
CCIE 8 Weeks
Learn, Practice, Achieve
ALL-IN-ONE
CCIE ROUTING AND SWITCHING V5.1
WRITTEN EXAM 400-101 CERT GUIDE
Learn, Practice and Ace the New CCIE Written Exams with Test Prep Resources from CCIEin8Weeks™
CCIEin8Weeks.com
PAUL ADAM, CCIE®
4TH EDITION
ROUTING
SWITCHING V5.1
EXAM 400-101 CERT GUIDE
s helped thousands of test takers and learners
pare, practice and pass the CCIE® written exams
8Weeks™ contains supplemental resources such
s, study plans and blog articles to help you
r the exams and the real world. Our practice
ed frequently to ensure your exam success.
uide methodically and precisely covers all the
CCIE® Routing and Switching 400-101 V5.1 exam.
updated Cisco official exam blueprint for CCIE®
. Every chapter contains an Exam Essentials
d to reinforce the key concepts and to help you
ance
rk Principles
2 Technologies
r 3 Technologies
Technologies
structure Security
astructure Services
lving Technologies V1.1
CCIEIN8WEEKS | FACEBOOK.COM/CCIEIN8WEEKS
CCIE 8 Weeks
2nd EDITION

All-in-One CCIE Routing and Switching V5.1
400-101 Written Exam Cert Guide

4th Edition

Contents at a Glance

Table of Contents

Preface

Congratulations! You have taken your first step towards passing the CCIE R&S 400-101 (V5.1) written exam. This written exam cert guide along with practice quizzes from CCIEin8Weeks will help you prepare for this exam.

This book is dedicated to *all those souls who will never settle for less than they can be, do, share, and give!*

What this Exam Cert Guide covers

As you may already have noticed on the "Contents at a Glance" page that this guide has been formatted around the Cisco's official CCIE 400-101 (V5.1) exam topics or curriculum. So as you read through parts and chapters, you know exactly where you're within your learning journey. All contents are carefully covered in enough details however still trimmed to exactly what's necessary to pass the exam. The result of this approach is that you have 300 odd pages on your hands (as opposed to 700 to 1,400 pages long reference manuals!). Each exam topic has "Further Reading" section, which you can refer to if you are looking for more in-depth details. We have done the research for you.

How to use this Exam Cert Guide

This guide is for anyone who's studying for CCIE R&S written 400-101 (V5.1) exam, regardless if this is your first time sitting for a CCIE written exam or you're a pro who's recertifying.
I strongly suggest to take a methodical approach for exam preparation, i.e. start with a target date when you would like to sit for the actual exam and then work backwards to see what kind of study plan would work for you. I believe you can cover the entire contents within eight weeks including the practice questions (refer to website). We strongly suggest you to also use other study resources in addition to bringing your networking experience to bear in order to successfully pass the exam.

If you're totally in a crunch and consider yourself an advanced learner, you could try your luck by skimming through the "Exam Essentials" sections for each chapter- I guarantee you that you will be better prepared to tackle the exam with it than without!

What's available on the CCIEin8Weeks website

CCIEin8Weeks.com carries the extras (sold separately) that go hand in hand with this exam cert guide to further ensure your exam success!

Website includes:

- Seven practice quizzes (one for each section as per official curriculum), one final exam
- Four study plans, to help you track your progress or help you come up with your own plan
- CCIE Exam community forum, to help you interact with others, share knowledge, find study partners etc.
- Last but not least, this exam cert guide in printable PDF format

Part 1 Network Principles

Chapter 1: Network Theory

This chapter covers the following exam topics from Cisco's official 400-101 (v5.1) written exam curriculum.

