CHAPTER

9

The Transport Layer

Transport of the mails, transport of the human voice, transport of flickering pictures — in this century as in others our highest accomplishments still have the single aim of bringing men together.

— Antoine de Saint-Exupéry

The last chapter talked about the upper layers of the OSI reference model. You learned the specific purpose of each layer and how the layers interact with each other. This chapter covers the Transport layer, Layer 4 of the OSI reference model.

The Transport layer is the highest layer of the lower layers of the OSI reference model. The Transport layer sits on top of the Network layer and below the Session layer. This layer is responsible for the end-to-end connection and datagram delivery, as well as congestion control and flow control. The two main protocols that operate at this layer are UDP and TCP, which were discussed in Chapter 5.

The purpose of the Transport layer is to set up connections, maintain connections, shut down connections, and perform error checking.[1] The protocols that operate at this layer are considered either connection-oriented (i.e., TCP) or connectionless (i.e., UDP). Remember that connection-oriented means that the connection must be set up before data can be transmitted, and connectionless means that data can flow without the connection being established first.

[1]Error checking and other transport reliability attributes can be handled at this layer, if they are not already performed at the lower layers.

So far this book has explained what the Transport layer is and the services and protocols it provides. This chapter takes a little deeper look into some of the functions that operate at this layer.

RANDOM BONUS DEFINITION

1000BASE-SX — A baseband Ethernet system operating at 1000 Mbps over two multimode optical fibers using shortwave laser optics.

9.1 The Terms and Conditions of Chapter 9

Much like many other chapters in this book, there are some terms you need to have an understanding of, but not necessarily in-depth knowledge. Therefore, we start this chapter off with a few basic Transport layer functions and terms relating to these.

RANDOM BONUS DEFINITION

root port — In the Spanning Tree Protocol, the port through which a designated bridge forwards traffic in the direction of the root bridge.

9.1.1 End-to-End Delivery

The Transport layer provides logical communication between upper layer processes[2] running on different nodes on a network (see Figure 9-1).

Notice in the figure that the lower layer processes are transparent to the Transport layer. The sending node takes the upper layer data and breaks it into smaller segments that are then passed to the lower layers to be encapsulated and transported to a receiving node. The receiving node will cache the data, put the segments back into the message, and pass it to the upper layers to be delivered to the Application layer.

POP QUIZ

True or false: UDP is an example of a connectionless protocol.

[2]Notice this says processes and not nodes. The Network layer provides the logical connection between nodes.

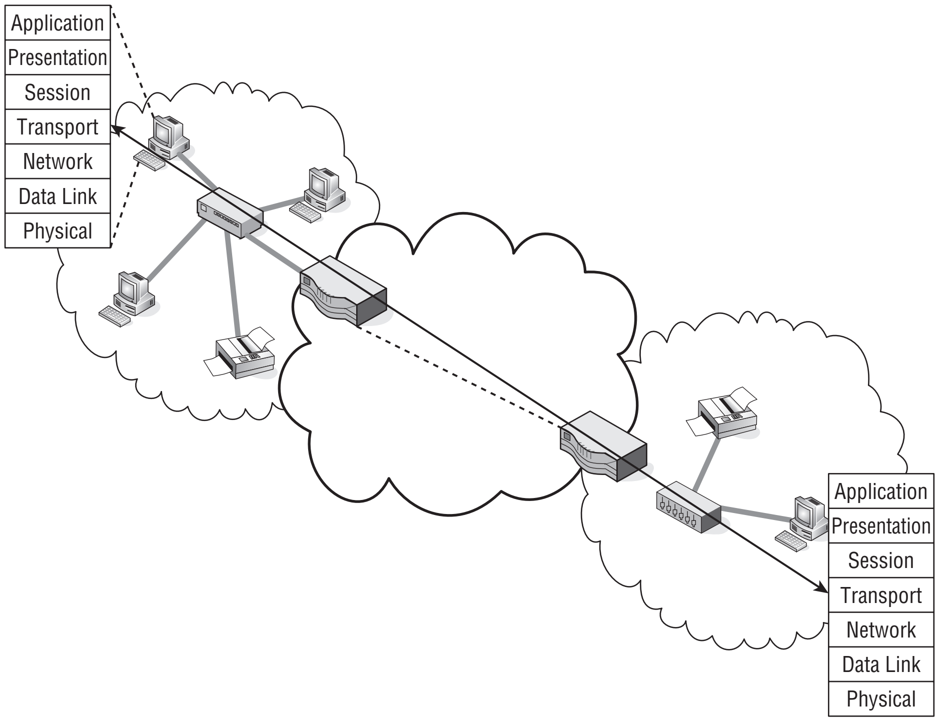

Figure 9-1 Logical Transport layer communications

9.1.2 Standards

Before getting too much further into this chapter, there are a couple of standards that need to be mentioned that deal with the services and operations of functions at the Transport layer. The first of these is the ISO/IEC 8072 standard (Information technology – Open Systems Interconnection – Transport service definition), and the other is the ISO/IEC 8073 standard (Information technology – Open Systems Interconnection – Protocol for providing the connection-mode transport service).

