1.

Basic Forwarding Functions

- Route Lookup:
  - The destination address of the packet is used to search the forwarding table for determining the output port.
  - The result of this search will indicate whether the packet is destined for the router or to an output port (unicast) or to a set of multiple output ports (multicast).
- Fragmentation:
  - It is possible that the maximum transmission unit (MTU) of the outgoing link is smaller than the size of the packet that needs to be transmitted.
  - This means that the packet would need to be split into multiple fragments before transmission.
- Handling IP Options:
  - The presence of the IP options field indicates that there are special processing needs for the packet at the router.
  - While such packets might arrive infrequently, a router nonetheless needs to support those processing needs.

Complex Forwarding Functions

- Besides the basic functions, complex issues such as security, different user requirements, and service guarantees based on different service level agreements have become paramount and need to be addressed.
- These issues translate to additional processing when forwarding a packet, without essentially increasing overall packet processing time at a router.
- To cite an example of service differentiation, consider a scenario where customers are interested in watching a high-definition movie streaming directly over the Internet.
- Such a streaming requires not only high bandwidth but timely delivery of the data. The router needs to distinguish such packets so that it can forward them earlier

Complex Forwarding Functions

- This results in the notion of differentiated services, and consequently requires that routers support a variety of mechanisms such as the following:
- Packet Classification:
  - For distinguishing packets, a router might need to examine not only the destination IP address but also other fields such as source address, destination port, and source port.
  - The process of differentiating the packets and applying the necessary actions according to certain rules is known as packet classification.
- Packet Translation:
  - As the public IPv4 address space is being exhausted, there is a need to map several hosts to a single public address. Thus, a router that acts as a gateway to a network needs to support network address translation (NAT).
  - NAT maps a public IP address into a set of private IP addresses and vice versa.
  - This requires a router to maintain a list of connected hosts and their local addresses and to translate the incoming and outgoing packets.
- Traffic Prioritization:
  - A router might need to guarantee a certain quality of service to meet service level agreements.
  - This involves applying different priorities to different customers or data flows and providing a level of performance in accordance with the predetermined service agreements.
  - For example, the agreement might specify that a fixed number of packets must be delivered at a constant rate, which is necessary for real-time streaming multimedia applications such as IPTV, or real-time interactive applications such as VoIP.

Routing Process Functions

- Besides packet forwarding, i.e., the data plane function, a router needs to ensure that the contents of the forwarding table reflect the current network topology.
- For this, a router also needs to provide control plane and management plane functions.
- In particular, a router needs to handle:
  - Routing Protocols
  - System Configuration
  - Router Management

Routing Process Functions

- Routing Protocols:
  - Routers need to implement different routing protocols, such as OSPF, BGP, and RIP for maintaining peer relationships by sending and receiving route updates from adjacent routers. These route updates are sent and received as normal IP packets.
  - Once the updates are received, the forwarding table is modified so that subsequent packets are forwarded to the correct outgoing links.
- System Configuration:
  - Network operators need to configure various administrative tasks such as configuring of interfaces, routing protocol keep alives, rules for classifying packets.
  - Hence, a router needs to implement various functions for adding, modifying and deleting these configuration data, as well as persistently storing them for retrieval later
- Router Management:
  - In addition to the configuration tasks, the router needs to be monitored for continuous operation. These functions include supporting various management functions that are implemented using protocols such as simple network management protocol (SNMP).

Routing Table versus Forwarding Table

Routing Table	Forwarding Table
The routing table is constructed by the routing algorithms based on the information exchanged between neighboring routers by the routing protocols.	The forwarding table, on the other hand, is consulted by the router to determine the output interface an incoming packet needs to be forwarded.
Each entry in the routing table maps an IP prefix to a next hop.	Each entry in the forwarding table maps an IP prefix to an outgoing interface
The routing tables are usually implemented in software	Forwarding table is implemented in a specialized hardware for high-speed routers.
The routing table indicates the next-hop IP address for a destination IP prefix.	The forwarding table tells us a packet bound to the network identified by the IP prefix should be forwarded to interface eth0 with the appropriate MAC address.

Routing table

IP prefix	Next hop
10.5.0.0/16	192.168.5.254

Forwarding table

IP prefix	Interface	MAC address
10.5.0.0/16	eth0	00:0F:1F:CC:F3:06

Session 2

Types of Router, Elements of Router

Types of Routers

- Routers can be of different complexity based on where in the network they are deployed and how much traffic they need to sustain.
- Three types of routers:
  - Core Routers
  - Edge Routers
  - Enterprise Routers

CORE ROUTERS

- Core routers are used by service providers for interconnecting a few thousand small networks so that the cost of moving traffic is shared among a large customer base.
- Since the traffic arriving at the core router is highly aggregated, it should be capable of handling large amounts of traffic
- Hence, the primary requirements for a core router are high speed and reliability.
- With the increase in the number of systems connected to the Internet and the associated surge in traffic growth, demand is placed on core routers to forward more packets per second.
- Hence, specialized algorithms implemented in hardware are required for fast and efficient lookups.
- The reliability of a router depends on the reliability of physical elements such as the line cards, switch fabric, and route control processor cards.
- The reliability of these physical elements is achieved by full redundancy—dual power supplies, standby switch fabric, and duplicate line cards and route control processor cards.

EDGE ROUTERS

- Edge routers, also known as access routers, are deployed at the edge of the service provider networks for providing connectivity to customers from home and small businesses.
- The edge routers need to support an aggregation of customers using different access technologies.
- In addition to legacy remote access protocols, these routers need to implement newer protocols such as point-to-point tunneling protocol (PPTP), point-to-point protocol over Ethernet (PPPoE), and IPsec that support VPNs.
- The edge routers support a large number of ports capable of different access technologies and many protocols operating at each port.

ENTERPRISE ROUTERS

- Enterprise networks interconnect end systems located in companies, universities, and so on.
- The primary requirement of routers in these networks is to provide connectivity at a very low cost to a large number of end systems.
- In addition, a desirable requirement is to allow service differentiation to provide quality of service (QoS) guarantees for different departments of an enterprise.
- For enterprises, the network is considered as an operational expense and the goal is to minimize this expense.
- Hence, the routers targeted for enterprise deployment are required to have low cost per port, a large number of ports, and the ease of maintenance.
- Hence, it is challenging to design an enterprise router that satisfies these requirements for every port and still keep the cost low per port.

Elements of a Router

Elements of a Router

- A generic router consists of six major functional modules:
  - Network Interfaces
  - Forwarding Engines
  - Queue Manager
  - Traffic Manager
  - Backplane
  - Route Control Processor
- The various architectural components of a router and the functional modules each implements and they are:
  - Port Cards
  - Line Cards
  - Switch Fabric Cards
  - Route Processor Cards

Functional Modules

- Network Interfaces
  - A network interface contains many ports that provide the connectivity to physical network links.
  - A port terminates a physical link at the router and serves as the entry and exit point for incoming and outgoing packets, respectively.
  - A port is specific to a particular type of network physical medium.
- Forwarding Engines
  - These are responsible for deciding to which network interface the incoming packet should be forwarded.
  - When a port receives a new packet, it decapsulates L2 headers and sends the entire IP packet, or just the packet header, to the forwarding engine.
  - The forwarding engine consults a table, i.e., engages in a route lookup function, and determines to which network interface the packet should be forwarded.
  - This table is called the forwarding information base or simply the forwarding table.

