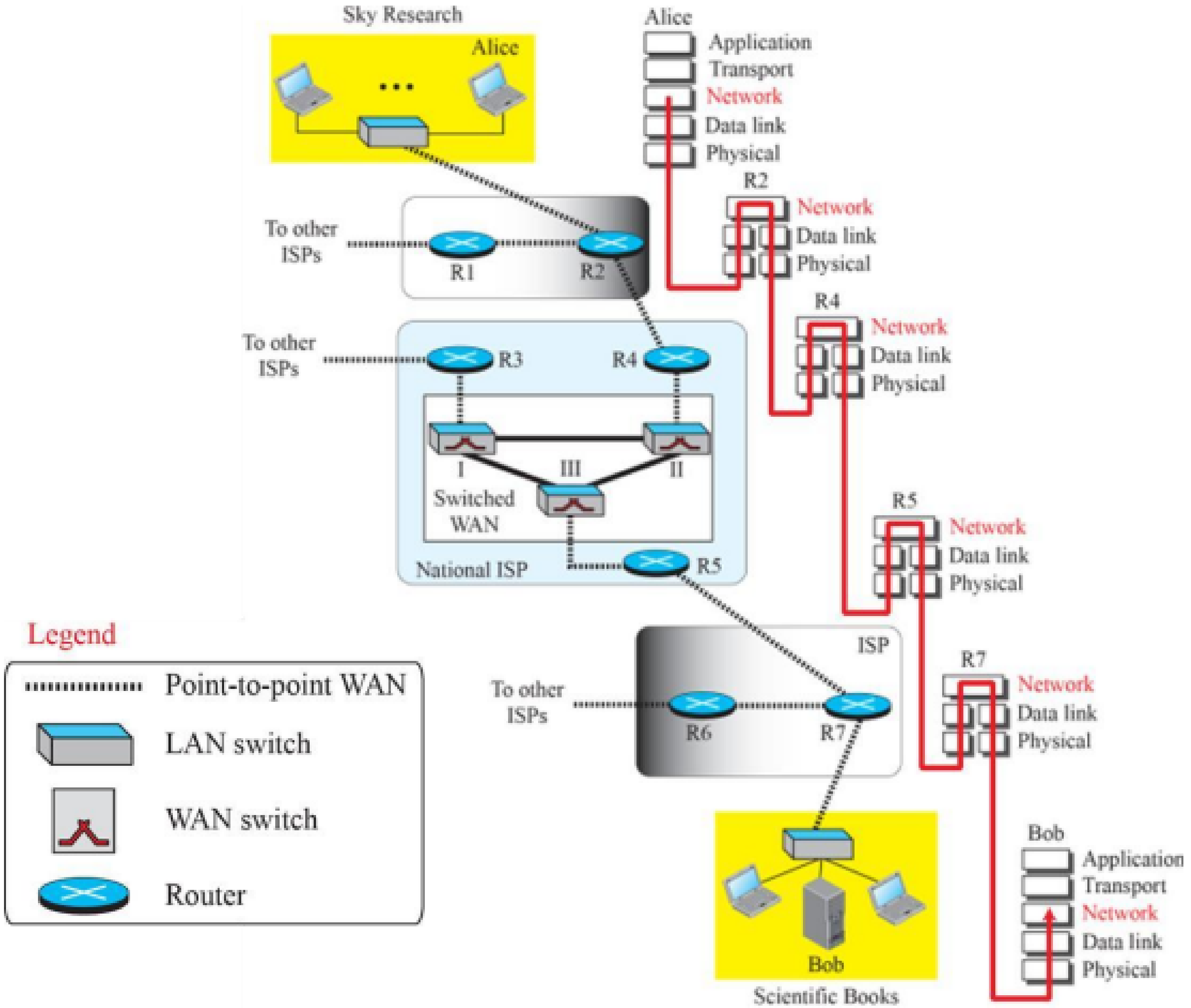
Chapter 4Networklayer

Objective

- We first discuss services that can be provided at the networklayer:packetizing, routing, and forwarding.
- We then discuss the network layer at the TCP/IP suite: IPv4andICMPv4. We also discuss IPv4 addressing and relatedissues.
- We then concentrate on the unicast routing and unicast routingprotocols.
- We then move to multicasting and multicast routing protocolsprotocol.



Sky Research
Alice
To other ISPs
R1
R2
To other ISPs
R3
R4
I
II
III
Switched WAN
National ISP
R5
ISP
To other ISPs
R6
R7
Bob
Scientific Books
Legend
Point-to-point WAN
LAN switch
WAN switch
Router
Alice
Application
Transport
Network
Data link
Physical
R2
Network
Data link
Physical
R4
Network
Data link
Physical
R5
Network
Data link
Physical
R7
Network
Data link
Physical
Bob
Application
Transport
Network
Data link
Physical

Network-Layer Services

Before discussing the network layer in the Internet today, let’s briefly discuss the network-layer services that, in general, are expected from a network-layer protocol.

Before discussing the network layer in the Internet today, let’sbrieflydiscuss the network-layer services that, in general, are expectedfromanetwork-layer protocol.

- Packetizing
- Routing
- Forwarding

Packetizing

The network layer has an important job. It takes data fromone place andsendsit to another without altering or using the data. It's like a postal servicethat delivers packages without changing their contents. Here's how it works:

- The source computer gets data from a higher-level programand puts it inanetwork packet. **This packet includes sender and receiver information**andother details needed for the network.
- The source computer then sends the packet to the data-link layer for delivery. **It can't change the data unless it's too big**, and it needs to split it intosmallerparts.
- The destination computer gets the network packet fromits data-link layer. Ittakes out the data and gives it to the right program. **If the packet was split intoparts, the network layer waits until all parts arrive, puts themtogether**, andthengives the data to the program.
- Routers in between don't open the packets unless they need to