Following is a quick summary of these two standards. The remainder of the chapter covers the information that is defined in the standards.

9.1.2.1 ISO/IEC 8072

The ISO/IEC 8072 standard defines the recommended services provided by the OSI Transport layer while working with the Network layer to serve the needs of protocols used at the Session layer. These are only recommendations or guidelines, and strict adherence is not upheld.[3] Defined in this standard are recommendations for the implementation for the following functions:

- Connection-oriented mode services
- Connectionless-mode services

The main thing to remember about this standard is that it defines the way the Transport layer interoperates with the other OSI layers it works with.

ACRONYM ALERT

BGP — Border Gateway Protocol

9.1.2.2 ISO/IEC 8073

The ISO/IEC 8073 standard sets the recommendations to be followed by nodes (entities) within a network that are utilizing the services of the OSI Transport layer. This standard is also available to future node deployments within an open systems environment. Defined in this standard are recommendations for the following functions:

- The recommendation and scope for classes of procedures that should be taken into account by the nodes when transporting data
- How peer nodes exchange data
- How the nodes exchange information with the transport service
- The manner in which the nodes exchange information with a service provider

9.1.3 This, That, and the Other

This section takes a look at a few other ''items of interest'' regarding the Transport layer.

RANDOM BONUS DEFINITION

aggregated link — A set of two or more physical links that appear to higher layer entities as though they were a single, higher capacity link.

9.1.3.1 Types of Transport Service

This is an easy one.[4] There are two types of transport service: connection-oriented and connectionless.

[3]Keep in mind that all of the functions at each of the layers in the reference model are only recommendations and guidelines that can be followed for conformity sake.
[4]At least we hope it is.

9.1.3.2 Data Units

The following two data units operate at the Transport layer:

- Transport protocol data unit (TPDU)
- Transport service data unit (TSDU)

So, what is the difference between the two types of data units? The *TSDU* is the data that is transmitted to the various layers on both ends of a connection. The *TPDU* is the data that is sent from a protocol on one end to the peer protocol at the other end.

POP QUIZ

TCP is a connection-____________ protocol.

9.1.3.3 Classes of Transport Service

The Transport layer defines the functions of service performed by it within five difference classes of transport service, as shown in Table 9-1.

ACRONYM ALERT

IETF — Internet Engineering Task Force

Table 9-1 Classes of Transport Service

CLASS NAME	CLASS FUNCTION
Class 0	Simple class
Class 1	Basic error recovery class
Class 2	Multiplexing class
Class 3	Error recovery and multiplexing class
Class 4	Error detection and recovery class

9.1.3.4 Types of Network Service

The Transport layer takes into consideration the current error rate status of the connection being used. There are three types of network service used to classify the connection status. The data units are classified into one of the three types based on signal quality:

- **Type A** — A network connection with an acceptable residual error rate as well as an acceptable rate of signal failures.

- **Type B** — A network connection with acceptable residual error rate but an unacceptable rate of signal failures.
- **Type C** — A network connection with an unacceptable residual error rate for the user of the transport service.

RANDOM BONUS DEFINITION

promiscuous mode — A mode of operation of a network interface in which it receives (or attempts to receive) all traffic, regardless of the destination address.

9.1.3.5 Multiplexing

Multiplexing is the act of grouping several signals into a shared single signal. Multiplexing at the Transport layer is performed between the Transport layer and its adjoining layers. Multiple upper layer users can be multiplexed to share the services of a single Transport layer protocol. The signals are separated by what are known as *transport service access points (TSAP)*. An example of this is shown in Figure 9-2.