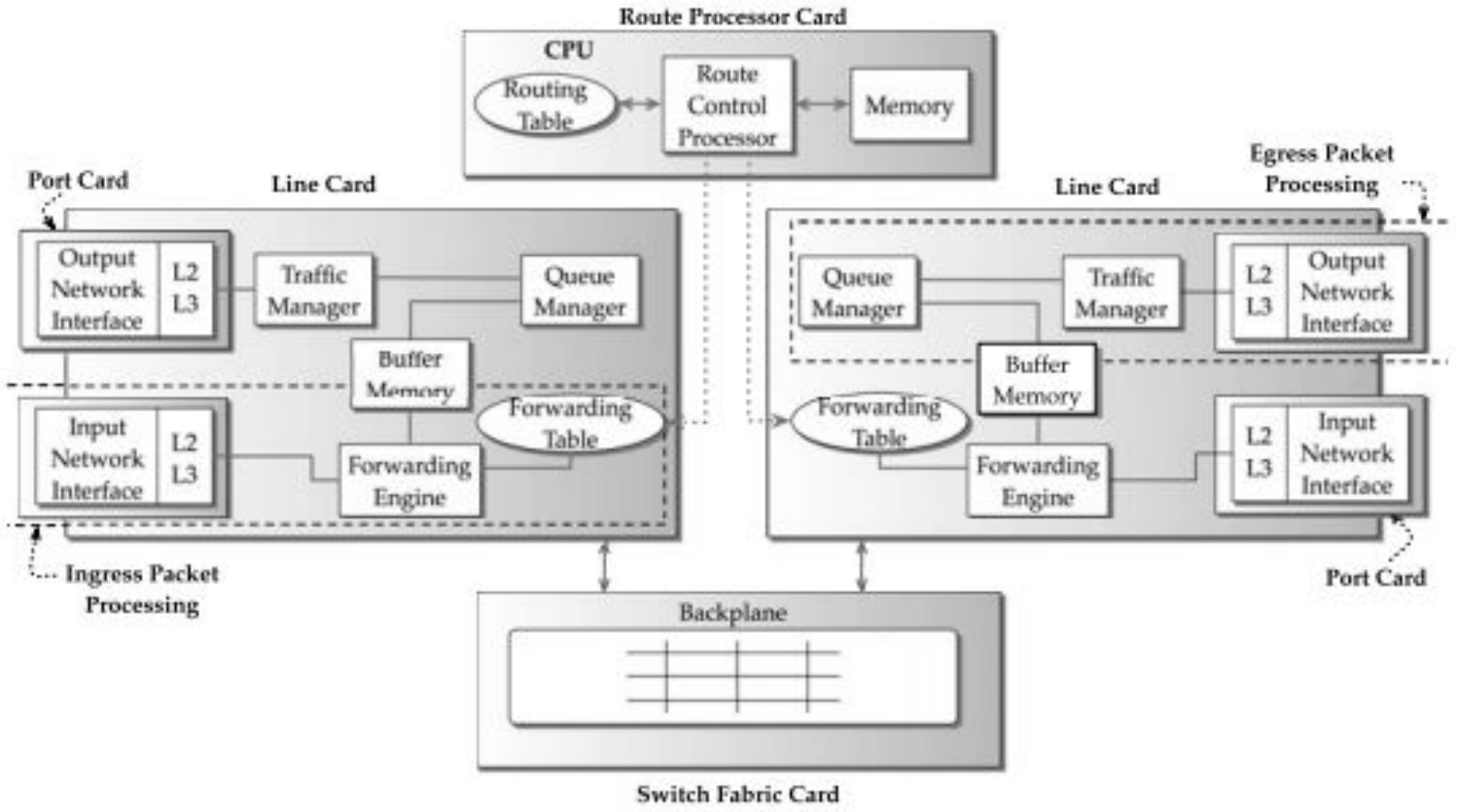
Functional Modules

- Queue Manager
  - This component provides buffers for temporary storage of packets when an outgoing link from a router is overbooked.
  - When these buffer queues overflow due to congestion in the network, the queue manager selectively drops packets.
- Traffic Manager
  - This component is responsible for prioritizing and regulating the outgoing traffic, depending on the desired level of service.
  - This is necessary as routers carry traffic from different subscribers and it is important to ensure that they get the level of service for which they pay.
  - The traffic manager shapes the outgoing traffic to the subscriber according to the service level agreement.

Functional Modules

- Backplane
  - This component provides connectivity for the network interfaces so that packets from an incoming network interface can be transferred to the outgoing network interface card.
  - The backplane can be either shared, where only two interfaces can communicate at any instant, or switched, where multiple interfaces can communicate simultaneously.
  - The aggregate bandwidth of all the attached network interfaces defines the bandwidth required for the backplane.
- Route Control Processor
  - The control processor is responsible for implementing and executing routing protocols. It maintains a routing table that is updated whenever a route change occurs.
  - Based on the contents of the routing table, the forwarding table is computed and updated.
  - In addition, it also runs the software to configure and manage the router. A route control processor also performs complex packet-by-packet operations like errors during packet processing.

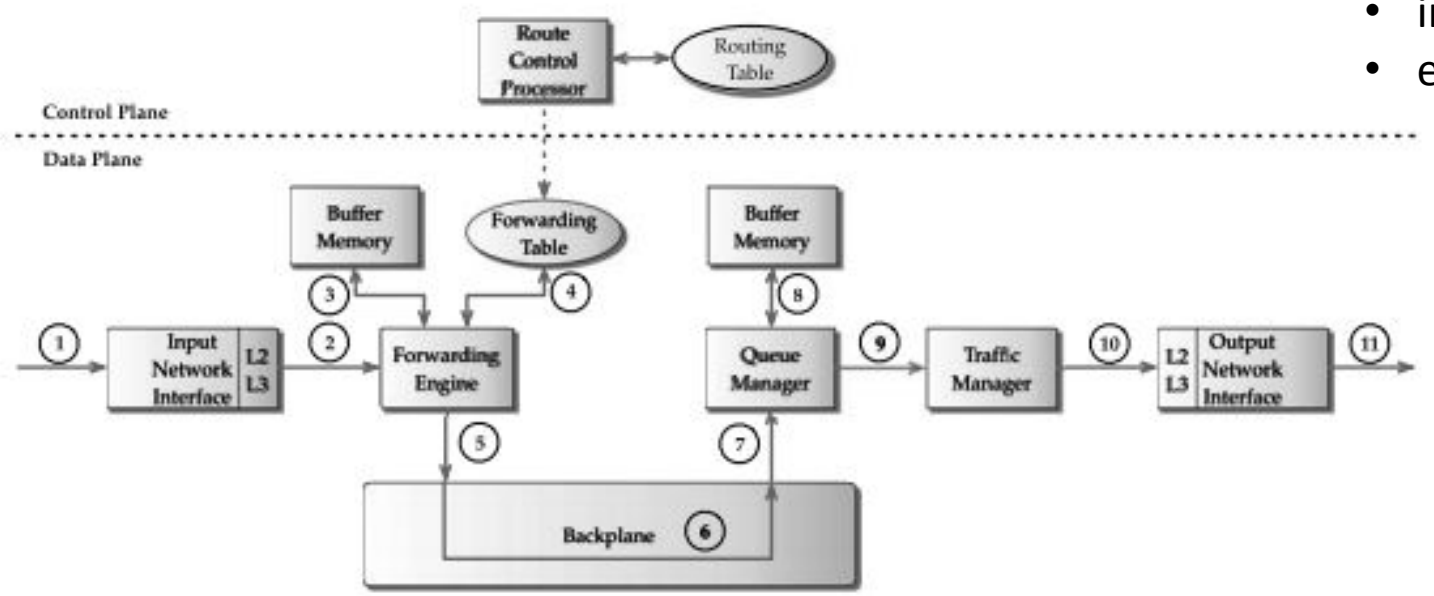
Architectural Components of a Router



Session 3

Packet Flow, Packet Processing

Packet Flow



Packet flow in a router.

The processing steps can be broadly grouped into

- ingress packet processing
- egress packet processing.

Ingress Packet Processing

- When an IP packet arrives from the network, it first enters the network interface.
- For the sake of discussion, let us assume that the packet is received on an Ethernet port.
- The network interface interprets the Ethernet header, detects frame boundaries, and identifies the starting point of the payload and the IP packet in the frame.
- The L2 processing logic in the card removes the L2 header and constructs a packet context.
- A packet context is a data structure that essentially serves as a scratch pad for carrying information between different stages of packet processing inside the router.
- The L2 processing logic appends to the packet context information about L2 headers, for instance, in the case of Ethernet, the source and destination MAC address.

Ingress Packet Processing

- Now the L2 processing logic peels off the payload, which is an IP packet, and along with the packet context sends it to the L3 processing logic.
- The L3 processing logic locates the IP header and checks its validity. It extracts the relevant IP header information and stores it in the packet context.
- The header information includes the destination address, source address, protocol type, DSCP bits (for differentiated services), and if the IP packet is carrying TCP or UDP payload, the destination and the source ports as well.
- At this point, the packet context contains enough information for route lookup and classification of the packet.
- Next, the entire packet context is sent to the forwarding engine in the line card. The forwarding engine searches a table (the forwarding table) to determine the next hop.
- The next-hop information contains the egress line card and the outgoing port the packet needs to be transferred. This information is populated in the packet context.

Ingress Packet Processing

- When the forwarding engine completes, the packet context is appended with the address of the packet in memory and is sent to the backplane interface.
- From the packet context, the backplane interface knows to which line card the packet needs to be transferred.
- It then schedules the packet for transmission along with the packet context over the backplane.

Egress Packet Processing

- When the packet reaches the egress line card, the backplane interface on the egress line card receives the packet and stores it in the line card memory.
- Meanwhile, the received packet context is updated with the new address of the memory location and sent to the queue manager.
- The queue manager examines the packet context to determine the packet priority.
- Recall that the priority was determined by the forwarding engine in the ingress line card during packet classification.
- Next the queue manager inserts the context of the packet in the appropriate queue.
- As different queues, depending on the priority, consume different amounts of bandwidth on the same output link, the queue manager implements a scheduling algorithm.