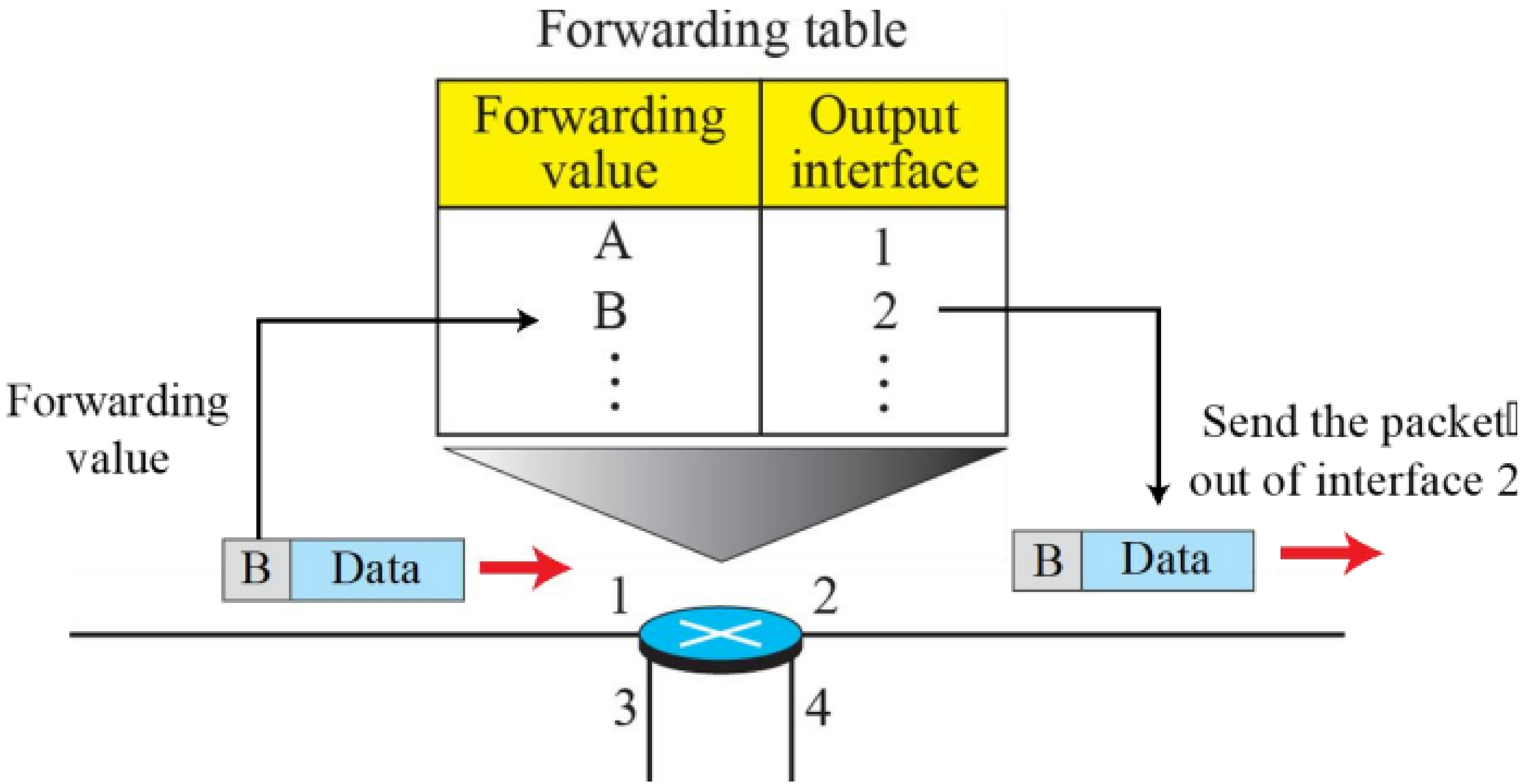
split theminto **smaller parts**. They also don't change the sender and receiver info. Theyonly look at the info to know where to send the packet next. If a packet is split, they copy the header to all parts and make some changes as needed.

Routing

The network layer has another important job called "routing."Thismeans it figures out **the best path for data to travel** fromwhereitstarts to where it needs to go. Think of it like finding thefastestwayto get somewhere on a road trip. In a computer network, there are different paths datacantake,likedifferent roads on a map. The network layer's job is topickthebest road for the data to travel on. To do this, the network layer has special plans, like GPS for data. **Itusessomething called "routing protocols" to help routers (liketrafficcopsfor data) know which path to choose**. These plans are set upbefore any data starts moving.

Forwarding

Think of routing like making a plan and using special rules todecidehow data should travel through a network. Now, **forwardingiswhat each router does when data actually arrives at its doorstep**. When data arrives at a router, it needs to know**where tosenditnext**.Imagine it's like a traffic cop directing cars at an intersection. **Therouter looks at the data's address (or a special tag) andchecksalist(sometimes called a forwarding table, or routing table) tofigureout which way to send it**. So, routing is the plan-making part, and forwarding is theactionpartwhere the router decides where data goes based on that plan. It'slike the router is following a map to guide the data to its destination.



Error control

In Chapter 3, we talked about how we handle mistakes in data transferatthetransport layer. In Chapter 5, we'll dive into how we deal with errorsat thedata-link layer. Now, even though we can also handle errors at thenetworklayer, **the folks who designed the network layer for the Internet didn't focus on it much**. One big reason is that data in the network layer can get brokenupintopieces at different points along its journey, like **splitting a puzzleintoparts. Checking for errors in this situation is not very efficient.** However, these designers **did add a special "checksum" field**totheheaderof the data packet. It's like a security seal that makes sure the header isn't tampered with during its journey from one place to another. It's important to note that even though the Internet's network layer doesn'tdirectly fix errors, it has a helper **called ICMP. This protocol helps out if adata packet is thrown**

away or has weird stuff in its header. We'll get into more detail about ICMP later in this chapter.

Flow control

Flow control is like managing how much data a sender computer can send toareceivercomputer without causing a traffic jam. If the sender's computer is making datafasterthan the receiver's computer can handle, it's like pouring too much water into a glass—it overflows. To avoid this, the receiver needs to let the sender know when it's getting overwhelmed with data. Interestingly, the network layer in the Internet doesn't directly handle this flowcontrol job.Instead, it lets the sender send data whenever it's ready, without worrying about whether the receiver can keep up. There are a few reasons for this approach.

First, **the receiver's job at the network layer is pretty simple** because it doesn't haveto deal with error checking, so it's less likely to get overwhelmed. Second, the higher-level programs using the network layer can **manage the incomingdata by storing it temporarily and don't need to consume it all at once.** Third, most of these higher-level programs **have their own flow control**, so addingmorecontrol at the network layer would make things more complex and less efficient.

Congestion control

Another thing to consider in network-layer protocols is congestioncontrol.Congestion happens when there are too many data packets inonepartofthe Internet. It's like a traffic jam on the information highway. This canoccurif computers send more data than the network or routers canhandle. Whenthis occurs, some packets might get dropped by routers to easethetraffic.

But here's the tricky part: **if too many packets get dropped, it canmakethings worse. That's because higher-level error checks might makethesender resend the lost packets, causing even more congestion. Inextremecases, everything**

can grind to a halt, and no data can get through.

