MONASH University
Information Technology

FIT3142 Distributed Computing

Topic 2: Protocols for Parallel and Distributed Systems

Dr Carlo Kopp
Dr Asad I. Khan
Faculty of Information Technology
Monash University

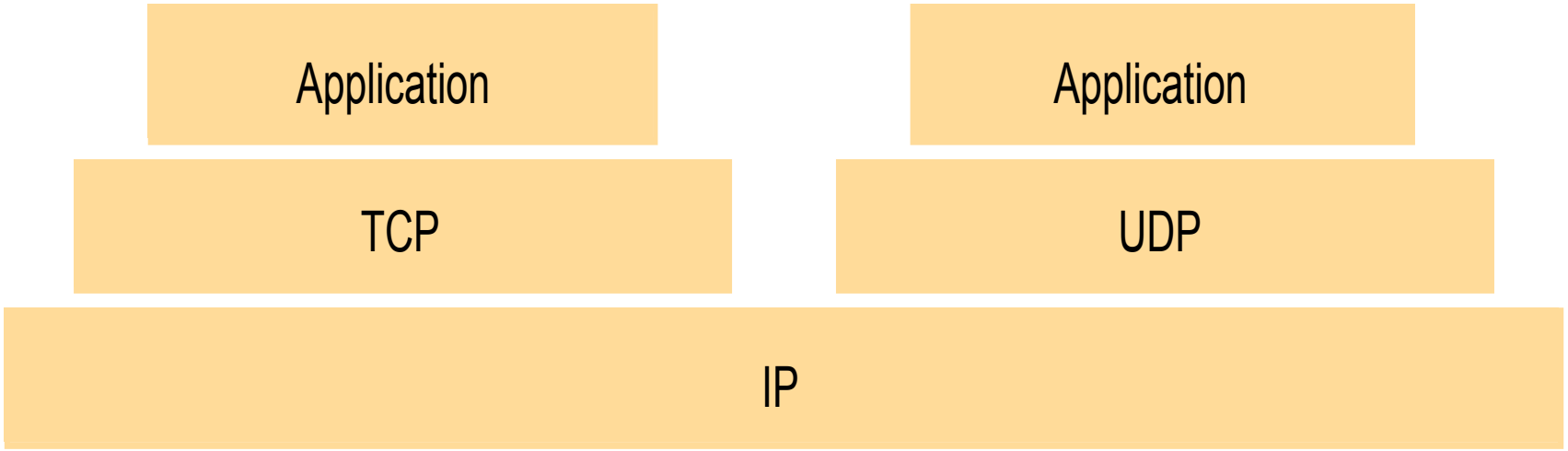
Why Study Protocols for Parallel and Distributed Systems?

- Protocols are the "glue" that binds the components of all distributed systems into an actual system;
- Foundation knowledge: Because the behaviour and performance of protocols determines many critical aspects of distributed system behaviour and performance, understanding protocol behaviour is essential;
- Foundation knowledge: The limitations of protocol can limit what a distributed application can or cannot do;
- Practical skills: In many distributed applications you will have to rely on middleware written around the underlying protocols and their limitations;
- Practical skills: The throughput performance of a protocol can become a 'live or die" problem for a distributed application, making this an important item to understand;

Legacy Protocols and Limitations

- The most widely used protocols for carrying network traffic are TCP and UDP, developed during the 1980s;
- The intent, when TCP and UDP were developed, was to support remote dumb terminal traffic, and file transfers;
- The earliest applications, `telnet` and *File Transfer Protocol*, were thus used to remotely log on to other hosts, and copy files between hosts;
- ***TCP and UDP therefore predate modern distributed computing techniques.***

TCP, UDP and IP Stack



Coulouris, Dollimore, Kindberg and Blair, Distributed Systems: Concepts and Design Edn. 5 Pearson Education 2012

- **This is a simplified model of the TCP/UDP and IP protocol stack;**
- **Applications interface to TCP and UDP via an API, which provides access to the middleware that handles the TCP and UDP connections;**

Key Limitations of TCP and UDP

- High performance in carrying files and other bulk data was not a consideration when TCP and UDP were developed;
- TCP was developed to provide for “reliable” data transfers, ensuring that data (in packets) always arrived, and arrived in order, regardless of performance losses;
- UDP was developed to provide for “unreliable” data transfers, using a simpler protocol with slightly better performance than TCP, due to lesser overheads – applications had to manage reliability and in-order transfers;
- In a distributed computing environment, especially where high volumes of traffic must be carried, TCP and UDP are often too slow, as they were not optimised for high performance;
- *GridFTP, SCTP, and Fast TCP are alternatives;*