Egress Packet Processing

- The scheduling algorithm chooses the next packet to be transmitted according to the bandwidth configured for each queue.
- In some instances, the queues could be full because of congestion in the network.
- Once the packet is scheduled to be transmitted, the traffic manager examines its context to identify the customer and if there are any transmit rate limitations that need to be enforced according to the service contract. Such a mechanism is referred to as traffic shaping.
- Finally, the packet arrives at the network interface where L3 processing logic updates its TTL and updates the checksum.
- The L2 processing logic adds the appropriate L2 headers and the packet is transmitted.

Packet Processing: Fast Path versus Slow Path

Packet Processing

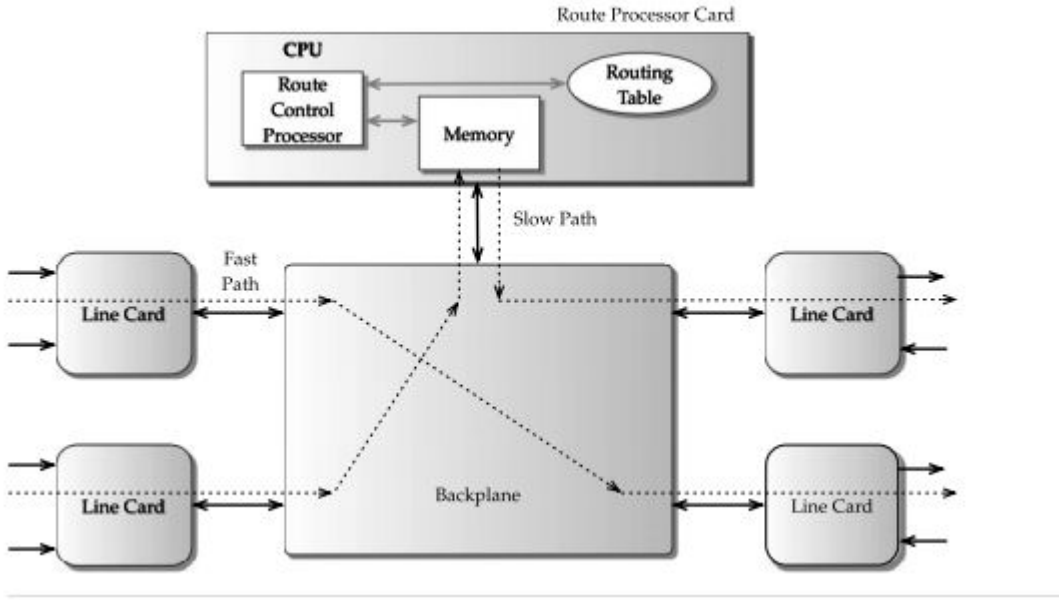
- The tasks performed by a router can be categorized into time-critical and non–time-critical operations depending on their frequency; they are referred to as fast path and slow path, respectively.
- The time-critical operations are those that affect the majority of the packets and need to be highly optimized in order to achieve gigabit forwarding rates.
- The time-critical tasks can be broadly grouped into header processing and forwarding.
- The header processing functions include packet validation, packet lifetime control, and checksum calculation, while forwarding functions include destination address lookup, packet classification for service differentiation, packet buffering, and scheduling.
- Since these tasks need to be executed for every packet in real time, a high performance router implements these fast path functions in hardware.

Packet Processing

- Non–time-critical tasks are typically performed on packets destined to a router for maintenance, management, and error handling.
- Such tasks include, but are not limited to:
  - Processing of data packets that lead to errors in the fast path and and generation of ICMP packets to inform the originating source of the packets
  - Processing of routing protocol keep-alive messages from adjacent neighbors and sending of these messages to the neighboring routers
  - Processing of incoming packets that carry route table updates and sending messages to neighboring routers when network topology changes
  - Processing of packets pertaining to management protocols, such as SNMP, and the associated replies

Packet Processing

- A packet using the fast path is processed only by the modules in the line cards as it traverses the router.
- On the other hand, a packet on the slow path is forwarded to the CPU, as many of the slow path tasks are implemented by the software running on it.
- Such an implementation is advantageous, as there is a clear separation between fast path and slow path.
- Consequently, there is no interference with the performance of packets on the fast path



Fast Path Functions

- In the fast path, the packets are processed and transferred from the ingress line card to the egress line card through the backplane.
- To achieve high speeds, the fast path functions are implemented in custom hardware, such as ASICs
- While such custom implementations are less flexible, the increasing need for more packet processing at the router, and the relatively small changes in IP packet format, makes the custom hardware implementation attractive.
- Now let us examine some of the fast path operations in detail.

Fast Path Functions - IP HEADER PROCESSING

- As soon as an IP packet enters a router, it is subjected to a set of validity checks to ensure that the packet is properly formed and the header is meaningful.
- The processing begins with a verification of the protocol version, as routers can support either IPv4 or both IPv4 and IPv6.
- If the version number does not match, then the packet could be malformed.
- The second step is for the router to check whether the length of packet reported by the MAC or the link layer is at least the minimum legal length of an IP packet.
- This test ensures that the IP header is not truncated by the MAC layer and filters packets less than the minimum intended length.
- Next, for IPv4, the value of the IP header checksum must equal the calculated header checksum computed by the router.
- The routers must decrement the TTL field in the IP header to prevent packets from getting caught in routing loops forever and consuming network resources

Fast Path Functions - PACKET FORWARDING

- The function of packet forwarding is to determine on which network interface a packet needs to be transmitted out of the router.
- The forwarding engine module controls this function using a forwarding table.
- The forwarding table is a summary of the routing table created by the route control processor.
- The router extracts the destination IP address from an incoming packet and performs a lookup in the forwarding table to determine the next-hop IP address for the packet.
- This procedure also decides which output port and network interface should be used to send the packet.
- The result of the lookup could lead to three possibilities: Local, Unicast and Multicast

Fast Path Functions - PACKET CLASSIFICATION

- The process of selectively identifying packets and applying the necessary actions according to certain rules is known as packet classification.
- A set of such rules is referred to as a classifier.
- A router should be capable of discriminating packets not only with the destination address, but also with the source address, source port, destination port, and protocol flags, commonly referred to as a 5-tuple.
- The source and destination addresses identify the participating endpoints, the protocol flags identify the type of payload, and the source and destination ports identify the application (assuming the payload is TCP or UDP).

Fast Path Functions - PACKET QUEUEING AND SCHEDULING

- As routers keep forwarding packets, there can be an instance where multiple packets arriving on different ingress network interfaces need to be forwarded to the same egress network interface simultaneously.
- Such burstiness in the Internet traffic requires buffers which serve as a temporary waiting area for packets to queue up before transmission.
- The order in which they are transmitted is determined by various factors such as the service class the packet, the service guarantees associated with the class, etc.
- Therefore, routers not only provide buffers but also require sophisticated scheduling function.
- The scheduling function prioritizes the traffic based on the bandwidth requirements and tolerable amount of delay by choosing the appropriate packet from these buffers.

Slow Path Operations

- The packets following the slow path are partially processed by the ingress line card before forwarded to the CPU for further processing.
- Once the CPU completes processing, it directly sends those packet to the egress line card.
- Some of the slow path functions are highlighted as follows.

Slow Path Operations - ADDRESS RESOLUTION PROTOCOL PROCESSING

- When a packet needs to be sent on an egress interface, the router needs to determine the data link or the MAC address for the destination IP address or the next-hop IP address.
- The mechanism for discovering dynamically requires the use of address resolution protocol (ARP).
- ARP assumes that the underlying network supports link-level broadcasts and sends a query ARP request containing the IP address.
- When the ARP reply comes in from the host with the link-level address, it is maintained as a part of the forwarding table in the router
- When a packet needs to be forwarded, these link-level addresses are obtained as a result of the address lookup operation on the forwarding table along with the outgoing interface.
- Hence a router designer might choose to implement ARP processing in the fast path for two reasons: performance and the need for direct access to the physical network.

Slow Path Operations - FRAGMENTATION AND REASSEMBLY

- Since a router connects disparate physical networks, there can be scenarios in which the message transfer unit (MTU) of one physical network is different from the other.
- When this happens, an incoming IP packet can be fragmented into small packets by the router if the output port is incapable of carrying the packet with its original length, that is, the MTU of the output port is less than that of the input port.
- Thus fragmentation enables transparent connectivity even across physical networks with different MTU sizes.
- Implementing reassembly in the fast path requires handling of packets arriving out of order, detecting lost fragments and discarding the remaining fragments in the buffers. Such tasks are complex to implement in hardware.
- However, packets destined for the router should be reassembled and usually it is implemented in software.
- Fragment reassembly can consume substantial amounts of both CPU and memory resources.

Slow Path Operations - ADVANCED IP PROCESSING

- Some of the advanced IP options include source routing, route recording, time stamping, and ICMP error generation.
- Source routing allows the sender of a packet to specify the route it should take to reach the destination.
- The main argument for implementing these functions in the slow path is that the packets requiring these functions are rare and can be handled as exceptional conditions.
- Hence, these packets can be processed in the control processor in the slow path.