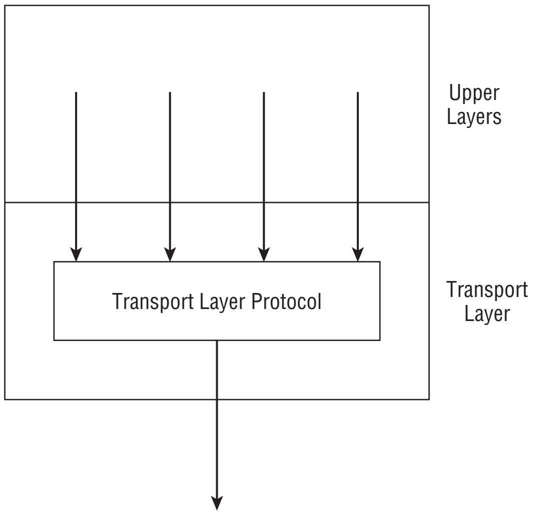

Figure 9-2 An example of multiplexing

Network service multiplexing is also supported at the Transport layer. Multiplexing can occur in both an upward (multiple Transport layer signals to a single network signal) and a downward (multiple network signals to a single transport signal) fashion.

The use of upward multiplexing (see Figure 9-3) is a cost-saving measure that allows multiple Transport layer signals to share a single network signal (a signal purchased from the network provider).

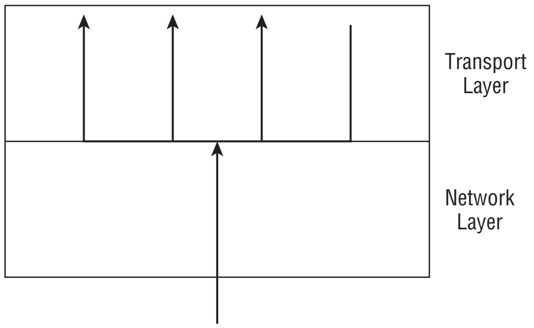

Figure 9-3 Upward multiplexing

Downward multiplexing (see Figure 9-4) is useful when bandwidth and throughput of data are priorities.

RANDOM BONUS DEFINITION

best-effort service — A service provided by an entity where frames or packets are delivered with high probability but with no absolute guarantee.

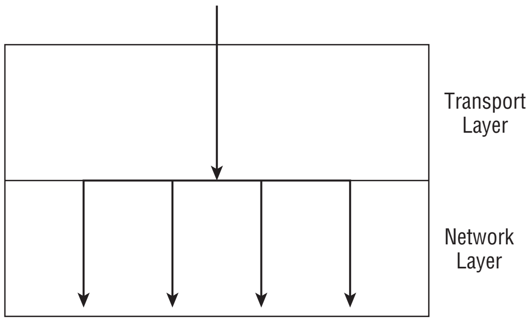

Figure 9-4 Downward multiplexing

AN UNRELATED MOMENT OF PAUSE: WEB ACRONYMS

It isn't just the networking world that uses acronyms. Millions of users are typing away with acronyms that a few years ago didn't exist. A lot of kids out there have added their own, such as POS (parent over shoulder). Jim has already prepared to start running a sniffer if he sees one of his kids using that one — having to warn a pal that the parent is looking in deserves a quick look-see.

Enough rambling. Here is a list of some common web acronyms that you may come across at some point.

(continued)

AN UNRELATED MOMENT OF PAUSE: WEB ACRONYMS *(continued)*

2L8	Too late
AFK	Away from keyboard
AFN	[That's] all for now
AISB	As I said before
B4	Before
B4N	Bye for now
BAK	Back at keyboard
BBL	Be back later
BCNU	Be seeing you
BRB	Be right back
BTW	By the way
CU	See you
CYA	See ya
DL	Download
EZ	Easy
F2F	Face to face
FWIW	For what it's worth
G2G	Gotta go
GMTA	Great minds think alike
HAND	Have a nice day
IC	I see
IDK	I don't know
IK	I know
IKWUM	I know what you mean
IMAO	In my arrogant opinion
IMHO	In my humble opinion
IMO	In my opinion
IYKWIM	If you know what I mean
IYO	In your opinion
IYSWIM	If you see what I mean
JK	Just kidding

(continued)

AN UNRELATED MOMENT OF PAUSE: WEB ACRONYMS *(continued)*

KISS	**Keep it simple, stupid**
LOL	**Laughing out loud**
ME2	**Me too**
NP	**No problem**
ROTFL	**Rolling on the floor laughing**
TC	**Take care**

9.2 Transport Layer Operations

The purpose of the Transport layer is to provide end-to-end delivery of data from one application to another. The Transport layer can deliver data in a reliable or an unreliable fashion. Data flow can be regulated and each end can communicate lost datagram data with the other end. Protocols can operate in a connection-oriented manner as well as a connectionless manner. In the connection-oriented approach, a logical connection between nodes must be established before any data is transmitted. The connectionless approach does not require connection establishment; data is sent as it is received.