Advanced Grid Networking Technologies

- Primarily developed to support High Performance Computing [HPC] Grid environments with high capacity connectivity;
- Evolved from existing IP protocol suites;
- GridFTP
- SCTP
- Fast TCP

Overview

- Distributed communication over distant communication lines requires efficient protocols;
- TCP and UDP may not be sufficient;
- SCTP and FastTCP are extensions to standard transport protocols;
- Grid uses the GridFTP protocol as an overlay on the transport-layer protocols;

GridFTP

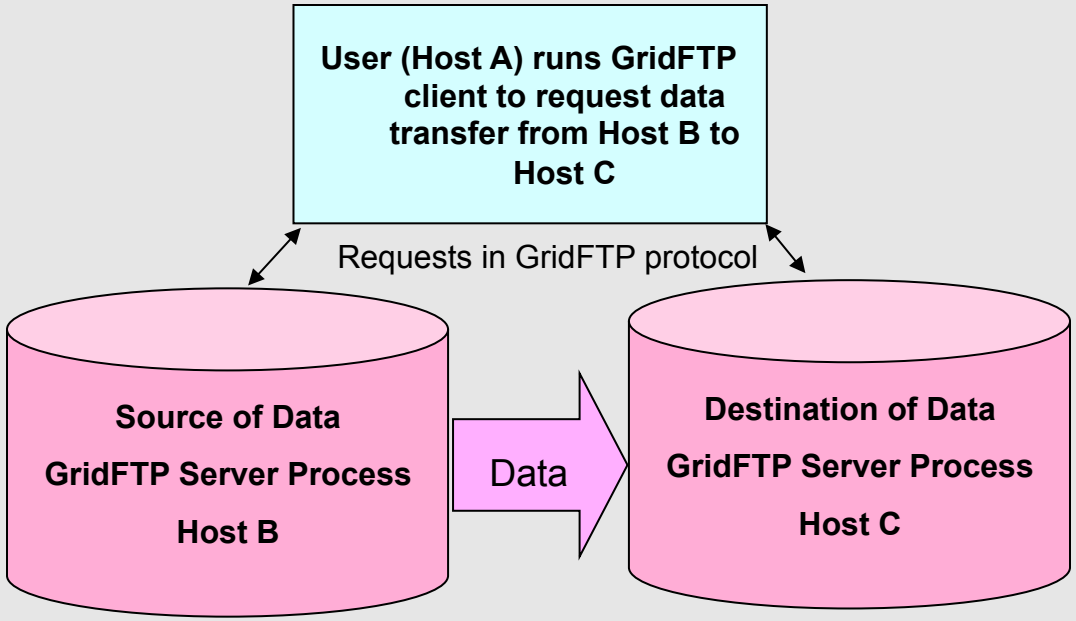
- A high-performance, reliable, secure data transfer protocol for high bandwidth wide area networks
- Interconnection of geographically dispersed grid resources
- Based on standard File Transfer Protocol

GridFTP

- Clients of the GridFTP protocol issue GridFTP-formatted requests to grid servers
- Standard GridFTP client called ‘GridFTP’ is non-existent
- TeraGrid project used the tgcp, globus-url-copy and uberftp as GridFTP client programs

GridFTP 3rd Party Transfers

- A third party transfer takes place between two GridFTP servers, rather than between the clients and servers



GridFTP Clients

Client requests in GridFTP format are authenticated by the server before being processed

uberFTP:

- A GridFTP client
- Has configurable TCP buffer size and number of parallel streams

UberFTP command list

- **Unix-like commands**
  - ls, cd, mkdir, rmdir, pwd, rm
- **Put "l" in front for "local" versions of commands**
  - lls, lcd, lmkdir, lrmdir, lpwd, lrm
- **put**
  - transfer from local host to remote host
- **get**
  - transfer from remote host to local host
- **mput, mget**
  - transfer multiple files between hosts
- **help**