Session 4

Shared CPU architecture,
Shared forwarding Engine Architecture,
Shared Nothing Architectures,
Clustered Architectures

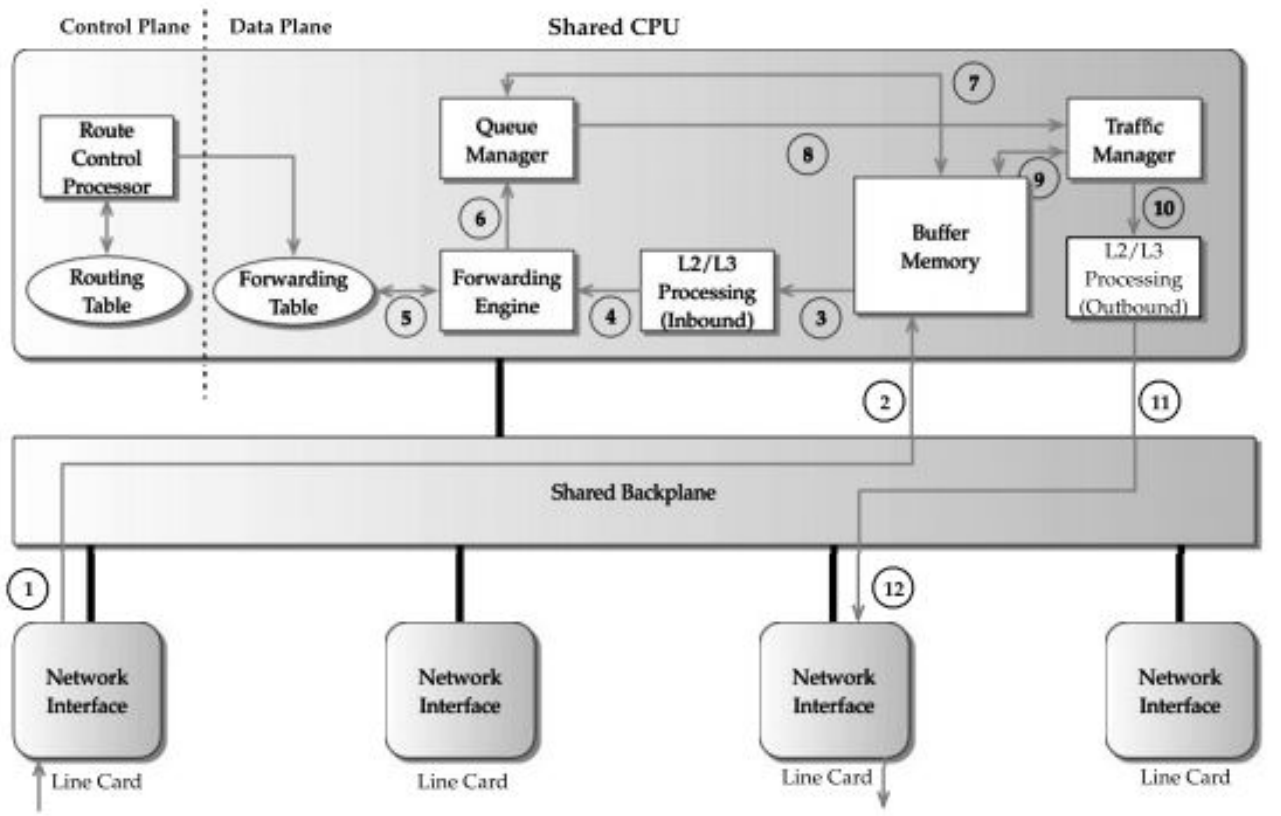
Router Architectures

- The router architectures are broadly classified into the following:
  - Shared CPU architectures
  - Shared forwarding engine architectures
  - Shared nothing architectures
  - Clustered architectures.

Shared CPU Architectures

- This architecture is built around a conventional computer architecture; a CPU with memory and multiple line cards are connected by a shared backplane.
- Each line card implements a network interface to provide connectivity to the external links.
- The CPU runs a commodity real-time operating system and implements the functional modules, including the forwarding engine, the queue manager, the traffic manager, and some parts of the network interface, especially L2/L3 processing logic in software.
- In addition, the same CPU also incorporates the functionality of the route control processor that implements the routing protocols, route table maintenance, and router management functions.
- All the line cards share the CPU for their forwarding function; hence, the name shared CPU architecture.

Shared CPU Architecture



Shared CPU architecture.

Shared CPU Architecture

- When a packet arrives at the line card, it raises an interrupt to the CPU.
- The interrupt service routine schedules a transfer of the packet to the buffer memory through the shared backplane.
- Once the transfer is complete, the CPU extracts the headers of the packet and uses the forwarding table to determine the egress line card and the outgoing port.
- The packet is subsequently prioritized by the queue manager and shaped by traffic manager.
- Finally, the packet is transferred from the memory to the appropriate output port in the egress line card.

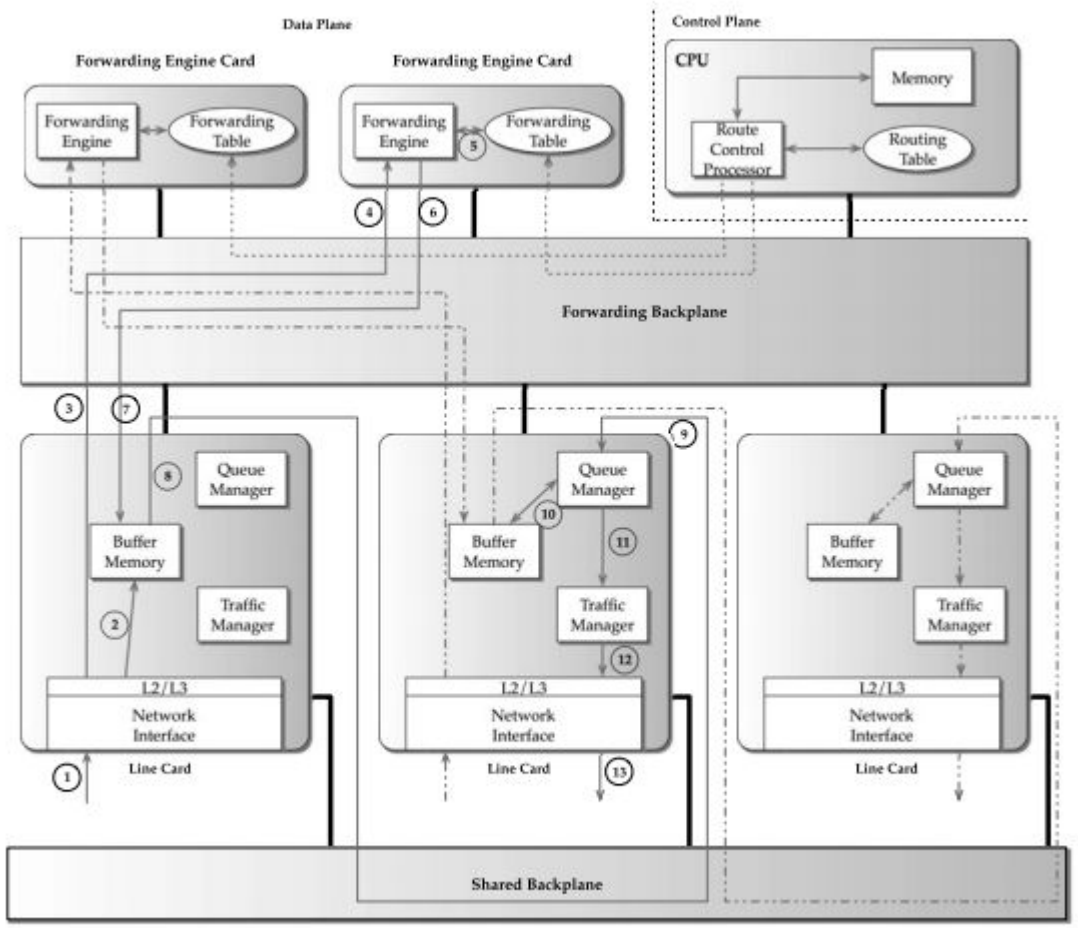
Shared CPU Architecture - Bottlenecks

- Each packet entering the system has to traverse the CPU; thus, the limited number of CPU cycles results in a processing bottleneck.
- The packet forwarding functions, such as forwarding table lookup, buffering and retrieval of the packet involve accessing memory. Due to mismatch in speed between memory and CPU, access to memory contributes to a larger amount of overhead. The memory access speeds have increased little over the last few years.
- The shared backplane becomes a severe limiting factor as each packet has to traverse the backplane twice. This effectively reduces the throughput by a factor of two

Shared Forwarding Engine Architectures

- In the shared CPU architecture, we identified that the shared CPU is one of the major bottlenecks, as it is in the path of every packet flow.
- The shared forwarding engine architecture is an attempt to mitigate the bottleneck by offloading the functionality of the forwarding engine to a dedicated card called forwarding engine cards.
- Each forwarding engine card contains a dedicated processor executing the software for route lookup and memory for storing the forwarding table.
- With multiple such cards, many packets can be processed in parallel, which considerably scales the packet forwarding speed.
- The shared forwarding engine architectures were used to build routers capable of forwarding gigabits per second.