We'll go into more detail about how to deal with congestioninthenetworklayer later in this chapter, even though the Internet doesn't directlyhandleit.

Quality of service

With the Internet, we can now do cool things like videochatsandstreaming music. But for these things to work smoothly, weneedtomake sure the quality of our communication is really good. TheInternet has gotten better at this by improving the qualityof service(QoS), which is like making sure your online experience is top-notch.

However, **most of these quality improvements happeninthehigherlayers of the Internet, not the network layer.** The networklayerremains mostly the same.

Security

Another thing to think about in network communicationis safety. When the Internet first started, it wasn't a big deal becauseonlyafewpeople at universities used it for research, and outsiders couldn'taccess it. So, they didn't worry much about security inthenetworklayer.

But now, security is a big deal**. To make the network layer moresecure, they've created something called IPSec**, whichadds alayerofprotection to the network.

Packet switching

how data moves around in a network. Think of it like a switchthat connectsthings, like how an electrical switch connects power. In networking, therearetwo ways to do this: **circuit switching and packet**

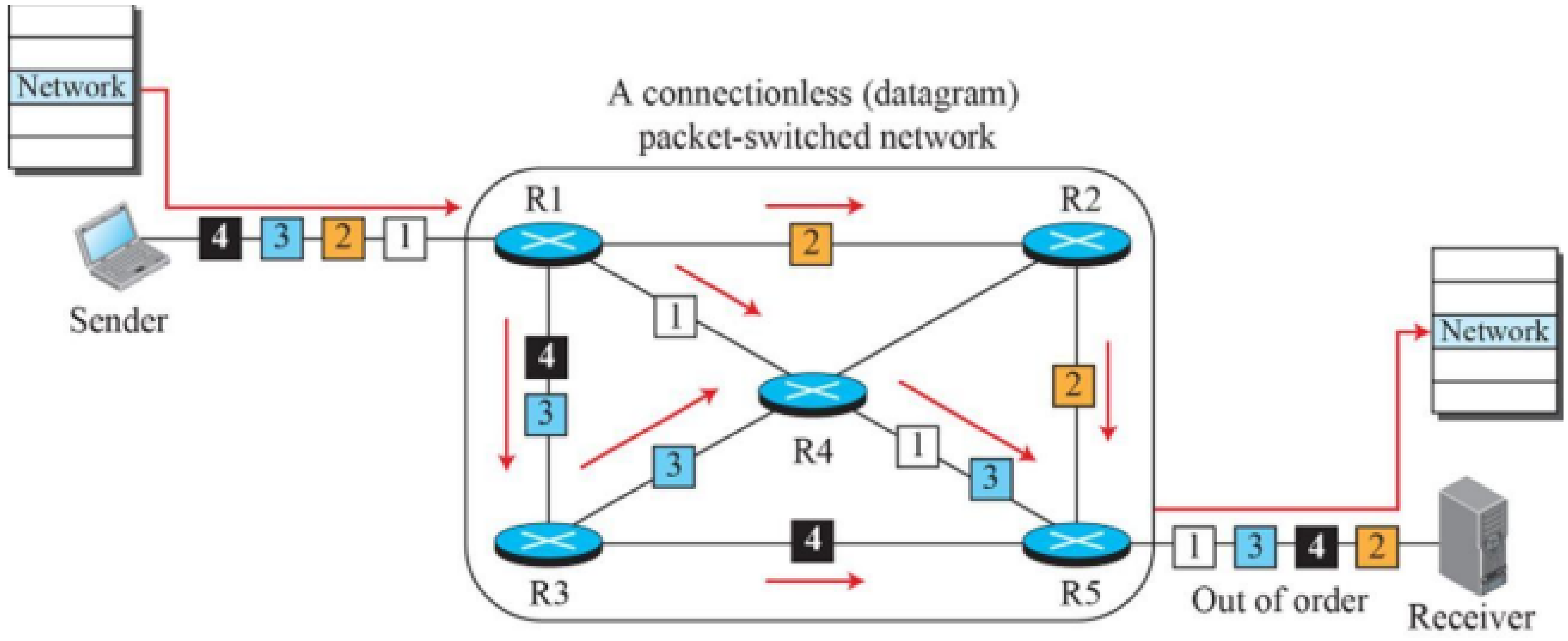
switching. We**usepacketswitching in the network layer because data is sent in packets here.** Circuit switching is used at a lower level. In the network layer, a message is split into packets and sent throughthenetwork. The sender sends them one by one, and the receiver getsthemoneby one. **The receiver waits for all the packets fromthe samemessagebefore passing it to the next level.** In a packet-switched network, devicesstill need to figure out how to send the packets to the right place. Thereare two ways to do this:

- the datagram way
- the virtual circuit way

Datagram Approach: ConnectionlessService

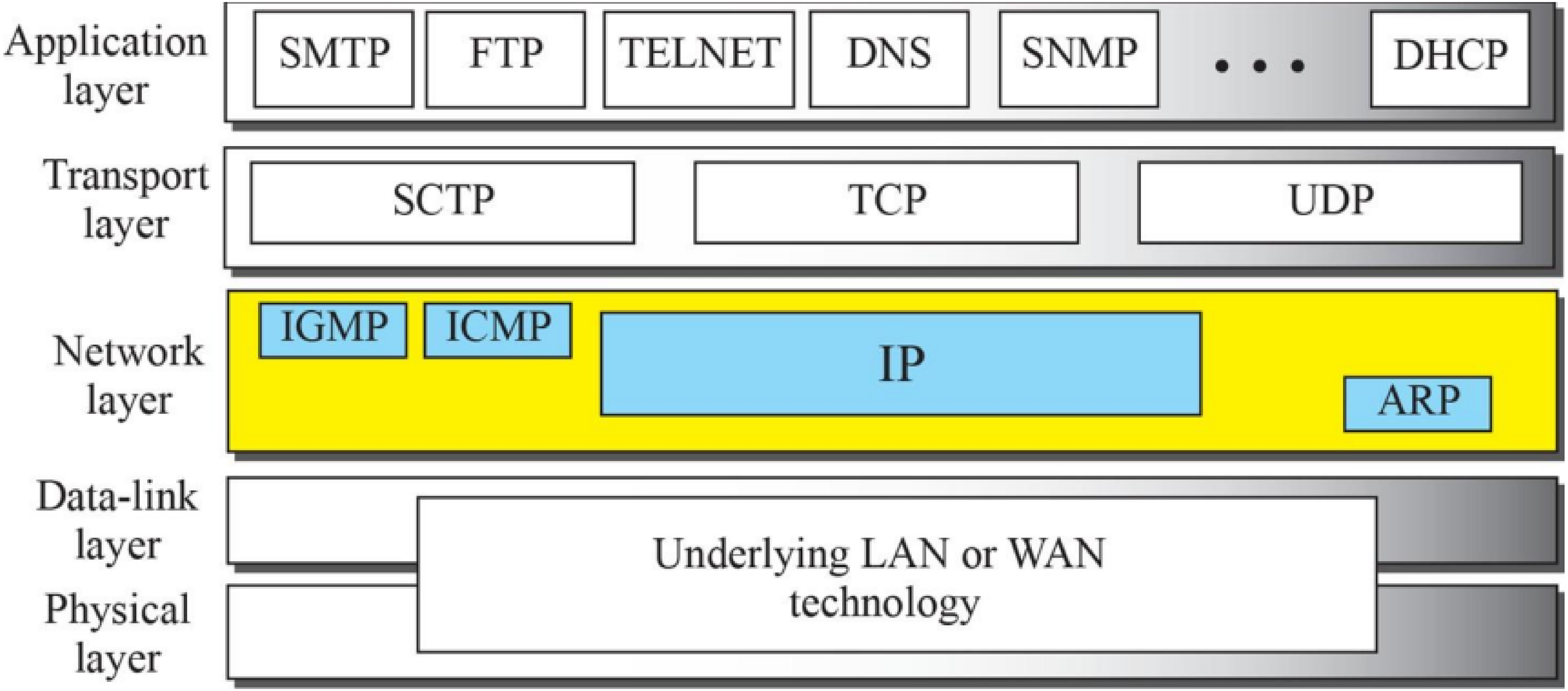
- When the Internet first started, the network layer was designedtokeep things simple. It offered a service where each packet of datawastreated separately, with no connection or relationshipbetweenthem.**The network layer's job was just to get packets fromoneplacetoanother**, and they could take different paths.
- Imagine each packet as its own independent traveler intheInternetjourney. **They don't know each other, and they don't havetofollowthe same route**. Routers, which are like traffic cops for Internetdata,help these packets find their way. They decide where tosendeachpacket based on the destination address written on it.
- **The source address is there to help send an error messagebacktothe sender if something goes wrong.** So, it's like eachpacket hasadestination it's trying to reach, and routers help themget there. Thinkof it like sending

letters to different places



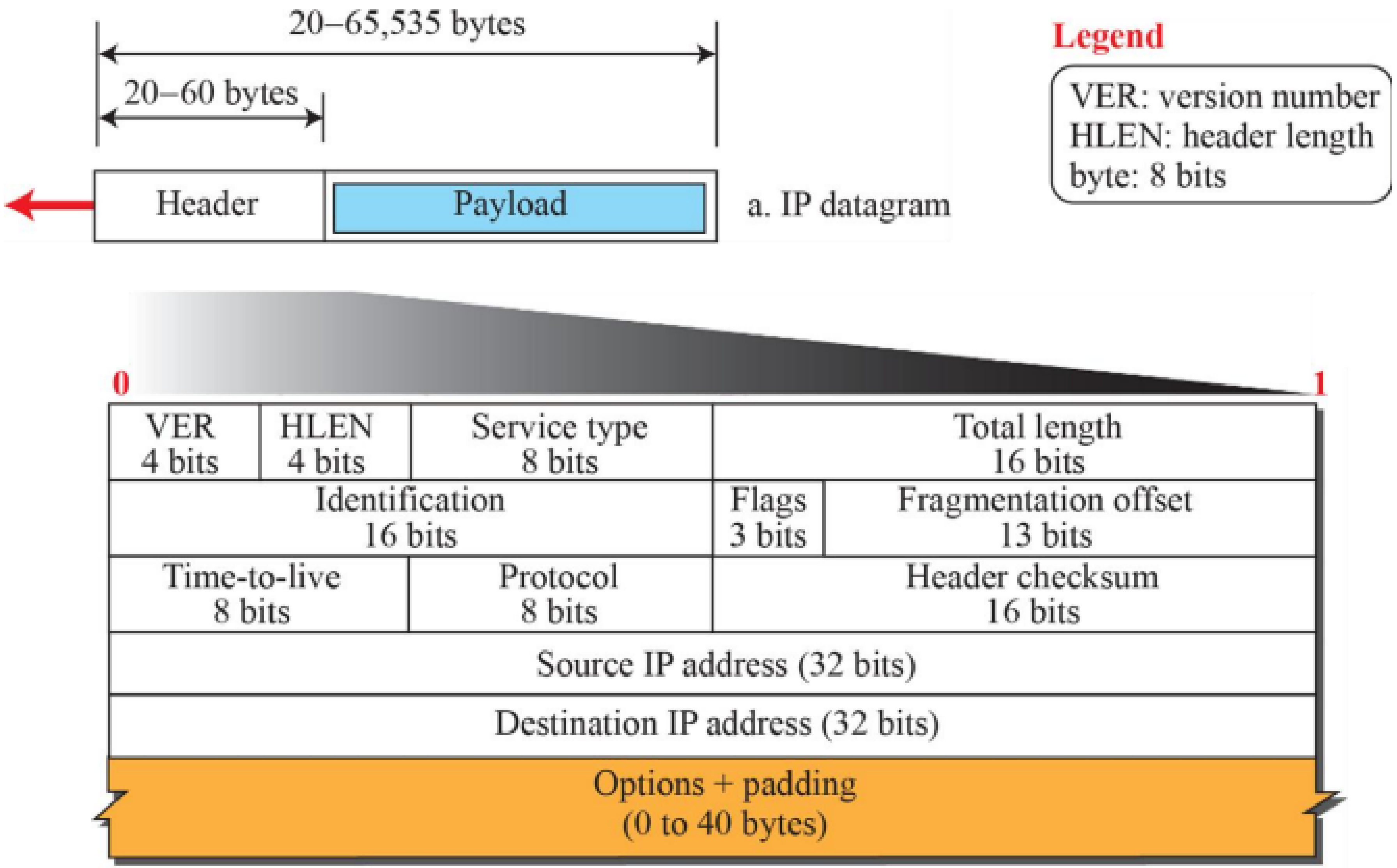
NETWORK-LAYER PROTOCOLS

- In this section, we show how the network layer is implementedintheTCP/IP protocol suite. The protocols in the network layer havegonethrough several versions; in this section, we concentrateonthecurrent version (4), in the last section of this chapter, webrieflydiscuss version 6, which is on the horizon.