GridFTP clients

globus-url-copy <source_url> <destination_url>

- command line interface
- -tcp-bs <size> | -tcp-buffer-size <size>
  - > specify the size (in bytes) of the buffer to be used by the underlying ftp data channels
- -p <parallelism> | -parallel <parallelism>
  - > specify the number of streams to be used in the ftp transfer

tgcp [gridFTP-server1:]file1 [gridFTP-server2:]file2

- command line interface
- friendly “scp-like” wrapper around globus-url-copy

SCTP

- Stands for ‘Stream Control Transmission Protocol’;
- Originally developed for Public Switched Telephone Networks (PSTNs) – to carry telephony signaling messages over IP networks;
- Is a reliable transport-layer protocol as an overlay on a connectionless protocol such as IP; Attempts to bridge the gap between TCP and UDP

SCTP

- **Is message-oriented**
- **Uses associations rather than connections**
- **Associations may have multiple parallel streams and are full-duplex**
- **Reliable transfer of segments of data is separated from data delivery**
  - Reliable transfer, unreliable connection.
- **Supports multi-streaming**

SCTP

Multi-streaming:

- **Provides for independent data delivery among streams**
- **Thus reduces the risk of head-of-line blocking among application objects**

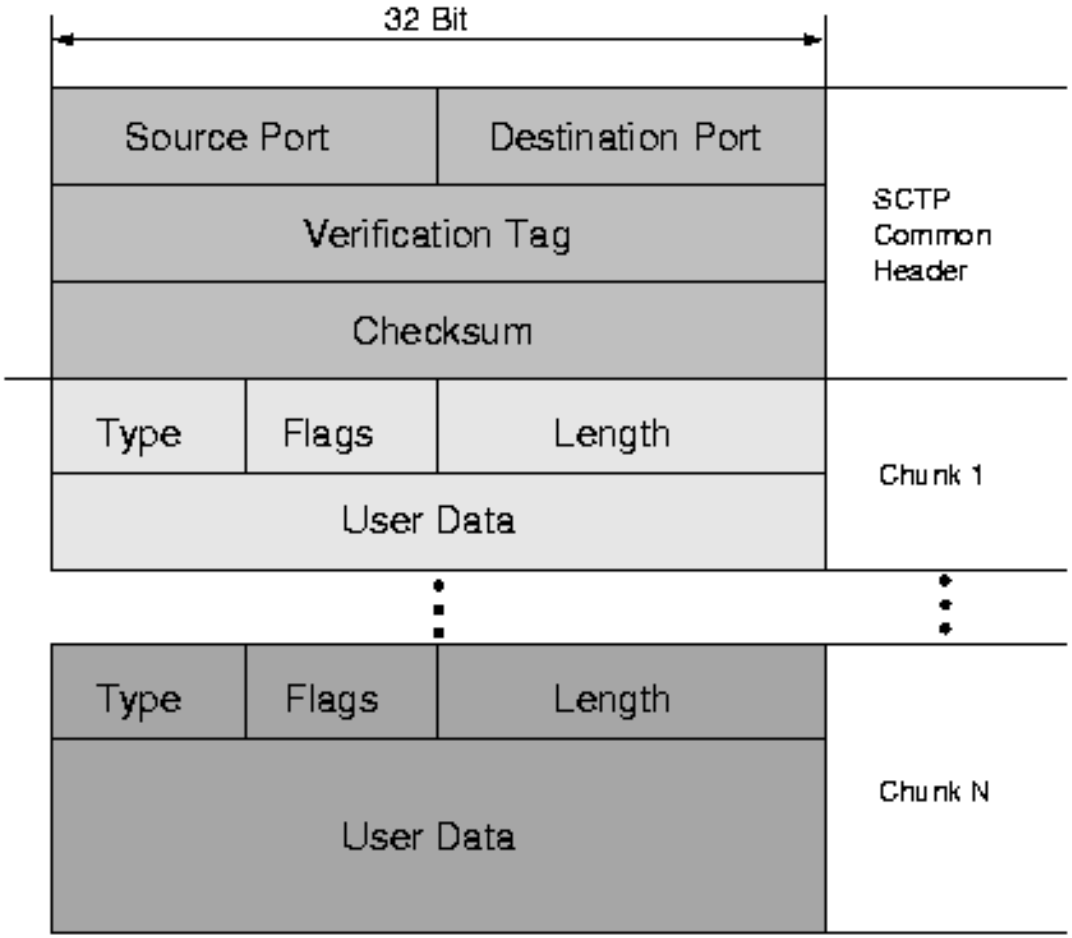
Multi-homing:

- **Provides redundancy at path-level**

4-way handshake rather than ‘3’:

- **Makes it resistant to SYN-based attacks**

SCTP


32 Bit
Source Port
Destination Port
Verification Tag
Checksum
SCTP Common Header
Type
Flags
Length
User Data
Chunk 1
Type
Flags
Length
User Data
Chunk N

SCTP properties

- **Slow Start and Congestion Avoidance phases like TCP:**
  - Selective Acknowledgement
  - Fast Retransmit
- **Provides partial data ordering**
- **Guarantees reliability in data transfer**
- **Improves on latency and throughput**

SCTP Congestion Control

- Uses 3 parameters: rwnd, cwnd and ssthresh
- cwnd: depicts amount of data to be sent rather than which data to be sent
- SCTP performs congestion control when cwnd > ssthresh
- The cwnd size during congestion increases by 1 Message Transmission Unit/Round Trip Time

FastTCP

- Purpose: to overcome congestion control shortcomings of TCP
- Quicker response to network congestion
- Uses queuing delays and packet loss data to control congestion

FastTCP

FastTCP congestion control mechanism is broken into 4 components:

- Data Control
- Window Control
- Burstiness Control
- Estimation

The above 4 components are an overlay on the TCP protocol processing layer

FastTCP

- Data control component determines which data packets need to be transmitted
- Window control determines how many packets to transmit – works on a Round Trip Time scale
- Burstiness control determines when to transmit packets – works on smaller time scales i.e. more frequently
- Estimation component provides information to assist the above components in decision-making

FastTCP

Estimation Component:

- **Provides estimations of various input parameters to the other three decision-making components**
- **When a positive ack is received for a packet, it calculates the RTT, and updates the average queuing delay and the minimum RTT**
- **When a negative ack (NACK) is received, signaled by timeout or three repeated acks, it generates a loss value for this packet**

FastTCP

Window Control:

- **Determines the congestion window size based on feedback from Estimation component**
- **Same algorithm for incrementing congestion window size is used each time, regardless of the state of the sender (unlike TCP)**

Reading and References

- Gross, K,, *Overview of GridFTP in the Open Science Grid*, URI: https://twiki.opensciencegrid.org/bin/view/Documentation/StorageGridFTP
- Mandrichenko,I., *GridFTP Protocol Improvements*, URI: http://www.ogf.org/documents/GFD.21.pdf
- Wei, D.X., et al, *FAST TCP: Motivation, Architecture, Algorithms, Performance*, URI: http://netlab.caltech.edu/publications/FAST-ToN-final-060209-2007.pdf
- Stewart, R., *Stream Control Transmission Protocol*, IETF RFC 4960, URI: https://tools.ietf.org/html/rfc4960
- http://www.cs.wisc.edu/~raj/sctp
- http://monalisa.cacr.caltech.edu