Shared Forwarding Engine Architecture



Shared forwarding engine architecture using two shared backplanes.

Shared Forwarding Engine Architecture

- In this architecture, multiple line cards are connected through a shared backplane through which the packets are transferred from one line card to another.
- Line cards and forwarding engine cards are connected through a separate shared backplane called forwarding backplane.
- The rationale behind using two different backplanes is to separate the data traffic from the traffic generated for the forwarding engine cards, thereby improving throughput

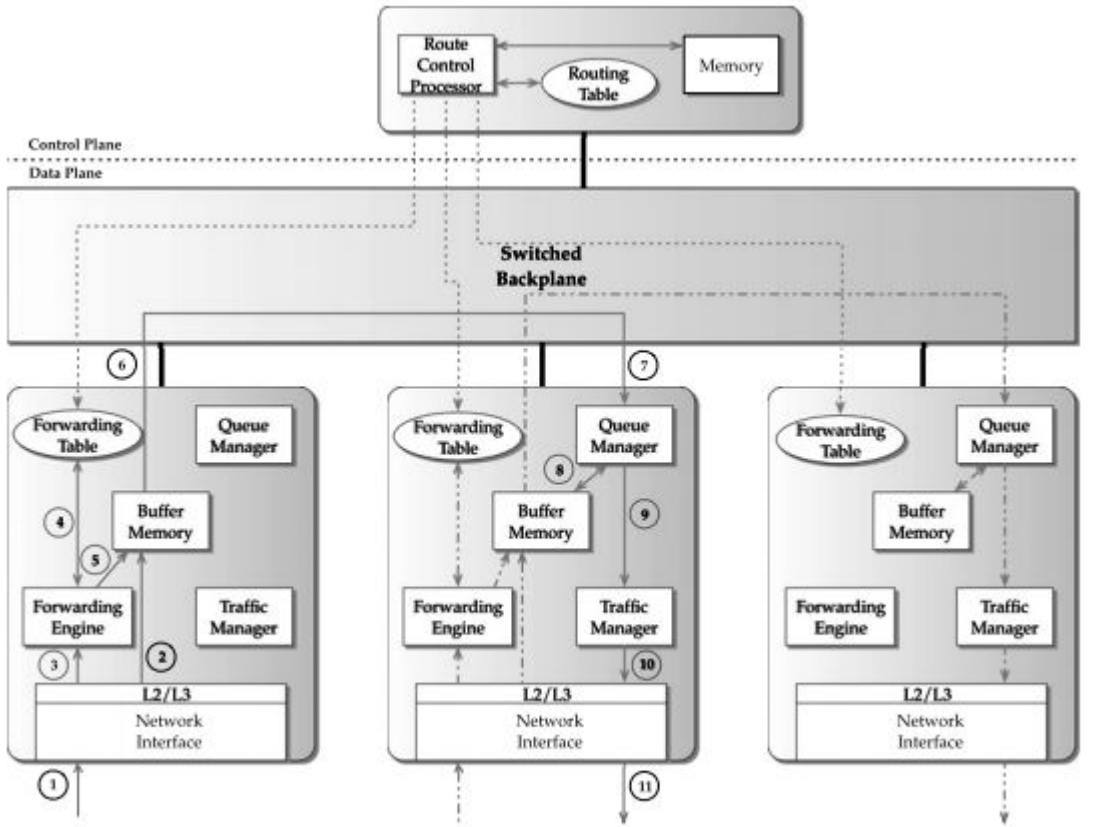
Shared Forwarding Engine Architecture

- Since there are multiple forwarding engines, multiple IP headers can be processed in parallel.
- This could lead to the situation where packets that arrived later might finish their route lookup earlier than the packets that entered the router earlier.
- The time required to process each packet depends on the actual load of the forwarding engine.
- The main advantage of this architecture is the ability to scale to higher forwarding speeds.
- Another advantage of this architecture is that it provides flexibility; the forwarding engine cards can be added whenever needed so that the necessary forwarding speed can be achieved for high-speed core routers.
- A key drawback is the use of a shared backplane that does not provide sufficient bandwidth for transmitting packets between line cards and limits the router throughput.

Shared Nothing Architectures

- A shared nothing router architecture offloads all the packet forwarding functions to the line cards.
- The line cards implement these functions using custom hardware for high performance and do not share any of these components with other line cards.
- Hence, this architecture is named as shared nothing.
- Now since the line cards are capable of handling large number of packets, the backplane should be fast enough to handle aggregate input from all the line cards.
- Hence, this architecture employs switched backplanes, which makes this setup capable of multiple transfers simultaneously

Shared Nothing Architecture



Shared nothing architecture.

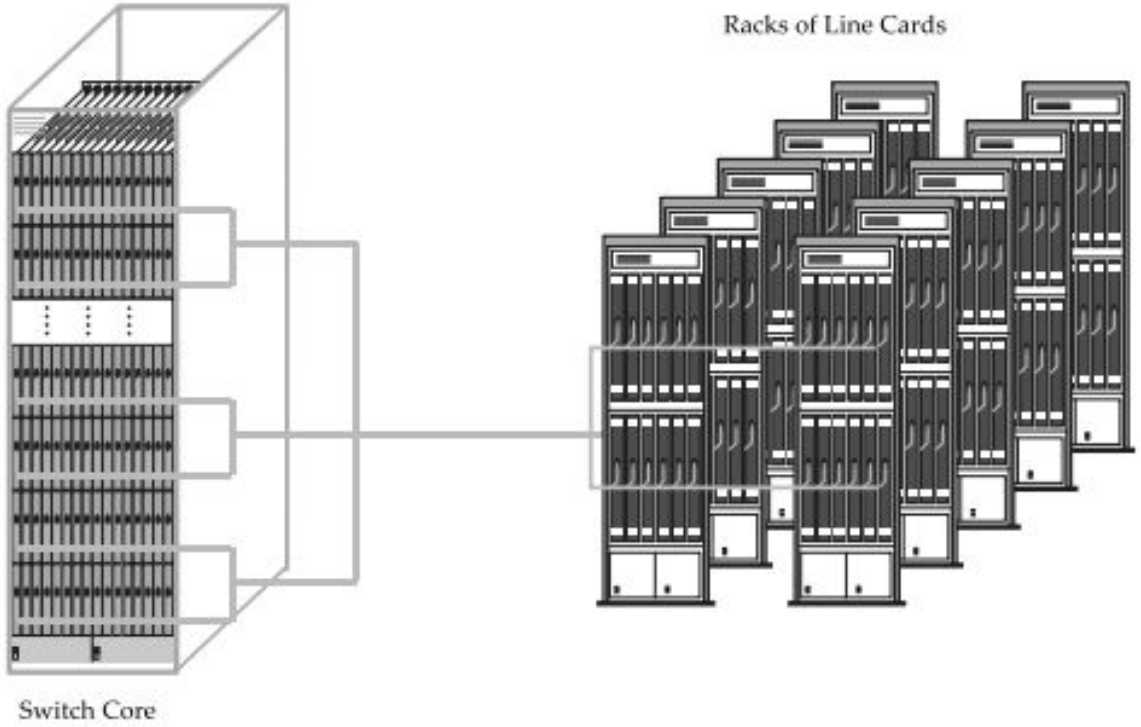
Clustered Architectures

- For increasing the number of line cards and the aggregate system throughput, major vendors use a clustering approach.
- This approach uses multiple chassis–containing line cards connected to central switch core.
- A variation of this approach is the use of multiple independent routers connected to a central switch core but function as a single router.
- In these architectures, the chassis-containing line cards are connected to the switch core using very-high-speed optical links.