- Basic software architecture differences between classic IOS and IOS XE
- Cisco Express Forwarding (CEF)
- General network challenges (such as unicast flooding, out of order packets, asymmetric routing and impact of microbursts)
- IP operations (such as ICMP unreachable/redirect, IPv4 options, IPv6 extension headers, IPv4/v6 fragmentation, TTL, IP MTU)
- TCP operations (IPv4/IPv6 PMTU, Maximum Segment Size-MSS, Latency, Windowing, BW delay product, global synchronization)
- UDP operations (TCP starvation / UDP dominance, latency, RTP/RTCP concepts)

Chapter 1: Network Theory

Describe basic software architecture differences between IOS and IOS XE

Cisco classic IOS has always had a monolithic software architecture, which means that it is both downloaded and run as a single binary image where all processes share the same memory address space. Monolithic and non-modular architecture leads to no memory protection between processes, as a result software defects in classic IOS code can potentially corrupt data used by other processes. It also has a run to completion scheduler, which means that the kernel does not preempt a running process — the process must make a kernel call before other processes can be scheduled and get a chance to run.

In all variations of classic Cisco IOS, packet routing and forwarding (switching) are distinct functions. Routing and other protocols run as IOS processes and contribute to the formation of Routing Information Base (RIB). This is processed to generate the final IP forwarding table (FIB, Forwarding Information Base), which is used by the forwarding function of the router. On router platforms with software-based forwarding (e.g., Cisco 7200 or Cisco ISR G2) most traffic handling is done at interrupt level using Cisco Express Forwarding (CEF). This helps avoid process context switching that would need to be done otherwise to forward packets. Routing functions such as OSPF or BGP run at the process level. In routers with hardware-based forwarding, such as the Cisco ASR1000 (which runs IOS XE), ASR9000 or CRS-1 or NCS series (which run IOS XR), IOS computes the FIB in software running on route processor (RP) hardware (typically x86 CPUs) and loads it into the forwarding hardware (such as an ASIC or a network processor), which performs the actual packet forwarding function.
The IOS XE is a POSIX based environment along with various open source software for the common drivers, tools and utilities needed to manage the system. In addition to the standard set of off-the-shelf drivers, IOS XE also includes a set of Cisco specific drivers and associated chassis/platform management modules.

On top of the base operating system (Linux) and drivers, IOS XE provides a comprehensive set of infrastructure modules which define how software is installed, how processes are started and sequenced, how high-availability (HA) and software upgrades are performed. The core application that runs on top of this new infrastructure is the IOS feature set in the form of IOS daemon (IOSd). By running Cisco IOS, products reap the benefits of an extensive feature set for routing and switching platforms that has been built into IOS over last two decades.

Finally, the evolved IOS architecture is specifically designed to accommodate other applications outside of IOS blob or IOSd. These applications can be upgraded or restarted independently of IOSd. If an application does require services from IOS, it can integrate with IOS through a set of client libraries called service points. These service points generically extend IOS information and services to outside applications such that these services are not replicated or managed separately. IOS XE is not a new network "OS" per se, it is rather an incarnation of classic IOS ("IOS") where role of classic IOS is reduced to an application running on top of a Linux kernel. This approach also allows building routing/switching platforms that use a variety of data plane hardware (ASICs or network processors such as Cisco's QFP or CPP) by way of the abstraction provided between control and data planes.

Each Cisco IOS XE Software subpackage provides specific functions for the Cisco ASR 1000 Series router.

Table 1-1, shows functions of Cisco IOS XE Software Subpackages

Package	Function
RPBase	Provides the operating system software for the route processor
RPControl	Controls the control plane processes that interface between Cisco IOS Software and the rest of the platform
RPAccess	Provides software required for router access ● The non-K9 version of this subpackage is included only in consolidated packages that do not have cryptographic support. ● The K9 version of this sub-package includes restricted components (Secure Sockets Layer [SSL] and Secure Shell [SSH]); consolidated packages with this subpackage are subject to export controls.
RPIOS	Provides the Cisco IOS Software kernel, which is where Cisco IOS Software features are stored and run; each consolidated package has a different RPIOS subpackage
ESPBase	Provides the ESP operating system and control processes and the ESP software

SIPSPA	Provides the shared port adaptor (SPA) driver and associated field-programmable device (FPD) images
SIPBase	Controls the Session Initiation Protocol (SIP) carrier card operating system and control processes

Control plane and Forwarding plane

IOS XE allows development of data plane ASICs outside the IOS instance and have them program to a set of standard APIs which in turn enforces Control Plane and Data Plane processing separation. It achieves Control Plane / Data Plane separation through the introduction of the Forwarding and Feature Manager (FFM) and its standard interface to the Forwarding Engine Driver (FED). FFM provides a set of APIs to control plane processes. FFM programs the data plane via the FED and maintains forwarding state for the entire system. The FED is the instantiation of the hardware driver for the data plane.