POP QUIZ

Which standard defines the way the Transport layer interoperates with the other OSI layers it works with?

In this section, we take a deeper look into the operations for both the connection-oriented as well as the connectionless protocols that are available within the Transport layer.

ACRONYM ALERT

ROM — Read-only memory

9.2.1 Connection-Oriented Operations

Connection-oriented protocols require that a logical connection between two nodes is established before any data can be sent. To do this, rules are established that lay out how a connection is set up, maintained, and terminated.

9.2.1.1 Setting Up the Connection

If a node needs to pass data in a connection-oriented environment, a series of messages is passed between the node and the destination node it wants to send the data to. The series of messages is known as the *three-way handshake*, and it works like this:

1. The originating node will send a request known as a *SYN*[5] to the destination node.
2. The destination node will let the originating node know that it has received the SYN request by sending back a *SYN-ACK*[6] message.
3. The originating node will respond to the SYN-ACK by sending back an *ACK* message.

Figure 9-5 shows an example of this.

ACRONYM ALERT

MSTI — Multiple spanning tree instance

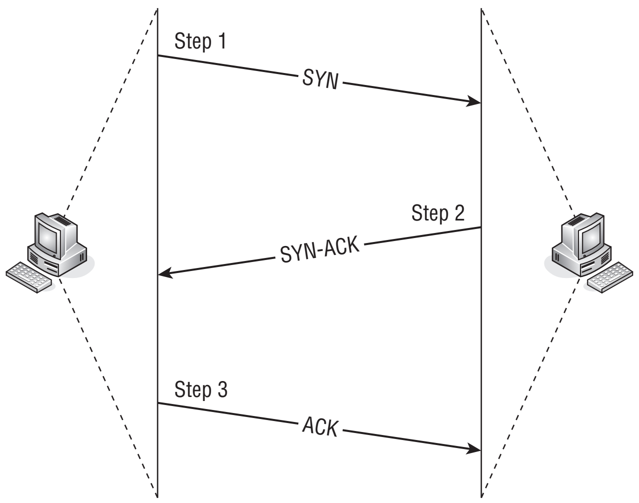

Figure 9-5 An example of a three-way handshake

Don't be fooled into believing this is all that's going on in the connection setup phase. A number of variables are being negotiated during this phase. User node quality of service is matched to any available services that are provided by the Network layer. Some of the services negotiated include

- Which network services best match requirements set by the user for the connection

[5]SYN stands for synchronize.
[6]ACK stands for acknowledgment.

- Whether multiplexing can (or should) be used
- Datagram size
- Address mapping
- Ability to separate multiple connections
- Inactivity timer information

RANDOM BONUS DEFINITION

optical fiber — A communications medium capable of carrying and directing light signals. Normally extruded or drawn from transparent glass or plastic material.

9.2.1.2 Maintaining the Connection

Maintaining the connection is nothing more than ensuring the connection remains stable during the transfer of data between the endpoint nodes. The following activities occur during this phase:

- Segmentation of data
- Reassembly of data
- Splitting data over multiple connections
- Flow control
- Setting the identification parameters for a particular connection between endpoint nodes
- Attending to prioritized datagrams
- TSDU delimiting

9.2.1.3 Terminating the Connection

Just like with the connection setup phase, there has to be a way to terminate the connection when the endpoint nodes are finished exchanging data. This phase operates much like the connection establishment phase.

Any node that has an active connection can initiate a connection termination by sending out a *FIN*[7] packet (or by setting a flag in a datagram). The other node can continue receiving data until it sends out a FIN-ACK, acknowledging the request to terminate the session.

RANDOM BONUS DEFINITION

collision detection — The act of detecting when packets collide during transmission.

[7]FIN stands for finished.

9.2.2 Connectionless Operations

Connectionless protocols do not require a connection; a transmitting device simply sends data as soon as it has data that is ready to be sent. Protocols that operate in a connectionless manner have a space available in the datagram to identify the source and destination addresses for the endpoint nodes. Connectionless protocols do need an available route to the destination in order to work. This means there must be some type of medium, a data link protocol, and a networking protocol to transmit the data. Other than these, there really is no other requirement.

Protocols that use the connectionless method of transport will often provide error checking and recovery methods, which are lacking in the connectionless environment. Some of these include:

- Hop count verification
- Verification of the reassembly of fragmented data
- Datagram priority information and verification
- Datagram size verification

ACRONYM ALERT

SNA — Systems network architecture

POP QUIZ

How many types of transport service are there?