Clustered Architectures

- A packet entering a network interface in a line card, depending on the result of route lookup, can be destined to a line card in the same chassis or a line card in a different chassis.
- In the latter case, the packet has to be forwarded through the switch core that sends it to the correct chassis.
- Once the packet reaches the chassis, it is forwarded through the appropriate egress line card.
- The advantage of this architecture is the ability to incrementally add the line card chassis depending on the need. A disadvantage of this architecture is that the switch core is a single point of failure

Clustered Architecture



Clustered router architecture with a central switch core

Session 5

Impact of Addressing on lookup,
Longest Prefix Matching

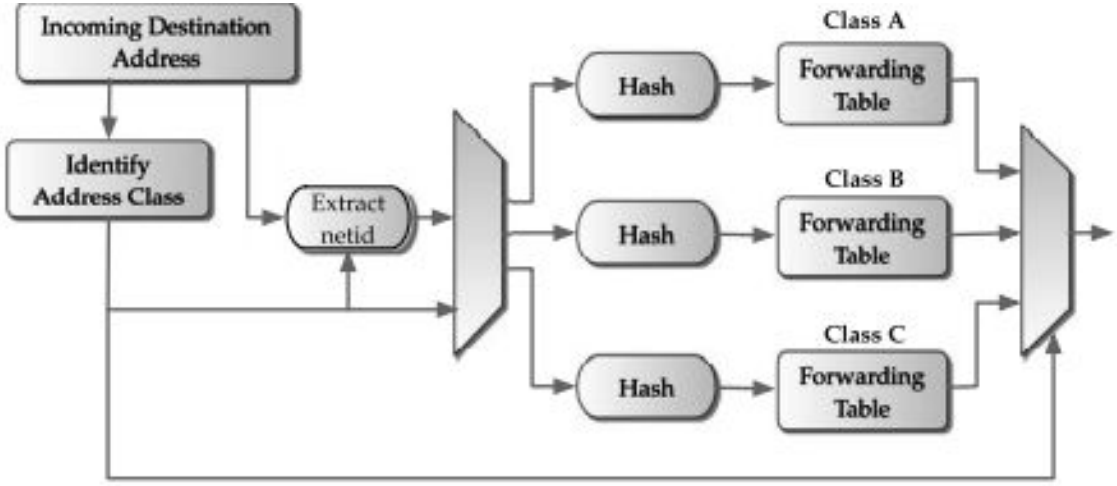
Address Lookup

- The primary function of routers is to forward packets toward their final destination.
- To accomplish this, a router must decide for each incoming packet where to send it next.
- To be precise, the forwarding decision consists of two components:
  1. finding the address of the next-hop router to forward the packet to, and
  2. determining the egress interface through which the packet should be sent.
- This forwarding information, referred to as next-hop information, is stored in a forwarding table populated by information gathered from the routing protocols.
- This forwarding table is consulted using the packet's destination address as the key. Such an operation is called an address lookup.

Impact of Addressing on Lookup

- With the classful addressing scheme, the forwarding of packets is straightforward.
- Routers need to examine only the network part of the destination address to forward it to the destination.
- Thus, the forwarding table needs to store a single entry (the network part) for routing the packets destined to all the hosts attached to a given network.
- Such a technique is called address aggregation and uses prefixes to represent a group of addresses.

Addressing on Lookup



Lookup operation in a classful IP addressing scheme

Longest Prefix Matching

- The problem of identifying the forwarding entry containing the longest prefix among all the prefixes matching the destination address of an incoming packet is defined as the longest matching prefix problem.
- This longest prefix is called the longest matching prefix.
- It is also referred to as the best matching prefix.

Example - Identifying the longest matching prefix

- Consider the forwarding table at a router.

A forwarding table.

Entry Number	Prefix	Next-Hop
1	98.1.1.1/24	eth3
2	171.1.0.0/16	so6
3	171.1.1.0/24	fe5

- Each entry contains the prefix and the name of the outgoing interface.
- Entry 1 indicates that the packet matching prefix 98.1.1.1/24 will go out on interface eth3.
- If the destination address of the incoming packet is 171.1.1.2, then it will match prefix 171.1.0.0/16 in entry 2 as well as 171.1.1.0/24 in entry 3. Since prefix 171.1.1.0/24 is the longest matching prefix, the packet will be forwarded on the outgoing interface fe5.

Longest Prefix Matching

- The difficulty of the longest prefix matching arises because of the following reasons.
- First, a destination IP address does not explicitly carry the netmask information when a packet traverses through.
- Second, the prefixes in the forwarding table against which the destination address needs to be matched can be of arbitrary lengths; this could be as a result of an arbitrary number of network aggregations.
- To conclude, the adaptation of CIDR has made route lookups more complex than they were when the classful addressing scheme was used.

Session 6

Naïve Algorithms, Binary Tries

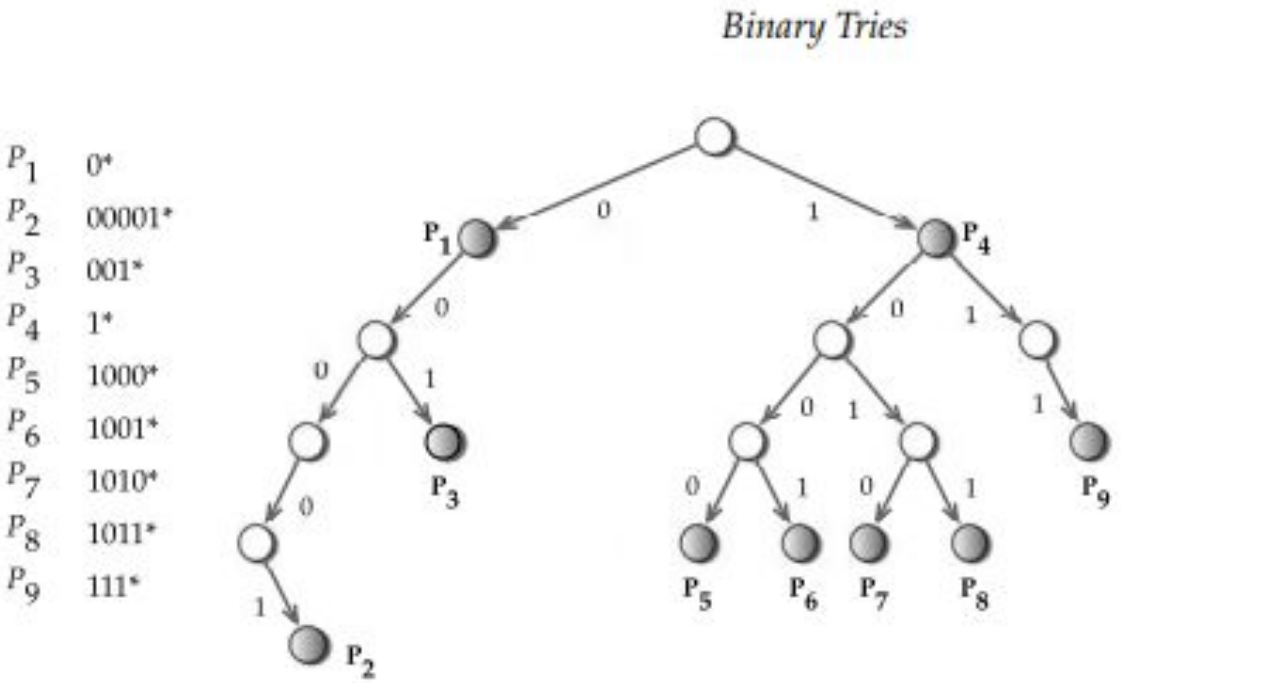
Naïve Algorithms

- The simplest algorithm for finding the best matching prefix is a linear search of prefixes.
- It uses an array in which the prefixes are stored in an unordered fashion.
- The search iterates through each prefix and compares it with the destination address.
- If a match occurs, it is remembered as the best match and the search continues.
- The best match is updated as the search walks through each prefix in the array.
- When the search terminates the last prefix remembered is the best matching prefix.
- The time complexity for such a search is O(N). Linear search might be useful if there are very few prefixes to search; however, the search time degrades as N becomes large.

Binary Tries

- The binary trie is the simplest of a class of algorithms that is tree-like.
- The term trie comes from “retrieval” and is pronounced “tree.”
- However, most often to verbally distinguish a trie from a general tree, it is pronounced as “try.”
- A binary trie is a tree structure that allows a natural way to organize IP prefixes and uses the bits of prefixes to direct the branching.
- Each internal node in the tree can have zero, one, or two descendants.
- The left branch of a node is labeled 0 and the right branch is labeled 1.