REVISION MATERIAL

Encapsulation in Protocols [TCP/IP] - Stallings

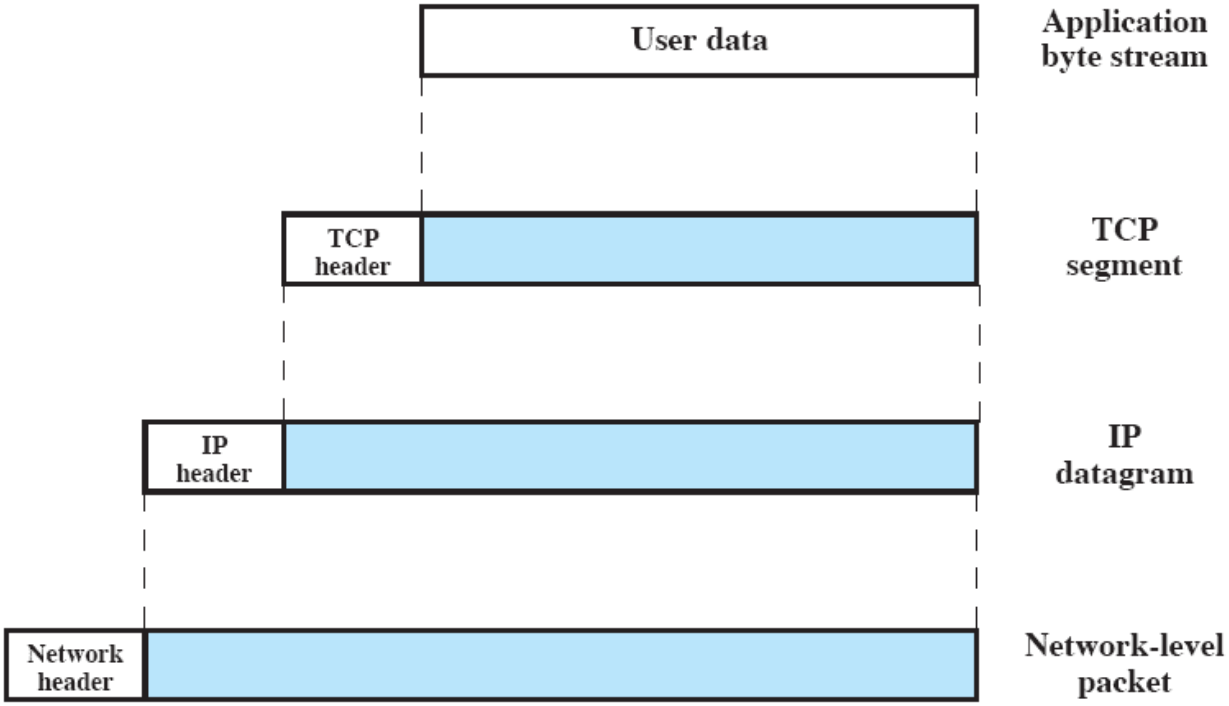
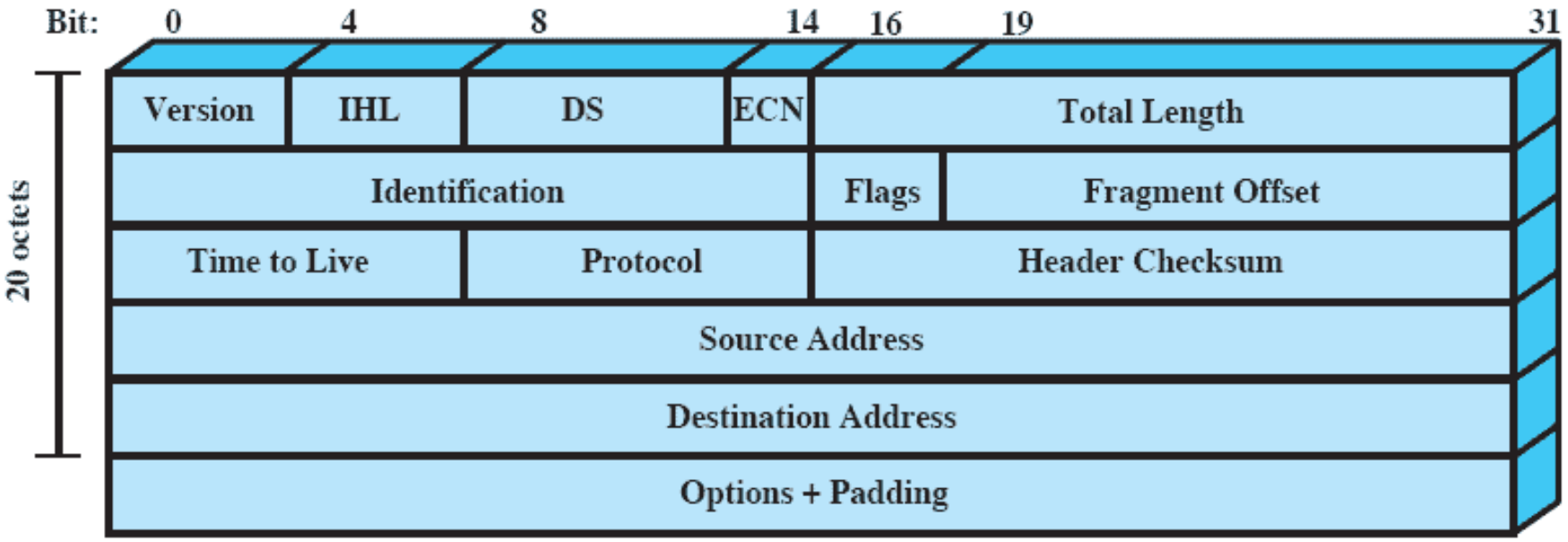