Table 1-2, compares classic IOS ("IOS) and IOS XE architectures

Software Architecture	Classic IOS ("IOS")	IOS XE
Monolithic	Yes	No
Control and Data Plane separation (Software)	No	Yes
Control and Data Plane separation (Hardware)	Platforms that run classic IOS do not have clear separation of control and data plane in hardware	Platforms that run IOS XE do have clear separation of control and data plane hardware
Feature parity	All IOS versions contain a singular set of IOS features (E.g. all platforms in T train contain same features such as Cisco ISR G2 and 7200)	All IOS XE versions contain software that is specific to one or a set of platforms E.g., IOS XE running on Cisco ASR 900 contains different feature set than one running on Cisco ASR1000
Example Platforms	All software based platform	Cisco ISR 4K

	Cisco ISRs (3900, 2900, 1900, 8XX), 7200 (end of life)	Cisco ASR1000 series (1006, 1004, 1002, 1001)

Impact to troubleshooting and performances

Significant differences in software architecture lead to significant differences in how you troubleshoot the platforms that run classic IOS versus IOS XE. We can break down those differences in few larger areas.

Table 1-3, shows comparison of troubleshooting differences between classic IOS ("IOS) and IOS XE

Troubleshooting Area	Classic IOS ("IOS")	IOS XE
Methodology	All software based platforms use single set of troubleshooting approach	Different set of show/debug platform CLIs for control and data planes
Command Line Interface (CLI) - show and debug commands	All software based platforms use single set of CLIs	Show/Debug CLIs can be specific to a given platform (e.g. ASR1000 versus ISR 4451)
Files and locations	All software based platforms use single set of rules (E.g., crash dump go to internal bootflash by default) Classic IOS image naming conventions	It can be specific to a given platform (e.g. ASR1000 versus ISR 4451) ASR1000 dumps SPA driver cashdump on harddisk: by default Newer software packages naming conventions
Level of difficulty	Straightforward, upside of software-based platforms	Complex, downside or cost of modularity i.e. clear separation of control and data plane software and hardware leads to more involved troubleshooting

Identify Cisco express forwarding concepts

Cisco Express Forwarding (CEF) is advanced, layer 3 IP forwarding technology. CEF optimizes network performance and scalability where networks have large and dynamic traffic patterns, such as the Internet itself.
CEF offers the following benefits:

- Improved performance—CEF is less CPU-intensive than older fast switching. As a result, more CPU processing power can be dedicated to other layer 3 services such as quality of service (QoS) and encryption.
- Scalability—CEF offers full switching capacity at each line card or blade when distributed CEF (dCEF) mode is active.
- Resilience—CEF offers switching consistency and stability in large dynamic networks. In dynamic networks, fast-switched cache entries go through high level of churn and are frequently invalidated due to routing changes. These changes can cause traffic to be process switched using the routing table, rather than fast switched using the route cache. With CEF, Forwarding Information Base (FIB) lookup table contains all known routes that exist in the routing table, it eliminates route cache maintenance and as a result avoids sub optimal forwarding scenarios that takes place with the fast-switch or process switching.

CEF uses a FIB to make IP destination prefix-based switching decisions. The FIB is conceptually similar to a routing table or information base. It maintains a mirror image of the forwarding information contained in the IP routing table. When routing or topology changes occur in the network, the IP routing table is updated, and those changes are reflected in the FIB. The FIB maintains next hop address information based on the information in the IP routing table.