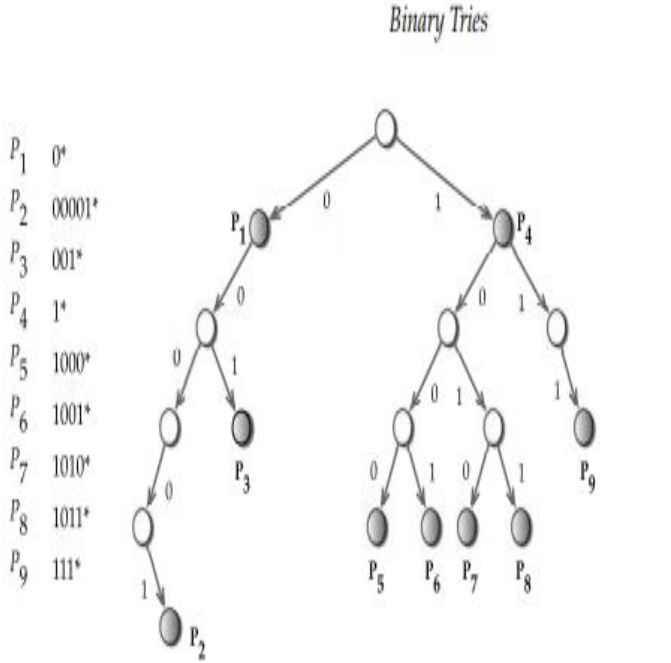
Binary Tries



Binary trie data structure

Binary Tries

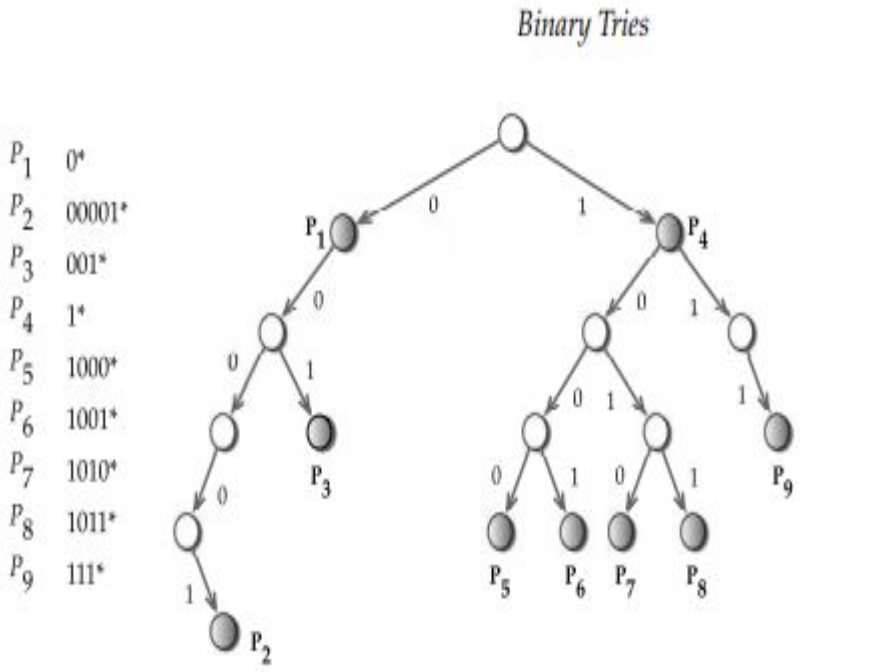
- In a binary trie, a node l represents a prefix formed by concatenating the labels of all the branches in the path from the root node to l.
- For example, the concatenated label along the path to node P2 is 00001, which is the same as prefix P2.
- Note that some of the nodes are shaded in gray while the remaining nodes are not.
- The gray-shaded nodes correspond to actual prefixes.
- These nodes contain the next-hop information or a pointer to it.
- As can be seen, prefixes can be either in the internal nodes or at the leaf nodes.
- Such a situation arises if there are exception prefixes in the prefix aggregation.
- For instance, in Figure, the prefixes P2 and P3 represent exceptions to prefix P1.



Binary trie data structure

Binary Tries - Search Operations

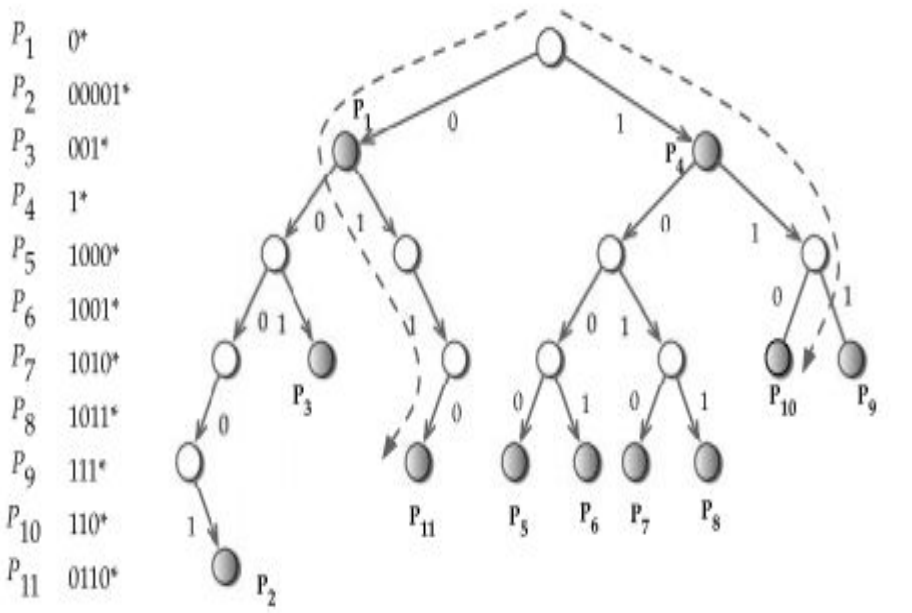
- Consider searching the binary trie shown in Figure for an address that begins with 001.
- The search starts at the root node of the trie and the first bit is examined.
- Since it is a 0, the search proceeds toward the left branch and encounters the node with the prefix P1.
- We remember P1 as the best matching prefix found so far.
- Then, we move left as the second address bit is another 0; the node encountered this time does not have a prefix, so P1 remains the best matching prefix so far.
- Next, we examine the third bit, which is a 1, and leads to prefix P3.
- Since P3 is a better matching prefix than P1, it is remembered.
- Now, P3 is a leaf node and, thus, the search terminates and the prefix P3 is declared the best matching prefix.



Binary trie data structure

Binary Tries – Insert Operations

- Consider inserting prefixes 110 and 0110, referred to as P10 and P11, respectively, in the binary trie shown in Figure.
- Since the first bit of P10 is 1, the search moves to the right and reaches the gray node P4.
- Now the second bit is examined, which again guides the search right.
- As the third bit is 0, there is no left branch to take and, thus, a new node P10 is created and attached.
- The next-hop information for this prefix is stored in the node itself.
- Now consider inserting prefix P11. After inspecting the bits, we find that there is no right branch to take on the node with prefix P1.
- Thus, new nodes are added that create the path to prefix node P11.



Inserting new prefixes in a binary trie.

Session 7

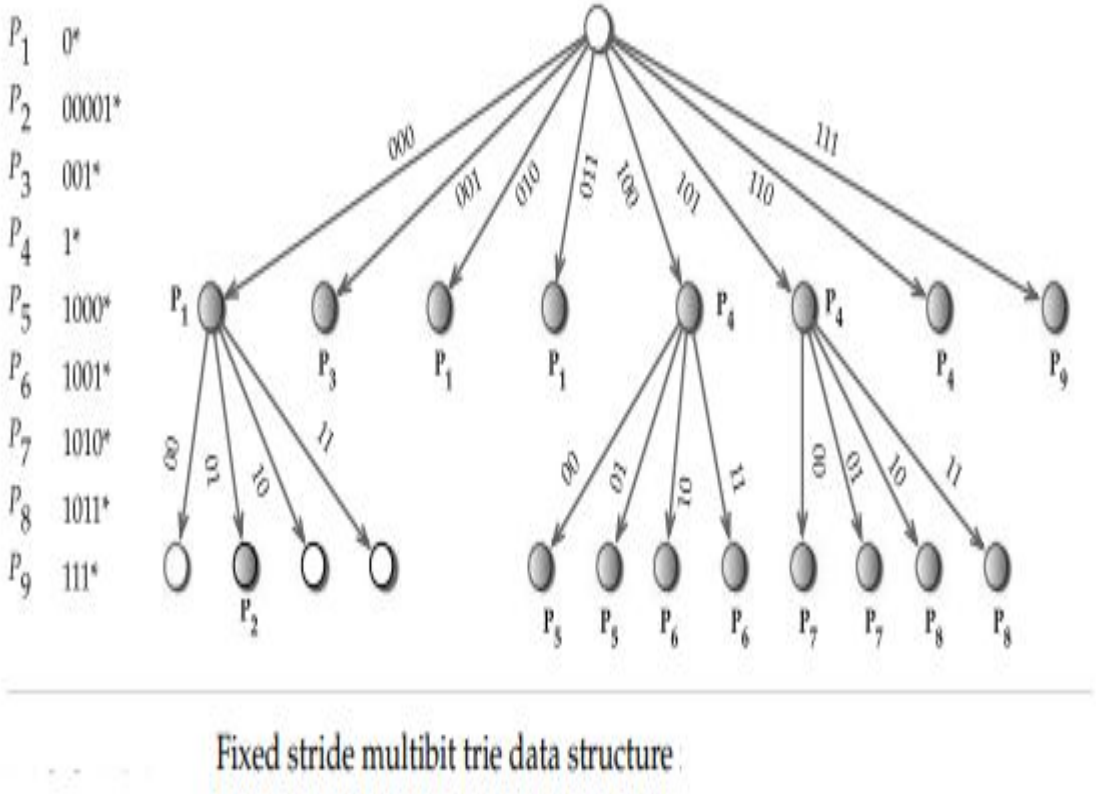
Multi-bit Tries, Compressing multi-bit strides

Multibit Tries

- A multibit trie is a trie structure that allows the inspection of bits in strides of several bits.
- Each node in the multibit trie has 2^k children where k is the stride.
- If all the nodes at the same level have the same stride size, we call it a fixed stride; otherwise, it is a variable stride.
- As one can see, since multibit tries allow the data structure to be traversed in strides of several bits at a time, they cannot support prefixes of arbitrary lengths.
- To use a given multibit trie, a prefix must be transformed into an equivalent prefix of longer length to conform with the prefix lengths allowed by the structure.