IPv4 DatagramFormat

- the IPv4 datagram format. A datagram is a variable-lengthpacketconsisting of two parts: **header and payload (data**). Theheaderis20to 60 bytes in length and contains information essential toroutingand delivery. It is customary in TCP/IP to showthe header in4-bytesections



b. Header format

- **Version Number**

The 4-bit version number (VER) field defines the versionof theIPv4protocol, which, obviously, has the value of 4. • **Header Length (HLEN)**

In IPv4 is a 4-bit field that tells us how long the header of adatagramisin terms of 4-byte words. **Since the header's length canvary, thisfieldhelps devices figure out where the header ends and theactual databegins.** To fit this length into 4 bits, we calculate it in 4-bytewords. Toget the actual length in bytes, you need to multiply the valueinthisfield by 4.

- **Service Type**

Used to be called Type of Service (TOS) in the IP header, andit determined how to treat the datagram. In the late 1990s, theIETFchanged it to provide Differentiated Services (DiffServ). (PPPDTRCO)

- **Total Length**:

This 16-bit field tells us the size of the entire IP datagramin bytes, includingboth the header and data. It can go up to 65,535 bytes, but most datagramsare smaller. It helps the receiving device know when the entirepackethas arrived. • **Identification, Flags, and Fragmentation** Offset: These three fields are connected to breaking up large IP datagramsintosmaller pieces when the network can't handle their size. We'll explainthese fields in more detail in the next section when we talk about fragmentation. • **Time-to-live**: To prevent data packets from endlessly travelingthroughtheinternet due to routing issues, the Time-to-live (TTL) field is used. It limitsthe number of routers a packet can pass through. The sourcesetsaninitialvalue, typically twice the maximum number of hops betweenhosts. Eachrouter reduces this value by one. If it reaches zero, the router dropsthe packet.

- The "**Protocol**" field in TCP/IP identifies which type of dataiscarriedin a packet. It uses an 8-bit number assigned to eachprotocol, helpingdevices know how to handle the packet's content. This fieldworkslikea label, ensuring the right protocol deals with the packet, similartohow port numbers operate at the transport layer.

- **Header Checksum** in IP ensures the accuracy of the header butnotthe payload. Errors in the IP header can lead to misdirectionorfragmentation issues. This 16-bit checksumis recalculatedat eachrouter to account for changes in certain fields like TTL. Checksumcalculation is discussed in detail in Chapter 5 for error detection.

- **Source and Destination Addresses**: These 32-bit fields in anIPheaderspecify the

source and destination IP addresses. The source host knowsitsIP address, while the destination address is provided by the protocol usingIP or through DNS. These addresses must stay unchanged duringthedatagram's journey. We'll dive deeper into IP addresses later inthis chapter.

• **Options**: An IP datagram header can hold up to 40 bytes of optional information, useful for network testing and debugging. Whilenot mandatory, all IP implementations must handle options if present. Havingoptions can affect datagram handling and may require routers torecalculate the header checksum. We discuss one-byte andmulti-byte options in more detail on the book's website.

• **Payload**: The payload, or data, is the primary content of a datagram. It'sthe information sent by other protocols using IP. Think of a datagramlikeapackage: the payload is the package's content, and the header is likethe label on the package.

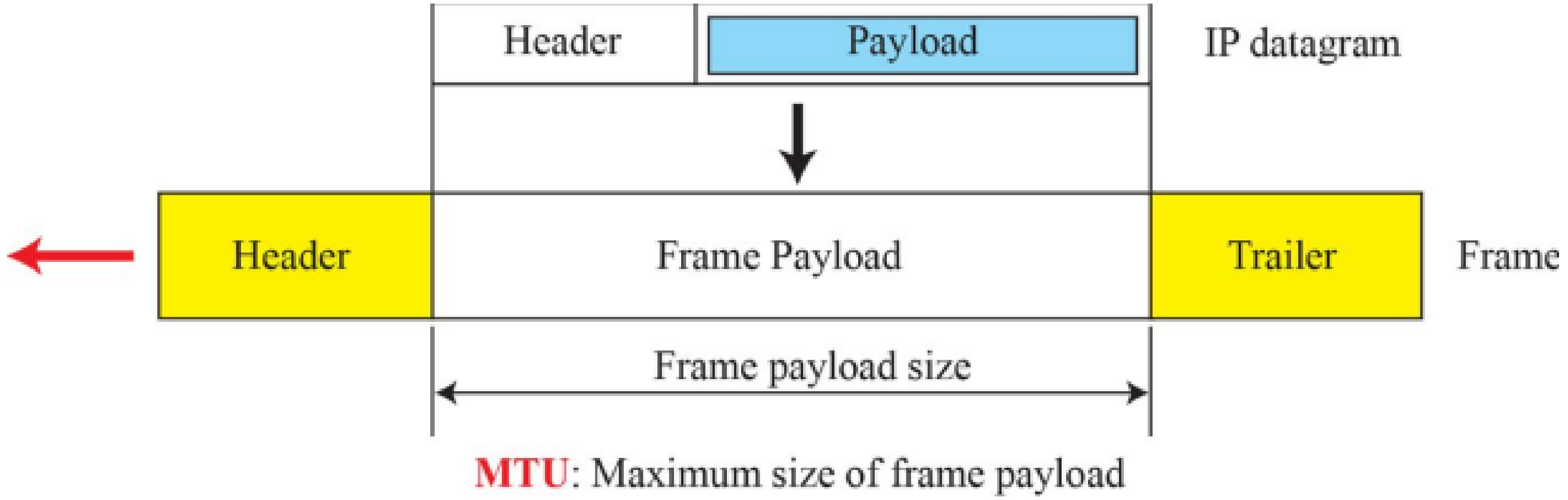
Fragmentation

- A datagram can move through various networks. Whenaroutergetsa datagram, it takes out the IP data and works on it. Then, it putsitinto a new frame. The frame's look and size are different basedonthenetwork it's on. If a router links a local network (LAN) toawide-areanetwork (WAN), it gets a LAN-style frame and sends out aWAN-styleframe.

Maximum Transfer Unit (MTU)

- Different networks use their own rules for packaging data. Eachnetwork type has a limit on how big a piece of data, calledapayload,can be put into its package. This limit is called the

MaximumTransmission Unit (MTU), and it varies depending onthetypeofnetwork. For example, a local network can handle a payloadof upto1500 bytes, but a wide-area network might allowmoreor less.