Hardware based switching platforms use Content Addressable Memory (CAM) for storing the CEF related information. These tables are finite and can fill up to exhaustion, which would cause forwarding to fall back to software. Catalyst 4500, as an example, can carry up to 128K entries in Supervisor IV/V CAMs. Once those entries are filled up, it switches to software forwarding with an error message “C4K_L3HWFORWARDING-2-FWDCAMFULL”. You can verify CAM table usage by show platform hardware ip route summary command.
“show mls cef exception status” can be used on Catalyst 6500 to check on FIB TCAM usage. A switch with FIB TCAM full will repeatedly throw an error message like below:

%MLSCEF-DFC4-7-FIB_EXCEPTION: FIB TCAM exception, Some
entries will be software switched

As a result of this TCAM exception condition, connectivity is affected and might result in elevated CPU usage due to software switching.

RIB, FIB, LFIB, Adjacency table

Routing Information Base (RIB)

RIBs (Routing Information Base) maintain the network topologies and routing tables for each protocol. This would include many routes going to the same destination prefix. It is built on per routing protocol basis, so RIP and OSPF have their own copy of RIBs.

Forwarding Information Base (FIB)

FIBs are the best routes from possibly many routing protocols in the RIBs pushed down to fast forwarding lookup memory (or just DRAM for software-based platforms) for the best path(s). This is what you see in show ip route command output. There is one copy of FIB per system for centralized forwarding platforms, or one for each line card in case of distributed systems.

Label Information Base (LIB)

LIB (Label Information Base) is the software table maintained by IP/MPLS capable routers to store the details of port and the corresponding MPLS router label to be popped or pushed on incoming or outgoing MPLS packets respectively. LIB entries are populated from label-distribution protocols. LIB functions in the control plane of Cisco routers. It is used by the label distribution protocol for mapping the next hop labels.

Label forwarding information base (LFIB) is a data structure and way of managing forwarding in which destinations and incoming labels are associated with outgoing interfaces and labels. The forwarding paradigm employed by MPLS is based on the notion of label swapping. When a packet with a label is received by an Label Switching Router (LSR), the switch uses the label as an index in its LFIB to determine the outgoing interface.

Adjacency Tables

Routers or Switches in a network are considered adjacent if they can reach each other with a single hop across a link layer. In addition to the FIB, CEF uses adjacency tables to prepend Layer 2 addressing information such as MAC addresses. The adjacency table maintains Layer 2 next-hop addresses for all FIB entries.

Load balancing Hash

In a router, act of distributing packets across multiple links based on layer 3 routing information is known as load balancing. If a router discovers multiple paths to a destination, the routing table is updated with multiple entries for that destination.
Router> show ip route

[...]

I 192.168.25.0/24 [115/10] via 192.168.24.6

[115/10] via 192.168.24.10

[115/10] via 192.168.24.14

[...]

Usually the paths have the same metric, however there are routing protocols that allow unequal cost (or metric) load balancing. A router learns about the existence of parallel paths through the routing protocols and builds its routing table accordingly.
The number of paths used is limited by the number of entries the given IP routing protocol puts in the routing table, the default in IOS is 4 entries for most routing protocols with the exception of BGP, where it is one entry (only the best path). The maximum number of paths that can be configured are 6. Cisco IOS supports two primary modes of load balancing, i.e. per-destination and per-packet basis.

Per-Destination load balancing

In per-destination mode all packets for a given destination are forwarded along the same path. This preserves packet order however may lead to unequal usage of the links. If one host receives the majority of the traffic all packets will use one link, leaving bandwidth on other links unused. This is the default load balancing mode in IOS using universal algorithm. Original per-destination algorithm creates a 4-bit hash of the source and destination IP address and load balances based on this 16 (2^4) value hash.

Per-Packet load balancing

Per-packet load balancing guarantees equal load across all links however packets may arrive out-of-order at the destination as differential delay may exist for each link used. In particular,

per-packet load balancing can result in unsatisfactory data transmission for video and voice streaming.

```
Router(config)#ip cef load-sharing algorithm ?
 include-ports  Algorithm that includes Layer 4 ports
 original       Original algorithm
 tunnel         Algorithm for use in tunnel only environments
 universal      Algorithm for use in most environments
```

The following load-balancing algorithms are provided for use with Cisco Express Forwarding traffic.