Fixed Stride Multibit Trie

- If all the nodes at the same level have the same stride size, then the multibit trie is called a fixed stride multibit trie.
- The example multibit trie uses a stride of 3 bits and 2 bits for all nodes in level 1 and level 2, respectively
- Here prefixes of lengths other than 3 and 5 should be transformed into prefixes of lengths 3 and 5.
- Applying prefix expansion, prefix P1 is expanded into four prefixes 000*, 001*, 010*, and 011* of length 3.
- One of the expanded prefixes 001* is the same as prefix P3.



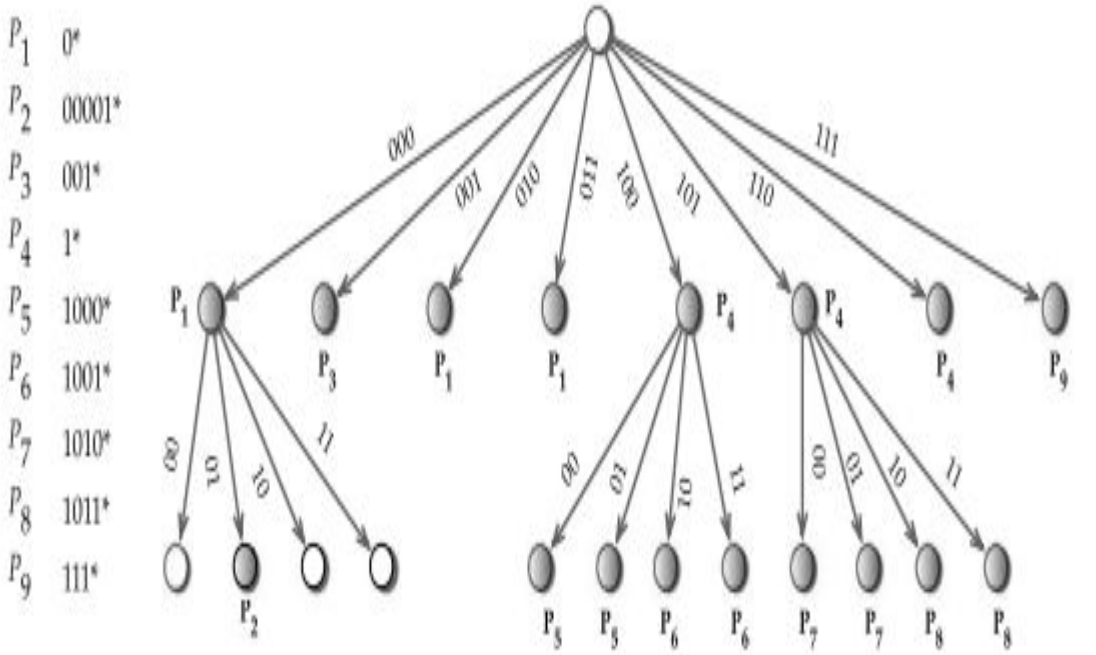
Fixed stride multibit trie data structure

Multibit Trie - Search Algorithm

- The search proceeds by breaking up the destination address into chunks that correspond to the strides at each level.
- Then these chunks are used to follow a path through the trie until there are no more branches to take.
- Each time a prefix is found at a node, it is remembered as the new best matching prefix seen so far.
- At the end, the last best matching prefix found is the correct one for the given address.

Multibit Trie - Update Algorithm

- Consider the example in Figure, In the subtrie at the first level we are interested in finding the best matching prefix among prefixes P1, P3, P4, and P9.
- For the leftmost subtrie at the second level the best matching prefix will be selected from only prefix P2.
- Similarly, in the second subtrie at the second level, the best matching prefix is selected from the prefix set P5 and P6 while for the rightmost subtrie it is selected from prefixes P7 and P8.
- Thus, multibit tries divide the problem of finding a best matching prefix into smaller subproblems in which the local best matching prefixes are selected from among a subset of prefixes.
- This works out to the advantage of prefix updates



Fixed stride multibit trie data structure

Compressing Multibit Tries

- The aggressive use of prefix expansion in multibit tries introduces several new prefixes.
- These new prefixes inherit the same next-hop information as that of the original prefix.
- Furthermore, the use of large strides creates a greater number of contiguous nodes that have the same best matching prefix.
- Such redundant information can be compressed, saving memory and at the same time making the search faster because of the smaller height of the trie.
- After compression, the entire data structure can even fit into an L1 cache, which further speeds up the search as the access times are an order of magnitude faster than SRAM

Session 8

Search By Length Algorithms,
Search By value approaches

Search by Length Algorithms

- The search using the length dimension can be either linear or binary.
- To facilitate the search, the prefixes can be organized in two different ways on the length dimension.
- As we have seen earlier, one approach is to arrange the prefixes in a trie.
- Searching of the trie can be viewed as a sequential search on length.
- Another possible approach for organizing the prefixes is to use a hash table for each distinct length and employ either a linear search or binary search on these tables to locate the best matching prefix

Linear Search on Prefix Lengths

- Since we need to look for the longest matching prefix, the search begins with the table containing the longest prefixes.
- If the destination address is D and the longest prefix is l, the search extracts the first l bits and initiates a search in the hash table for length l entries.
- If an entry matches, then the search terminates since it is the best matching prefix.
- If not, the search moves to the first length smaller than l, say l', such that L[l '] is nonempty. Thus the search continues by examining the tables of decreasing prefix lengths until it either finds a match or runs out of lengths.

Binary Search on Prefix Lengths

- While a linear search requires O(W) in the worst case, a better search strategy is to employ a binary search on the array L
- The binary search starts at the median prefix length and divides the search space (in this case, prefix lengths) in each step by half.
- At every step, the hash table associated with that length is searched.
- Based on the results of searching the hash table, the choice of the half on which to continue the search is determined.
- The result can be one of the values: found or not found.
- If a match occurred at length l, then the search is directed to the half in which the lengths are strictly greater than l for a longer match.
- If no match was found at length l, then the search is continued on the half in which the lengths are strictly less than l.

Search by Value Approaches

- Sequentially searching all the prefixes is the simplest method to find the best matching prefix.
- While this approach is not scalable as N becomes large, the exhaustive search does get rid of the length dimension.
- Alternatively, it is possible to use a binary search on the prefix values that could perform better than an exhaustive search.

Session 9

Hardware Algorithms,
Comparing Different Approaches

Hardware Algorithms

- The primary motivation for implementing lookups in hardware comes from the need for higher packet processing capacity driven by high Internet traffic growth.
- Such a growth leads to the development of faster interfaces that can support OC-192 and OC-768 speeds.
- At such high speeds, a software-based implementation using random access memory (RAM) is not fast enough.
- While the software implementations have the flexibility for later modifications, the need for such modifications is minimal.
- Since IPv4 is used so widely, disruptive modifications to addressing schemes or the best matching prefix mechanism seem unlikely in the near future.

Hardware Algorithms

- RAM-Based Lookup
  - RAM supports the two major operations: writing a data item into a specific address and reading a data item from a given address.
  - RAM can be used to perform the lookup in a single memory access if the destination address is used as a direct index (RAM address) into the memory.
  - The data item stored in that address will be the nexthop information.
- Ternary CAM-Based Lookup
  - Content-addressable memories (CAMs) provide a performance advantage over conventional RAM-based memory search algorithms by comparing the desired information against the prestored entries simultaneously.
  - This results in an order of magnitude reduction in search time.
  - Since CAMs are an outgrowth of RAM technology, they employ conventional memory (usually SRAM) with the additional circuitry for comparisons that enable search operations to complete in a single clock cycle
- Multibit Tries in Hardware
  - It uses a two-level multibit trie with fixed strides for each level.
  - The first level uses a stride of 24 bits while the second level uses a stride of 8 bits. This scheme is based on the following two key observations:
    - Most of the prefix entries in routing tables in core routers are 24 bits or less. This is attributed to the aggressive route aggregation at intermediate routers.
    - The cost of memory continues to decline while the density of memory doubles every year.

Comparing Different Approaches

- Simple binary tries and path compressed tries are primarily used for software implementations
- An LC-trie is a compacted version of a path compressed trie in which complete subtries are level compressed from the top down
- The Lulea scheme use multibit tries, but the wasted space in the trie nodes is compressed using bitmaps
- The binary search on prefix lengths scales very well to IPv6 addresses that are 128 bits long.
- The binary search on prefix ranges provides reasonably fast lookup performance and consumes only a reasonable amount of storage