- The Internet Protocol (IP) allows data packets to be as large as 65,535bytes, which is good for future networks. However, in current networks withsmallerlimits (MTUs), these large packets must be broken into smaller parts, called"fragmentation." If a network with an even smaller limit is encountered, furthersplitting may occur. This means one data packet can be split multipletimesbefore reaching its destination.

- This splitting can be done by the source computer or any router alongtheway.However, the task of reassembling these pieces into the original packet isdoneonly by the destination computer. Each piece acts as an independent packetasit travels through the network.
- Even though fragmented packets can take different paths, they are designedtoeventually reach the destination. It's more efficient to reassemble themat the end, rather than during the journey, to avoid unnecessary complexity.
- Fragmentation mainly involves breaking up the payload of an IP datapacket. Most header information, except for a few options, must be copiedineachfragment. The device doing the fragmentation needs to adjust threefields:

flags,fragmentation offset, and total length. Everything else in the header remainsthesame. Also, the checksum value must be recalculated, regardless of howmany times the data is split.

Fields Related to Fragmentation

- **Identification field**

The 16-bit ID field in a datagram helps identify where it comesfrom.Toensure each datagram is unique, the IP protocol uses a counter. Thiscounter starts at a positive number and is increased by oneforeachdatagram sent. As long as this counter stays in the computer'smemory, uniqueness is guaranteed. When a datagram gets split into smaller parts (fragments), all of themget the same ID number as the original datagram. This IDnumberhelpsthe destination computer put the fragments back together. It knowsthat all fragments with the same ID should be assembledintoone datagram.

- **Flag field**

The 3-bit flags field has three flags. The first one isn't used. Thesecondflag, called the "**do not fragment**" bit (D bit), when set to1, meansthedatagram shouldn't be broken into pieces. If it can't fit throughanetwork, it's discarded, and an error message is sent back. WhentheDbit is set to 0, the datagram can be split into smaller parts if needed.

The third flag, called the "**more fragment**" bit (Mbit), is set to1whenthere are more fragments to come after this one. If it's set to0, itmeans this is the last part or the only part of the datagram.

Offset

The 13-bit **fragmentation offset field tells us where this piecefitsinthe whole datagram**, counting in chunks of 8 bytes. Imaginewehaveadatagram with 4000 bytes split into three parts. Here's howit works:• The first piece covers bytes 0 to 1399 and has an offset of 0. • The second piece covers bytes 1400 to 2799 and has anoffsetof175.• The third piece covers bytes 2800 to 3999 and has anoffset of 350.

Imagine the fragments are like pieces of a puzzle sent fromoneplacetoanother. Even if they arrive in a different order, the receivingendcanstill put the puzzle together correctly. This happens because:

• The identification and flags fields are the same in all fragments, exceptfor the last one. • The offset field in each fragment shows its position relative

totheoriginal data. If a fragment itself gets split into smaller pieces, the offset is still

relativetothe original data. So, even if pieces take different paths and

arriveoutof order, they can be correctly reassembled. **To reassemble, follow these steps:**

- a. The first fragment has an offset of 0.
- b. Divide the length of the first fragment by 8 to get the offset of the second fragment.
- c. Divide the total length of the first and second fragments by 8togetthe offset of the third fragment.
- d. Continue this process. The last fragment has its "more" bit set to0.
- Remember that the value of the offset is measured inunitsof 8bytes.**This is**

done because the length of the offset field is only13bitslongand cannot represent a sequence of bytes greater than8191. Thisforces hosts or routers that fragment datagrams to **choosethesizeofeach fragment so that the first byte number is divisibleby8.**

Q - A datagram of 3000 B (20B of IP header + 2980B IPpayload) reached at router and must be forward to link with MTUof 500B. Howmany fragments will be generated and also write: MF, Offset, totlelength value for all.

Security of IPv4 Datagrams

The IPv4 protocol, as well as the whole Internet, was startedwhentheInternet users trusted each other. No security was providedfortheIPv4protocol. Today, however, the situation is different; the Internet isnotsecure anymore. There are three security issues that areparticularlyapplicable to the IP protocol:

- packet sniffing

- packet modification
- IP spoofing.

Packet Sniffing

- Packet Sniffing **An intruder may intercept an IP packet andmakeacopy of it.** Packet sniffing is a passive attack, in whichtheattacker**does not change the contents of the packet**. This typeof attackis very difficult to detect because the sender and the receiver maynever know that the packet has been copied. Althoughpacketsniffingcannot be stopped**, encryption of the packet can maketheattacker'seffort useless.** The attacker may still sniff the packet, but thecontentis not detectable

Packet Modification

- The 2nd type of attack is to modify the packet. The **attacker intercepts the packet, changes its contents, and sends thenewpacket to the receiver**. The receiver believes that the packet iscoming from the original sender. This type of attack canbedetectedusing a data integrity mechanism. The receiver, beforeopeningandusing the contents of the message, can use this mechanismtomakesure that the packet has not been changed during thetransmission.

IP Spoofing

- **A person with malicious intent can pretend to be someoneelseandsend an IP packet that looks like it's froma different computer.** Forexample, they can send a packet to a bank, making it seemlikeit'sfrom one of the bank's customers.

IPSec

The IP packets today can be protected from the previously mentionedattacks using a protocol called IPSec (IP Security). This protocol, whichisused in conjunction with the IP protocol, creates a connection-orientedservice between two entities in which they can exchange IP packetswithout worrying about the three attacks discussed above.

- **Defining Algorithms and Keys**. The two entities that want tocreateasecure channel between themselves can agree on some available algorithms and keys to be used for security purposes.
- **Packet Encryption.** The packets exchanged between two parties canbeencrypted for privacy using one

of the encryption algorithms andasharedkey agreed upon in the first step. This makes the packet sniffingattack useless.

- **Data Integrity.** Data integrity guarantees that the packet is notmodified during the transmission. If the received packet doesnotpass the data integrity test, it is discarded. This prevents thesecondattack, packet modification, described above.
- **Origin Authentication.** IPSec can authenticate the originof thepacketto be sure that the packet is not created by an imposter. Thiscanprevent IP spoofing attacks as described above.

IPv4 Addresses

- The Internet address, known as an IP address, is like a label fordevices on the Internet. It's a 32-bit code that tells theInternetwherea device is. It's important to note **that this address is for theconnection, not the device itself. If a device switches networks, itsIPaddress can change.** Each IP address is unique and correspondstoone Internet connection. So**, if a device has two Internet connections, it has two IP addresses.** These addresses areuniversal,meaning any Internet-connected device must recognizethem.