- Original algorithm—The original Cisco Express Forwarding load-balancing algorithm produces distortions in load sharing across multiple routers because the same algorithm was used on every router. Depending on your network environment, you should select either the universal algorithm (default) or the tunnel algorithm instead.
- Universal algorithm—The universal load-balancing algorithm allows each router on the network to make a different load sharing decision for each source-destination address pair, which resolves load-sharing imbalances. The router is set to perform universal load sharing by default.
- Tunnel algorithm—The tunnel algorithm is designed to balance the per-packet load when only a few source and destination pairs are involved.
- Include-ports algorithm—The include-ports algorithm allows you to use the Layer 4 source and destination ports as part of the load-balancing decision. This method benefits traffic streams running over equal cost paths that are not load shared because the majority of the traffic is between peer addresses that use different port numbers, such as Real-Time Protocol (RTP) streams.

Polarization concept and avoidance

CEF polarization occurs when traffic uses per destination load balancing and the same algorithm, which is default, is used throughout the network which causes traffic to not be load balanced after the first distribution.

As an example think of a layer 3 network with multiple layers or levels each with a possible path to the right or left. If 100Mbps of traffic was coming into a router, it would be load balanced 50/50, with 50Mbps to Router-right and 50Mbps to Router-left, but as Router-level-1-right & Router-level1-left will use the same algorithm to determine which path the traffic will take, but as the algorithm is identical it will be a 100/0 split, with 50Mbps going to Router-level2-right and

Router-level2-left and no data going to other paths. Whenever there is an even number of ECMP available, traffic will not be distributed evenly.

To counter this issue, a newer algorithm called the universal algorithm was developed where a 32-bit value is added to the hashing algorithm, this value can be manually set but defaults to the highest loopback IP on the router. This is based on the concept called unique-ID/universal-ID. Hash function is known as universal-ID, a randomly generated value at the time of the router or layer 3 switch boot up that can be manually controlled. This seeds the hash function on each router with a unique ID, which ensures that the same source/destination pair hash into a different value on different routers along the path within the network. This process provides a better network-wide load-sharing and avoids the polarization issue. In order to configure a custom ID, you can use the following CLI:
Router(config)#ip cef load-sharing algorithm universal <id>

Another way to avoid polarization would be to use alternate between default (Source IP and Destination IP) and full (Source IP + Destination IP + Layer 4 ports) hashing inputs configuration at each layer of the network. Of course, this is not practical if we're talking about a large network with many layers some possibly outside the control of the given network administrator.

Explain general network challenges

Unicast flooding

Unicast flood is the unintentional behavior of a switch treating a unicast packet as a broadcast packet; a packet destined for one host is flooded or transmitted out of all the ports of a switch. The underlying cause of flooding is that destination MAC address of the packet is not in the L2 forwarding table (there is one for each VLAN) of the switch.
The primary reasons for unicast flooding behavior include asymmetric routing, STP topology changes (i.e. repeated TCNs), and MAC forwarding table overflow.

Asymmetric Routing

When communication between two hosts (or endpoints of any type) take different paths on their way out and another on their way in, it is called asymmetric routing. It can also cause packets to arrive out of order if packets that are part of a given flow take different paths.
Large amounts of flooded traffic might saturate low-bandwidth links causing network performance issues or complete connectivity outage to devices connected across such low-bandwidth links. An example of such situation could be a topology where there are two switches

(ports in two VLANs, say A and B), two routers (doing inter-VLAN routing between A and B) and two hosts one in VLAN A and one in VLAN B. Now since the routers will proxy ARP for respective hosts as they are default gateways, switches will never be able to learn actual end hosts MAC addresses (router will rewrite them every single time to their own). Switch A and B will continue to flood traffic since they are unaware of the actual host A and host B MAC addresses.
The solution approach is normally to bring the router's ARP timeout and the switch's' forwarding table-aging time close to each other. This will cause the ARP packets to be broadcast, relearning must occur before the L2 forwarding table entry ages out.