TIME FOR SOMETHING NICE TO KNOW

Following are some helpful MS-DOS commands that are available with most Windows OS platforms.

- **To determine whether a remote node is reachable and its connection quality, use the `ping` command.**

```
C:\>ping
Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
            [-r count] [-s count] [[-j host-list] |
            [-k host-list]]
            [-w timeout] destination-list
Options:
  -t        Ping the specified host until stopped.
           To see statistics and continue - type
           Control-Break;
           To stop - type Control-C.
  -a        Resolve addresses to hostnames.
  -n count    Number of echo requests to send.
  -l size    Send buffer size.
```

(continued)

TIME FOR SOMETHING NICE TO KNOW *(continued)*

```
  -f         Set Don't Fragment flag in packet.
  -i TTL      Time To Live.
  -v TOS      Type Of Service.
  -r count    Record route for count hops.
  -s count    Timestamp for count hops.
  -j host-list  Loose source route along host-list.
  -k host-list  Strict source route along host-list.
  -w timeout   Timeout in milliseconds to wait for each reply.
```

◆ **To follow the path that is taken by a datagram to a remote node, use the `tracert`[8] command.**

```
C:\>tracert
Usage: tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout]
       target_name
Options:
  -d          Do not resolve addresses to
            hostnames.
  -h maximum_hops   Maximum number of hops to search for
            target.
  -j host-list    Loose source route along host-list.
  -w timeout     Wait timeout milliseconds for each
            reply.
```

◆ **To view and manage the local routing table, use the `route` command.**

```
C:\>route
ROUTE [-f] [-p] [command [destination]
        [MASK netmask]  [gateway] [METRIC metric]
        [IF interface]
 -f      Clears the routing tables of all gateway
       entries. If this is used in conjunction with
       one of the commands, the tables are cleared
       prior to running the command.
 -p      When used with the ADD command, makes a route
       persistent across
       boots of the system. By default, routes are
       not preserved
       when the system is restarted. Ignored for all
       other commands,
       which always affect the appropriate
       persistent routes. This option is not
       supported in Windows 95.
 command    One of these:
        PRINT    Prints a route
```

(continued)

[8]Tracert stands for trace route.

TIME FOR SOMETHING NICE TO KNOW *(continued)*

```
           ADD    Adds a route
           DELETE  Deletes a route
           CHANGE  Modifies an existing route
    destination Specifies the host.
    MASK     Specifies that the next parameter is the
          'netmask' value.
    netmask   Specifies a subnet mask value for this route
          entry.
          If not specified, it defaults to
          255.255.255.255.
    gateway   Specifies gateway.
    interface  the interface number for the specified route.
    METRIC    specifies the metric, ie. cost for the
          destination.
```

- **To view and manage the ARP table, use the `arp` command.**

```
C:\>arp
Displays and modifies the IP-to-Physical address translation
tables used by address resolution protocol (ARP).
ARP -s inet_addr eth_addr [if_addr]
ARP -d inet_addr [if_addr]
ARP -a [inet_addr] [-N if_addr]
 -a      Displays current ARP entries by
        interrogating the current
        protocol data.  If inet_addr is specified,
        the IP and Physical
        addresses for only the specified computer
        are displayed. If more than one network
        interface uses ARP, entries for each ARP
        table are displayed.
 -g      Same as -a.
 inet_addr   Specifies an internet address.
 -N if_addr  Displays the ARP entries for the network
        interface specified by if_addr.
 -d      Deletes the host specified by inet_addr.
        inet_addr may be wildcarded with * to delete
        all hosts.
 -s      Adds the host and associates the Internet
        address inet_addr with the Physical address
        eth_addr. The Physical address is
        given as 6 hexadecimal bytes separated by
        hyphens. The entry is permanent.
 eth_addr   Specifies a physical address.
 if_addr    If present, this specifies the Internet
        address of the interface whose address
        translation table should be modified.
        If not present, the first applicable
        interface will be used.
```

9.3 Transport Layer Protocols

Chapter 5 provided some information on the TCP and UDP Transport layer protocols. Although these are the most popular and most commonly used, the following Transport layer protocols are also in use in some networks today:

- AppleTalk Transaction Protocol (ATP)
- Datagram Congestion Control Protocol (DCCP)
- NetBIOS Extended User Interface (NetBEUI)
- Real-time Transport Protocol (RTP)

These are mentioned only to provide you with the names of a few more Transport layer protocols that you may come across. For the purposes of this book, TCP and UDP are the Transport layer protocols that we will stick with.