Address Space

- A protocol like IPv4 that defines addresses has an address space. Anaddress space is the total number of addresses used by theprotocol.If a protocol uses b bits to define an address, the address spaceis2^bbecause each bit can have two different values (0 or 1). IPv4uses32-bit addresses, which means that the address space is 2^(32) or4,294,967,296 (more than four billion). If there were norestrictions,more than 4 billion devices could be connected to theInternet.

Notation

IPv4 addresses can be written in three common ways: binary(0sand1s), dotted-decimal (numbers separated by dots, base 256), andhexadecimal (numbers and letters, base 16).

- In binary, it's 32 bits with spaces to make it readable usuallybetweeneach octet (8 bits).
- Dotted-decimal is the usual way, with dots separating numbersfrom0to 255.
- Hexadecimal uses numbers and letters and is commoninnetworkprogramming. Figure displays an IP address in these threeways.

Conversion D-B, B-D

Example 1

Example 2

Hierarchy in Addressing

In networks like telephone or postal systems, the addressing worksinastructured way. For example, a postal address includes country, state, city,street, house number, and recipient's name. Similarly, a phonenumberhascountry code, area code, local exchange, and the specific connection.

Now, in the case of a 32-bit IPv4 address, it's also structuredbut inasimplerway. It's divided into two parts: the first part (prefix) defines thenetwork,and the second part (suffix) defines the device on that network. Theprefixcan be either a fixed or variable length.

In the past, **IPv4 used fixed-length prefixes (classful addressing), butthat'soutdated. Nowadays, it uses variable-length prefixes (classless addressing).**We'll first touch on the old method and then focus on the newone.

Classful Addressing

- When the Internet started, an IPv4 address was designedwithafixed length prefix, but to accommodate both small and largenetworks,three fixed-length prefixes were designed instead of one(n=8, n=16, and n = 24). The whole address space was dividedintofiveclasses(class A, B, C, D, and E), as shown in Figure. This schemeis referredtoas classful addressing.

- **Class A**

the network length is 8 bits, but since the first bit, which is 0, defines theclass, wecan have only seven bits as the network identifier. This means there areonly2^7=128 networks in the world that can have a class A address.

- **Class B**

the network length is 16 bits, but since the first two bits, which are (10)_2, definethe class, we can have only 14 bits as the network identifier. This means thereareonly 2^14 = 16,384 networks in the world that can have a class B address.

- **Class C**

All addresses that start with (110)_2 belong to class C. In class C, the networklength is 24 bits, but since three bits define the class, we can have only21bitsasthe network identifier. This means there are 2^21 = 2,097,152 networks intheworld that can have a class C address.

- **Class D and Class E**

Class D is not divided into prefix and suffix. It is used for multicastaddresses. All addresses that start with 1111 in binary belongtoclassE. As in Class D, Class E is not divided into prefix and suffix andisusedas reserve.

Address Depletion

- Classful addressing became outdated because it led toaddressshortage. Internet addresses were not handed out efficiently, soweran out of them fast. For instance, class A addresses weregiventoonly 128 organizations, but each one got a massive chunkof addresses, which most didn't need. Class B addresses wereformidsize organizations, but many went unused. Class Caddresseshadtoo few usable addresses for companies. Class E addresses werehardly used at all, wasting the entire class. This inefficiencyinaddressdistribution caused problems, which is why we movedtoadifferentsystem.

Subnetting and Supernetting

- To deal with the problem of running out of addresses, twostrategies were suggested: subnetting and supernetting.
- Subnetting **involves breaking up a big address block (likeclassAorclass B) into smaller parts, creating more subnetworks withshorterprefixes.** This also allowed unused addresses to be sharedamongdifferent organizations. However, many **large organizationsdidn'tlike sharing their unused addresses**, so this idea didn't workwell.
- Supernetting, on the other hand, aimed to **combine multipleclassCblocks into a larger one to provide more addresses**

toorganizations. But this made it harder to route data packets effectively, soit

wasn'ta successful solution either.

Advantage of Classful Addressing

- Classful addressing, despite its issues and eventual obsolescence, hadone clear benefit:

it made it easy to determine the class of an address and, becauseeachclass had a fixed prefix length, you could instantly knowtheprefixlength from the address itself. In simple terms, the address itself contained all the information needed to figure out its prefixandsuffix;no extra details were required.

Classless Addressing

Address Deletion and the need for change

- Subnetting and supernetting in classful addressing didn't effectivelyaddress address depletion.
- The growth of the Internet highlighted the need for a larger addressspace.
- IPv6 was developed as a long-term solution, but a short-termsolutioncalled classless addressing was also introduced.
- Classless addressing removed the class distinctions in address distributionto address depletion.
- Internet Service Providers (ISPs) played a role in the adoptionof classlessaddressing.

Classless Addressing: A flexible solution

- Classless addressing introduced variable-length address blockswithout class distinctions.

• Address space is divided into variable-length blocks, wherethe prefixdefines the network, and the suffix defines the device. • Blocks can vary in size from 2^0 to 2^32 addresses.

• Prefix length ranges from 0 to 32, with smaller prefixes indicatinglarger networks and vice versa.

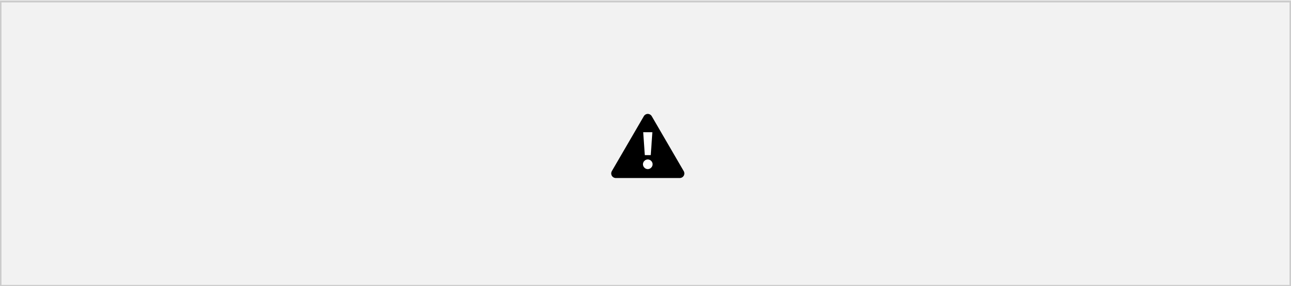
• Classless addressing can be applied to classful addressing, makingclassful addressing a special case of classless addressing.

Prefix Length: Slash Notation

• In classless addressing,

determining the prefix lengthis crucial.