Spanning-Tree Protocol Topology Changes

TCNs are triggered by a port that is transitioning to or from the forwarding state. After the TCN, even if the particular destination MAC address has aged out, flooding should not happen for long in most cases since the address will be relearned. The issue might arise when TCNs are occurring repeatedly within short period of time. The switches will constantly be fast-aging their forwarding tables so flooding will be nearly constant.
Typically, a TCN is rare occurrence in a well-configured network. As said before, when the port on a switch goes up or down, there is eventually a TCN once the STP state of the port is changing to or from forwarding. However, when a port is flapping, repetitive TCNs and flooding occurs.
The solution approach would be to configure ports with the STP portfast feature and avoid TCNs when going to or from the forwarding state. Configuration of portfast on all end-device ports (such as printers, PCs, servers, and so on) should limit TCNs to a low amount.

Forwarding Table Overflow

As mentioned before, another possible but not so common cause of flooding can be overflow of the switch forwarding table. In this case, new addresses cannot be learned and packets destined to such addresses are flooded until some space becomes available in the forwarding table. New addresses will then be learned. Since most modern switches have large enough forwarding tables to accommodate MAC addresses for most designs, L2 table overflows are uncommon.

Out of order packets

Using per-packet load balancing to share the traffic load across available paths to a given destination can lead to out-of-order packets for a given data flow.

Impact of micro burst

Micro-bursting is a phenomenon where rapid bursts of data packets are sent in quick succession, leading to periods of full line-rate transmission that can overflow packet buffers of the network stack, both in network endpoints and routers and switches inside the network. Symptoms of micro bursts will manifest in the form of ignores and/or overruns (also shown as accumulated in “input error” counter within show interface output). This is indicative of receive ring and corresponding packet buffer being overwhelmed due to data bursts coming in over extremely short period (microseconds). You will never see a sustained data traffic within show interface’s “input rate” counter as they are averaging bits per second (bps) over 5 minutes by default (way too long to account for microbursts). You can understand microbursts from a scenario where a 3-lane highway merging into a single lane at rush hour – the capacity burst cannot exceed the total available bandwidth (i.e. single lane), but it can saturate it for a period of time.
In order to troubleshoot microbursts, you need a packet sniffer that can capture traffic over a long period of time and allow you to analyze it in the form of a graph which displays the saturation points (packet rate during microbursts versus total available bandwidth). You can eventually trace it to the source causing the bursts (e.g. stock trading applications). You can implement large packet buffers to avoid or mitigate microbursts.

Explain IP operations

ICMP unreachable, redirect

If a router or a layer 3 switch receives a non-broadcast packet destined for itself that uses an unknown protocol, it sends an ICMP protocol unreachable message back to the source. Similarly, if the software receives a packet that it is unable to deliver to the ultimate destination because it knows of no route to the destination address, it will send an ICMP host unreachable message to the source. This feature is enabled by default.

You can disable it by using the following CLI:
Router(config-if)# no ip unreachables

IPv4 options, IPv6 extension headers

The possible options that can be put in the IPv4 header are as follows:

Table 1-5, shows IP header options and their description

Field	Size (bits)	Description
Copied	1	Set to 1 if the options need to be copied into all fragments of a fragmented packet.
Option Class	2	A general options category. 0 is for "control" options, and 2 is for "debugging and measurement". 1, and 3 are reserved.
Option Number	5	Specifies an option
Option Length	8	Indicates the size of the entire option (including this field). This field may not exist for simple options
Option Data	Variable	Option-specific data. This field may not exist for simple options

IPv6 uses two distinct types of headers:

- Main/Regular IPv6 Header
- IPv6 Extension Headers

The main IPv6 header is equivalent to the basic IPv4 one despite some field differences that are the result of lessons learned from operating IPv4.