ACRONYM ALERT

PPP — Point-to-Point Protocol

9.3.1 A Few More Words about TCP

TCP is a *connection-oriented* protocol. An originating node will contact a destination node to make sure it is available to get the message. Once confirmation is received that it is okay to send data, the transmission begins. TCP is also considered a *reliable* protocol because it has functions built into it that provide for various checks and balances to ensure the integrity of the data being transmitted.

TCP is able to break data down into segments so that smaller chunks of data are lost if there are problems with the transmission. TCP supports acknowledgments for received datagrams, and timers are set for the receipt of an acknowledgment to ensure that data is received on the destination end. TCP utilizes a checksum to monitor data receipt integrity. TCP also supports datagram reassembly, ensuring that it is put back into the same order it was sent. Finally, TCP supports both congestion control and flow control, allowing a sending node to monitor bandwidth availability as well as whether the receiving node can receive any more data.

RANDOM BONUS DEFINITION

jumbo frame — A frame longer than the maximum frame length allowed by a standard.

TCP uses sequence numbers between nodes to ensure that reliable communication is taking place. Receiving nodes use sequence numbers to put the data back in order when it is received. Sequence numbers are also used to identify problems (lost packets, duplicate packets, etc.) that may occur with a specific packet that had been transmitted. Each end of the connection maintains its own sequence numbers, so data transmission can operate in a full-duplex manner. TCP is known as a *byte-oriented sequencing protocol* because every byte[9] that is being transmitted is assigned a sequence number. The TCP packet is assigned the sequence number of the first byte of the packet. The following packet will get assigned the sequence number of its first byte, and so on. Figure 9-6 provides an example of sequencing.

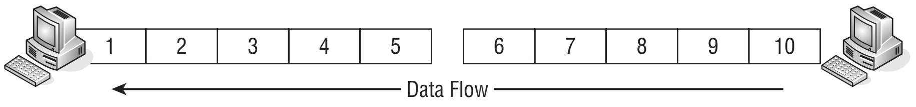

Figure 9-6 TCP sequencing

In the figure, you can see that data is flowing from one node to another. The receiving node recognizes that it is receiving a packet with a sequence number of 1. As the node receives the packet, the number of bytes in the packet is counted. This will tell the node what packet sequence number is expected next. As you can see, there were 5 bytes[10] in the first packet, so the next packet should start with a sequence number of 6. And that, my friend, is TCP byte sequencing.

RANDOM BONUS DEFINITION

D-compliant — A bridge or switch that complies with IEEE 802.1D.

TCP also uses acknowledgment numbers that work hand in hand with the sequence numbers. Acknowledgment numbers are simply the sequence numbers in reverse. They are the reply from the destination node that sequence number such-and-such has been received. Figure 9-7 provides an example of how this works.

POP QUIZ

The ______________ is the data that is transmitted to the various layers on both ends of a connection.

[9]As opposed to some protocols that assign a sequence number to a whole datagram.
[10]Five-byte TCP segments? Now, that's funny. This number was picked at random for use in the example. TCP segments normally have 512 bytes.

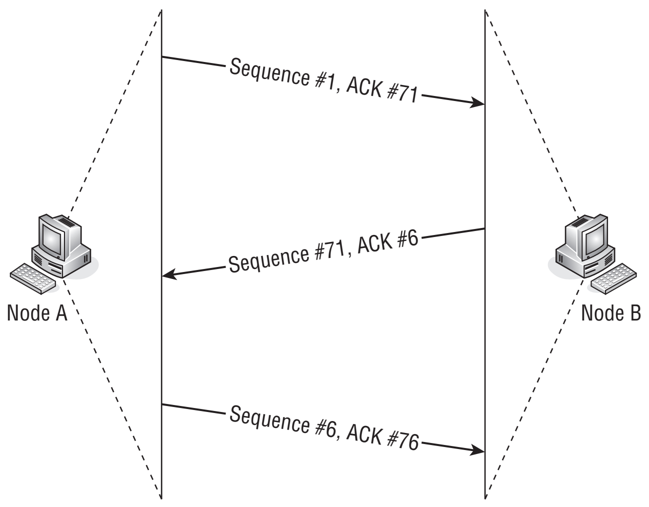

Figure 9-7 Sequencing and acknowledgement

The figure represents communication between a pair of nodes. The originating node sends a packet that is assigned sequence number 1 (because the first byte sequence number is 1) and then sends an acknowledgment of a received datagram. The acknowledgment number is actually the sequence number that the node is expecting next. In the figure, Node A sends a packet to Node B. The packet has a sequence number of 1, and an acknowledgment number of 71. This means that Node A is telling Node B that it has received a packet and the next one it is expecting is sequence number 71. Node B sends a packet with sequence number 71 and the acknowledgment that packet sequence number 1 was received and the node is ready to receive sequence number 6. Node A then sends the next packet and acknowledges receipt of a previous packet. This process continues until data transmission is no longer required.