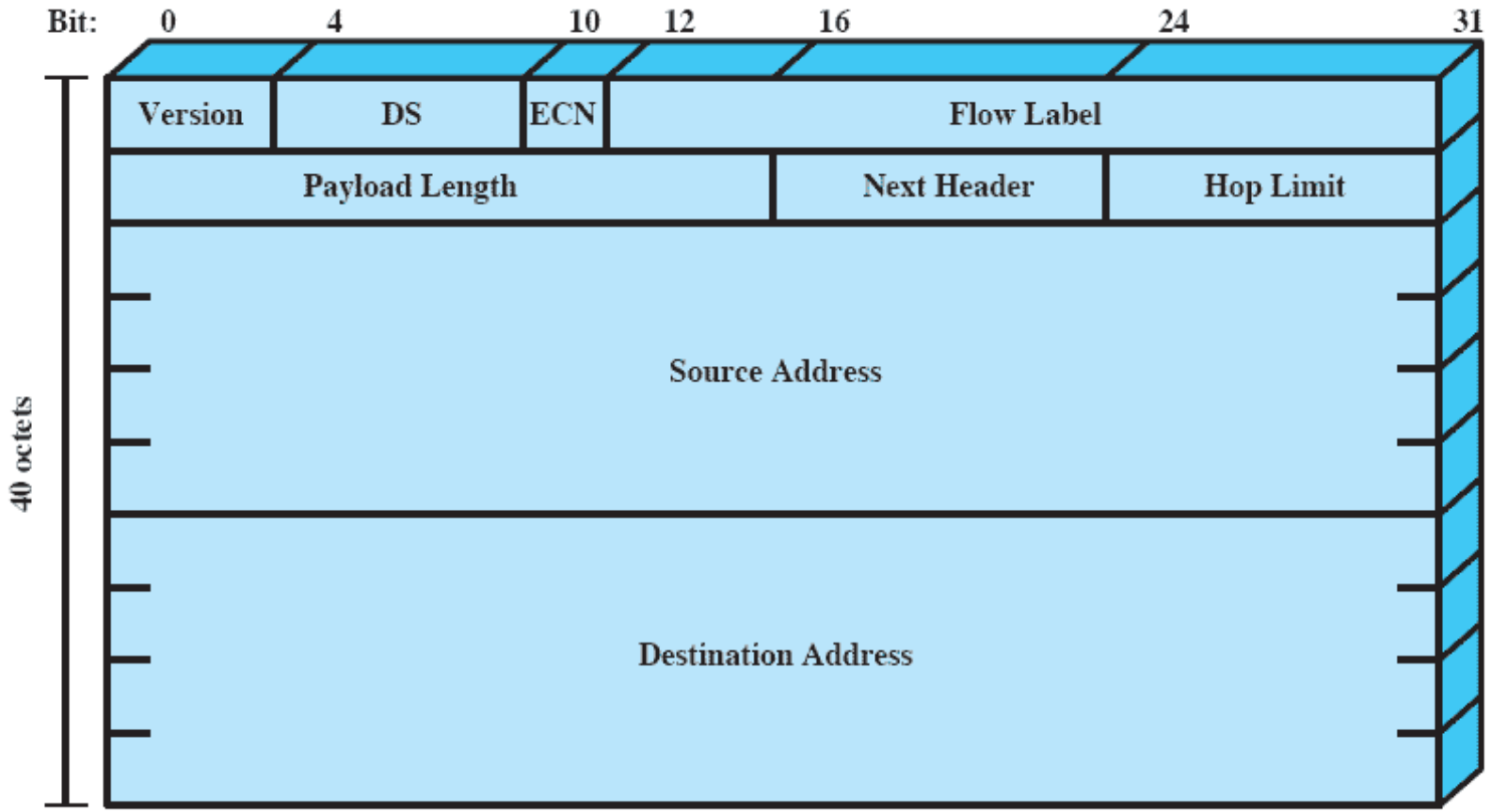
Figure 17.5 Protocol Data Units (PDUs) in the TCP/IP Architecture

IPv4 - Stallings



(a) IPv4 Header

IPv6 - Stallings



(b) IPv6 Header

DS = Differentiated services field
ECN = Explicit congestion notification field

Note: The 8-bit DS/ECN fields were formerly known as the Type of Service field in the IPv4 header and the Traffic Class field in the IPv6 header.