Table 1-6, IPv6 Extension Headers and their Recommended Order in a Packet

Order	Header Type	Next Header Code
1	Basic IPv6 Header	-
2	Hop-by-Hop Options	0
3	Destination Options (with Routing Options)	60
4	Routing Header	43
5	Fragment Header	44
6	Authentication Header	51
7	Encapsulation Security Payload Header	50
8	Destination Options	60
9	Mobility Header	135
	No next header	59

Upper Layer	TCP	6
Upper Layer	UDP	17
Upper Layer	ICMPv6	58

Extension headers are an intrinsic part of the IPv6 protocol and they support some basic functions and certain services. The following is a list of situations where EHs are commonly used:

- Hop-by-Hop EH is used for the support of Jumbo-grams or, with the Router Alert option, it is an integral part in the operation of Multicast Listener Discovery (MLD). Router Alert is an integral part in the operations of IPv6 Multicast through MLD) and RSVP for IPv6.
- Destination EH is used in IPv6 Mobility as well as support of certain applications.
- Routing EH is used in IPv6 Mobility and in Source Routing. It may be necessary to disable ipv6 source routing using ipv6 source-route command on routers to protect against DDoS.
- Fragmentation EH is critical in support of communication using fragmented packets (in IPv6, the traffic source must do fragmentation-routers do not perform fragmentation of the packets they forward)
- Mobility EH is used in support of Mobile IPv6 service
- Authentication EH is similar in format and use to the IPv4 authentication header
- Encapsulating Security Payload EH is similar in format and use to the IPv4 ESP header. All information following the Encapsulating Security Header (ESH) is encrypted and obfuscated and for that reason, it is invisible to intermediary network devices.

IPv4 and IPv6 fragmentation

IP implements datagram fragmentation, breaking it into smaller pieces, so that packets can pass through a link with a smaller maximum transmission unit (MTU) than the original datagram size. The Identification field, and Fragment offset field along with Don't Fragment (DF) and More Fragment (MF) flags in the IP protocol header are used for fragmentation and reassembly of IP datagrams.

The details of the fragmentation mechanism, as well as the overall architectural approach to packet fragmentation, are different between IPv4 and IPv6. In IPv4, routers perform fragmentation, whereas in IPv6, routers do not fragment, but drop the packets that are larger than the MTU. While the header formats are different for IPv4 and IPv6, analogous fields are used for fragmentation, so the algorithm can be reused for fragmentation and reassembly.

In IPv4, hosts must make a best-effort attempt to reassemble fragmented IP datagrams with a total reassembled size of up to 576 bytes - equal to the minimum MTU for IPv4. They may also attempt to reassemble fragmented IP datagrams larger than 576 bytes. In IPv6, this minimum MTU is increased to 1,280 bytes larger than the minimum MTU for IPv4. IPv6 fragment header and identification header are 64 bits and 32 bits long respectively.

TTL

Time to live (TTL) or hop limit is a mechanism that limits the lifespan or lifetime of data in a network. TTL prevents a data packet from circulating indefinitely. If the TTL field reaches zero before the datagram arrives at its destination, then the datagram is discarded and an ICMP error datagram (11 - Time Exceeded) is sent back to the sender.

IP MTU

IPv4 allows fragmentation: dividing the datagram into pieces, each small enough to pass over the single link that is being fragmented for, using the MTU parameter configured for that interface. This fragmentation process takes place at the IP layer (OSI layer 3) and marks packets it fragments as such, so that the IP layer of the destination host knows it should reassemble the packets into the original datagram. This method implies a number of possible drawbacks:

- All fragments of a packet must arrive for the packet to be considered received. If the network drops any fragment, the entire packet is lost.
- When the size of most or all packets exceed the MTU of a particular link that has to carry those packets, almost everything has to be fragmented. In certain cases the overhead this causes can be considered unreasonable or unnecessary. For example, various tunneling situations cross the MTU by very little as they add just a header's worth of data. The addition is small, but each packet now has to be sent in two fragments, the second of which carries very little payload. The same amount of payload is being moved, but every intermediate router has to do double the work in terms of header parsing and routing decisions.
- As it is normal to maximize the payload in every fragment, any further fragmentation that turns out to be necessary will increase the overhead even more.
- There is no simple method to discover the MTU of links beyond a node's direct peers.
- The Internet Protocol requires that hosts must be able to process IP datagrams of at least 576 bytes (for IPv4) or 1,280 bytes (for IPv6). However, this does not preclude Data Link Layers with an MTU smaller than IP's minimum MTU from conveying IP data. For example, according to IPv6's specification, if a particular Data Link Layer physically

cannot deliver an IP datagram of 1,280 bytes in a single frame, then the link layer must provide its own fragmentation and reassembly mechanism, separate from IP's own fragmentation mechanism, to ensure that a 1280-byte IP datagram can be delivered, intact, to the IP layer.