ACRONYM ALERT

VLAN — Virtual local area network

ACRONYM ALERT

SRAM — Static random-access memory

9.3.2 The TCP Header Format

The TCP header and the upper layer data are joined to form a TCP segment. The TCP header is where the sequencing number and acknowledgment number are maintained, as well as many other factors needed for proper data delivery. Figure 9-8 shows the format of the TCP header.

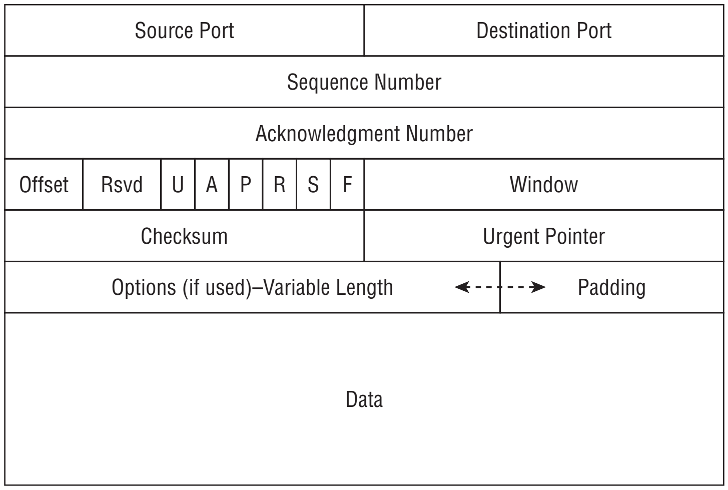

Figure 9-8 The format of the TCP header

- **Source Port** — A 16-bit number that identifies the application that sent the TCP segment.
- **Destination Port** — A 16-bit number that identifies the application the TCP segment is destined for.
- **Sequence Number** — A 32-bit number that identifies the first data byte in the segment.
- **Acknowledgment Number** — A 32-bit number that identifies the next data byte the node expects to receive.
- **Offset** — A field that identifies the length of the TCP header.
- **Rsvd** — An unused field reserved for potential future use.
- **U/A/P/R/S/F** — This field grouping contains the control fields:
  - **U** — Urgent. If this field is set, the destination (receiving) node knows there is urgent data waiting to be sent.
  - **A** — Acknowledgment. This is set when the packet has an acknowledgment for a received datagram.
  - **P** — Push. When this field is set, the receiver needs to deliver the segment to the receiving application ASAP.[11]

[11] As soon as possible.

  - **R** — Reset. When this is set, it tells the receiving node that the originator is terminating the connection.
  - **S** — Synchronize. This field is set at startup when setting sequence numbers.
  - **F** — Finished. There will be no more data coming.
- **Window —** A 16-bit number used by TCP for flow control. It indicates the number of available buffers the sending node has.
- **Checksum** — A 16-bit number used for error detection.
- **Urgent Pointer** — This is a 16-bit field. When the Urgent bit is set, there will be a number that points to the sequence number of the data that follows urgent data. This identifies to the destination node that the last byte of urgent data was received.
- **Options** — TCP support options that can be set for the data. This is a variable length field, depending on the option data.
- **Padding** — Padding fills the remainder of the 32-bit field. This is necessary due to the optional and variable length Options field.
- **Data —** The application data: the payload!

POP QUIZ

How many different classes of transport service are there?

9.3.3 A Little More on UDP

UDP is a connectionless protocol. It does not guarantee that data is going to be delivered to a destination. UDP simply transmits data when it has data that is ready to be transmitted. Remember that UDP is usually used to send short bursts of datagrams between nodes where reliability is not a big concern. UDP can get data to a destination quicker, as it avoids the overhead required by all the checks and balances in TCP. Also, because UDP is connectionless, it can support *broadcasting* (sending messages to all nodes within a broadcast domain) and *multicasting* (sending messages to all nodes that are subscribed to the catenet).