- Unlike classful addressing, the prefix length is not inherent intheaddress.
- To specify the prefix length, it's added to the address usingslashnotation or CIDR (pronounced cider) strategy.
- The format is like this: address/prefix_length.
  - This means that in classless addressing, an address alonedoesn'tfullydefine its network; we also need to provide the prefix length.




Extracting Information fromanAddress

Given any address in the block, we normally like to knowthreepiecesof information about the block to which the address belongs: • the number of addresses

• the first address in the block

• the last address.

If the value of prefix length, n, is given, we can easily findthesethreepieces of information

Method for extracting
informationfromanaddress

- The number of addresses in the block is found as N= 2^(32−n). • To find the first address, we keep the n leftmost bits andset the(32−n) rightmost bits all to 0s.
- To find the last address, we keep the n leftmost bits andset the(32−n) rightmost bits all to 1s.

Example

A classless address is given as 167.199.170.82/27. Wecanfind the above three pieces of information as follows. Thenumber of addresses in the network is $2^{32-n}=2^{5}=32$addresses. The first address can be found by keepingthefirst 27 bits and changing the rest of the bits to 0s.

The last address can be found by keeping the first 27bitsand changing the rest of the bits to 1s.

Address Mask

Another way to find the first and last addresses in the block is tousetheaddress mask. The address mask is a 32-bit number in whichthenleftmostbits are set to 1s and the rest of the bits (32 − n) are set to 0s. Acomputercan easily find the address mask because it is the complement of (2^(32−n)− 1). The reason for defining a mask in this way is that it can beusedbyacomputer program to extract the information in a block, using thethreebit-wise operations NOT, AND, and OR.

1. The number of addresses in the block N = NOT (Mask) + 1.
2. The first address in the block = (Any address in the block) AND (Mask). 3. The last address in the block = (Any address in the block) OR [(NOT (Mask)].

Example

The mask in dotted-decimal notation is 255.255.255.224 The AND, OR, and NOT operations can be applied to individual bytesusingcalculators.

Number of addresses in the block: N = NOT (mask) + 1= 0.0.0.31+1=32addresses
First address: First = (address) AND (mask) = 167.199.170. 82Last address: Last = (address) OR (NOT mask) = 167.199.170. 255

Network Address

- The provided examples demonstrate our ability to retrieve block-relatedinformation from any given address.
- The network address, which is the first address, is vital for

routingpacketstotheirdestination networks.

- Let's assume there are m networks in an internet, each connectedtoarouterwithm interfaces.
- When a packet arrives at the router from any source host, theroutermustdetermine which network to send the packet to and which interfacetouse.• The packet employs a different strategy to reach its final destinationhostonceitreaches the network.
- Figure visually represents this process.
- **After identifying the network address, the router checks its forwardingtabletofind the correct interface for sending out the packet.** • Each network is uniquely identified by its network address.

Block Allocation

The next concern with classless addressing is how blocks are handedout.The job of giving out these blocks falls on a global authority namedICANN.But, ICANN doesn't usually hand out addresses to regular internet users. Instead, it gives a big chunk of addresses to an ISP (or a similar bigorganization that acts like an ISP). To make CIDR work well, therearetwoimportant rules that need to be followed for these allocatedblocks.

1. The number of requested addresses, N, needs to be a power of 2. The reason is that $N = 2^{(32-n)}$ or $n = 32 - \log_2(N)$. If N is not a power of 2, we cannot have an integer value for n.
2. The requested block needs to be allocated where there are a contiguous number of available addresses in the address space. However, there is a restriction on choosing the first address in the block. The first address needs to be divisible by the number of addresses in the block. The reason is that the first address needs to be the prefix followed by (32 − n) number of 0s. The decimal value of the first address is then

$$\text{first address} = (\text{prefix in decimal}) \times 2^{(32-n)} = (\text{prefix in decimal}) \times N$$

Example

An ISP has requested a block of 1000 addresses.

Since 1000 is not a power of 2, 1024 addresses are granted. The prefixlengthiscalculated as n = 32 − log2(1024) = 22. An available block, 18.14.12.0/22, isgrantedtothe ISP. It can be seen that the first address in decimal is 302,910,464, whichisdivisible by 1024.

Subnetting

Subnetting allows for the creation of additional layers in the networkhierarchy. When an organization or ISP is given a block of addresses, **theycan split it into smaller sub-blocks and allocate each of thesetoseparatesubnetworks**. Importantly, there's no limit to how deep this hierarchycango.A subnetwork can be further divided into sub-subnetworks, andtheprocesscan continue with sub-sub-subnetworks, and so forth.

Finding Information about EachSubnetwork

After designing the subnetworks, the information about eachsubnetwork,such as first and last address, can be found using the process wedescribedto find the information about each network in the Internet.

EXAMPLE: An organization is granted a block of addresseswith the beginning address 14.24.74.0/24. The organizationneeds to have 3 subblocks of addresses to use initsthreesubnets: one subblock of 10 addresses, one subblockof60addresses, and one subblock of 120 addresses. Designthesubblocks.

Solution

There are $2^{32-24} = 256$ addresses in this block. Thefirst address is 14.24.74.0/24; the last address is 14.24.74.255/24. To satisfy the third requirement, we assign addressestosubblocks, starting with the largest and endingwiththesmallest one.

a. The number of addresses in the largest subblock, whichrequires 120 addresses, is not a power of 2. We allocate128addresses. The subnet mask for this subnet can be foundas$n_1 = 32 - \log_2 128 = 25$. The first address in this blockis14.24.74.0/25; the last address is 14.24.74.127/25.

b. The number of addresses in the second largest subblock, which requires 60 addresses, is not a power of 2either. Weallocate 64 addresses. The subnet mask for this subnet canbe found as $n_2 = 32 - \log_2 64 = 26$. The first address inthisb l o c k is 1 4 . 2 4 . 7 4 . 1 2 8 / 2 6 ; t h e l a st a ddr essis14.24.74.191/26.

c. The number of addresses in the smallest subblock, whichrequires 10 addresses, is not a power of 2. We allocate16addresses. The subnet mask for this subnet can be foundas$n_1 = 32 - \log_2 16 = 28$. The first address in this blockis14.24.74.207/28; the last address is

14.24.74.207/28.

If we add all addresses in the previous subblocks, theresultis 208 addresses, which means 48 addresses areleft inreserve. The first address in this range is 14.24.74.208. Thelast address is 14.24.74.255. We don’t knowabout theprefixlength yet. Figure 4.36 shows the configurationof blocks. We have shown the first address in each block.