Explain TCP operations

IPv4 and IPv6 PMTU

TCP Maximum Segment Size (MSS) defines the maximum amount of data that a host is willing to accept in a single TCP/IP datagram, it takes care of fragmentation at the two endpoints of a TCP connection, however it doesn't handle the case where there is a smaller MTU link in the middle between these two endpoints. PMTUD was developed to avoid fragmentation in the path between the endpoints. It is used to dynamically determine the lowest MTU along the path from a packet's source to its destination. PMTUD is only supported by TCP. If PMTUD is enabled on a host, and it almost always is, all TCP/IP packets from the host will have the DF bit set.
When a host sends a full MSS data packet with the DF bit set, PMTUD works by reducing the send MSS value for the connection if it receives information that the packet would require fragmentation. A host usually "remembers" the MTU value for a destination by creating a "host" (/32) entry in its routing table with this MTU value.
If a router tries to forward an IP datagram, with the DF bit set, onto a link that has a lower MTU than the size of the packet, the router will drop the packet and return an Internet Control Message Protocol (ICMP) "Destination Unreachable" message to the source of this IP datagram, with the code indicating "fragmentation needed and DF set" (type 3, code 4). When the source station receives the ICMP message, it will lower the send MSS, and when TCP retransmits the segment, it will use the smaller segment size.
There are three things that can break PMTUD, two of which are uncommon and one of which is common.

- A router can drop a packet and not send an ICMP message.
- A router can generate and send an ICMP message but the ICMP message gets blocked by a router or firewall between this router and the sender. (Common)
- A router can generate and send an ICMP message, but the sender ignores the message.
- When path MTU discovery fails, it results in application slowdowns and timeouts since intermediary devices may be doing both fragmentation and reassembly.

Latency

Maximum achievable throughput for a single TCP connection is determined by different factors. One trivial limitation is the maximum bandwidth of the slowest link in the path. But there are also others, less obvious limits for TCP throughput. Bit errors can create a limitation for the connection as well as round-trip time.

Windowing

Flow control allows us to deal with a situation when a sending computer tries to transmit information at a faster rate than the destination computer can receive and process it. This can happen if the receiving computers have a heavy traffic load in comparison to the sending computer, or if the receiving computer has less processing power than the sending computer. Sliding-window flow control is best utilized when the buffer size is limited and pre-established. During a typical communication between a sender and a receiver the receiver allocates buffer space for n frames (n is the buffer size in frames). The sender can send and the receiver can accept n frames without having to wait for an acknowledgement. The receiver acknowledges a frame by sending an acknowledgement that includes the sequence number of the next frame expected. This acknowledgement announces that the receiver is ready to receive n frames, beginning with the number specified. Both the sender and receiver maintain what is called a window. The size of the window is less than or equal to the buffer size.
Sliding window flow control has a far better performance than stop-and-wait flow control. For example in a wireless environment data rates are low and noise level is very high, so waiting for an acknowledgement for every packet that is transferred is not very feasible. Therefore, transferring data as a bulk would yield a better performance in terms of higher throughput.

Bandwidth delay product

Bandwidth-delay product refers to the product (or multiplication) of a data link's capacity (in bits per second) and its end-to-end delay (in seconds). The result, an amount of data measured in bits (or bytes), is equivalent to the maximum amount of data on the network circuit at any given time, i.e., data that has been transmitted but not yet acknowledged.
In wide area networks (WANs), where the end-to-end delay-bandwidth product becomes a significant factor, TCP uses the network buffers to sustain a steady-state throughput that matches the available network capacity. TCP receive window size can become limiting factor for links with larger bandwidth and smaller delay.

Global synchronization

Global synchronization happens when multiple TCP hosts reduce their transmission rates in response to packet dropping, then increase their transmission rates once again when the congestion is reduced.
WRED avoids the globalization problems that occur when tail drop is used as the congestion avoidance mechanism on the router.