RANDOM BONUS DEFINITION

E1 — A T-carrier technology commonly used in Europe, capable of multiplexing 32 DS-0 (64 Kbps) channels for a total data-carrying capacity of 2.048 Mbps.

UDP accepts data (the payload) from the Application layer. It adds a UDP header and passes the header and the payload to the Internet layer. There it is encapsulated into an IP packet and passed on to the Network Interface layer, then is passed over the transmission medium to the destination, where it makes its way up to the Application layer on the destination end of the connection.

UDP segments can be lost along the way. They can also be received out of sequence. This is why UDP is known as a *best-effort protocol*. UDP is beneficial when you need to transmit a lot of data. There is no delay with UDP, as there is no need to set up a connection prior to the distribution of the data. If an application needs a method of recovering from errors, the application will handle this task itself. UDP also uses a checksum, which is a method for detecting transmission errors.

POP QUIZ

True or false: FIN stands for finished.

9.3.4 The UDP Header Format

The UDP header and the upper layer data are joined to form a UDP segment. The UDP header is simpler than the TCP header due to the overhead required for the connection-orientation used by TCP. Figure 9-9 shows the format of the UDP header.

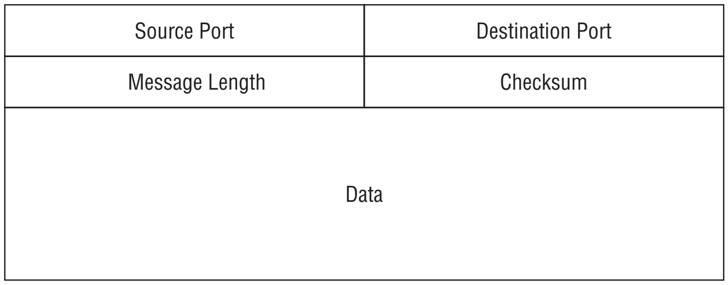

Figure 9-9 The format of the UDP header

- **Source Port** — A 16-bit number that identifies the application that sent the UDP segment.
- **Destination Port** — A 16-bit number that identifies the application the UDP segment is destined for.
- **Message Length** — A field that identifies the length of the UDP header.

- **Checksum** — A 16-bit number used for error detection.
- **Data** — The application data: the payload!

POP QUIZ

True or false: The AppleTalk Translucent Protocol is a transport layer protocol.

9.4 The Meaning of Control

In a connection-oriented environment, control of data transmission is important to ensure data delivery. Congestion control and flow control are two mechanisms used. Congestion control is used to avoid congestion on a link by avoiding the oversubscription of the rate that is supported by the link and reducing the rate of datagram transmission when congestion is on the link.

Flow control is a mechanism that an originating node uses to ensure that a destination node can handle the amount of data being transferred.

POP QUIZ

What is a TCP source port?

9.5 Chapter Exercises

1. What are the two ISO/IEC standards that define recommendations for the transport layer?

   ____

   ____

2. What are the two types of transport service?

   ____

   ____

3. From the following list, fill in the class function in the table below.

   Multiplexing class

   Error detection and recovery class

   Simple class

   Error recovery and multiplexing class

Basic error recovery class

Class Name	Class Function
Class 0	
Class 1	
Class 2	
Class 3	
Class 4	

4. Match the type with the correct description:

   Type __________ Network connections that maintain an unacceptable rate of residual errors

   Type __________ Network connections that maintain both an acceptable rate of signaled errors and residual errors

   Type __________ Network connections that maintain an acceptable rate of residual errors and an unacceptable rate of signaled errors
5. Define *upward multiplexing*.
6. Define *downward multiplexing*.
7. Explain how a three-way handshake works.
8. List four Transport layer protocols.

____

____

____

____

9.6 Pop Quiz Answers

1. True or false: UDP is an example of a connectionless protocol.

   True
2. TCP is a *connection-oriented* protocol.
3. Which standard defines the way the Transport layer interoperates with the other OSI layers it works with?

   ISO/IEC 8072

4. How many types of transport service are there?

   Two

5. The *TSDU* is the data that is transmitted to the various layers on both ends of a connection.

6. How many different classes of transport service are there?

   Five

7. True or false: FIN stands for finished.

   True

8. True or false: The AppleTalk Translucent Protocol is a Transport layer protocol.

   False. It is the AppleTalk Transaction Protocol. (Gotcha!)

9. What is a TCP source port?

   The TCP source port is part of the TCP header. It is the 16-bit number that identifies the application that sent the TCP segment.