B38CN:Introduction to Communications and Networks

Chapter 6: The Transport Layer

Instructor： Dr. Meaad Fadhel

2022.03.21

School of Telecommunication Engineering
Xidian University

Developed from Prof. Cheng-Xiang Wang slides

先 进 网 络 技 术 实 验 室

Advanced Networking Technology Lab

目录 CONTENTS

PART
01

The Transport Service

The Transport Layer

- The **heart** of the whole protocol hierarchy. Without the transport layer, the whole concept of **layered protocols** would make little sense.
- A true **end-to-end** layer, all the way from the source to the destination.
- **Main task:** Provides efficient, reliable, and cost-effective data transport from the source machine to the destination machine, independently of the physical network or networks currently in use.
- The **lower 3 layers** are **chained**, in which the protocols are between each machine and its immediate neighbors (routers), since the ultimate source and destination machines may be separated by many routers. The **upper layer(s)** are **end-to-end.**
- The **bottom 4 layers** are **transport service provider**. The **upper layer(s)** are the **transport service user.** This **distinction** of provider versus user puts the transport layer in a key position, since it forms the **major boundary** between the provider and user of the reliable data transmission service.

6.1 The Transport Service

- Provide efficient, reliable, and cost-effective service to its users, normally processes in the application layer, by using the services provided by the network layer.

- **Transport entity:** the hardware and/or software within the transport layer that does the work.
  - Can be located in the operating system kernel, in a separate user process, in a library package bound into network applications, or conceivably on the network interface card.

- **Two types** of transport service: connection-oriented and connectionless.

Relationship of the Network, Transport, and Application Layer

- **TPDU:** Transport Protocol Data Unit

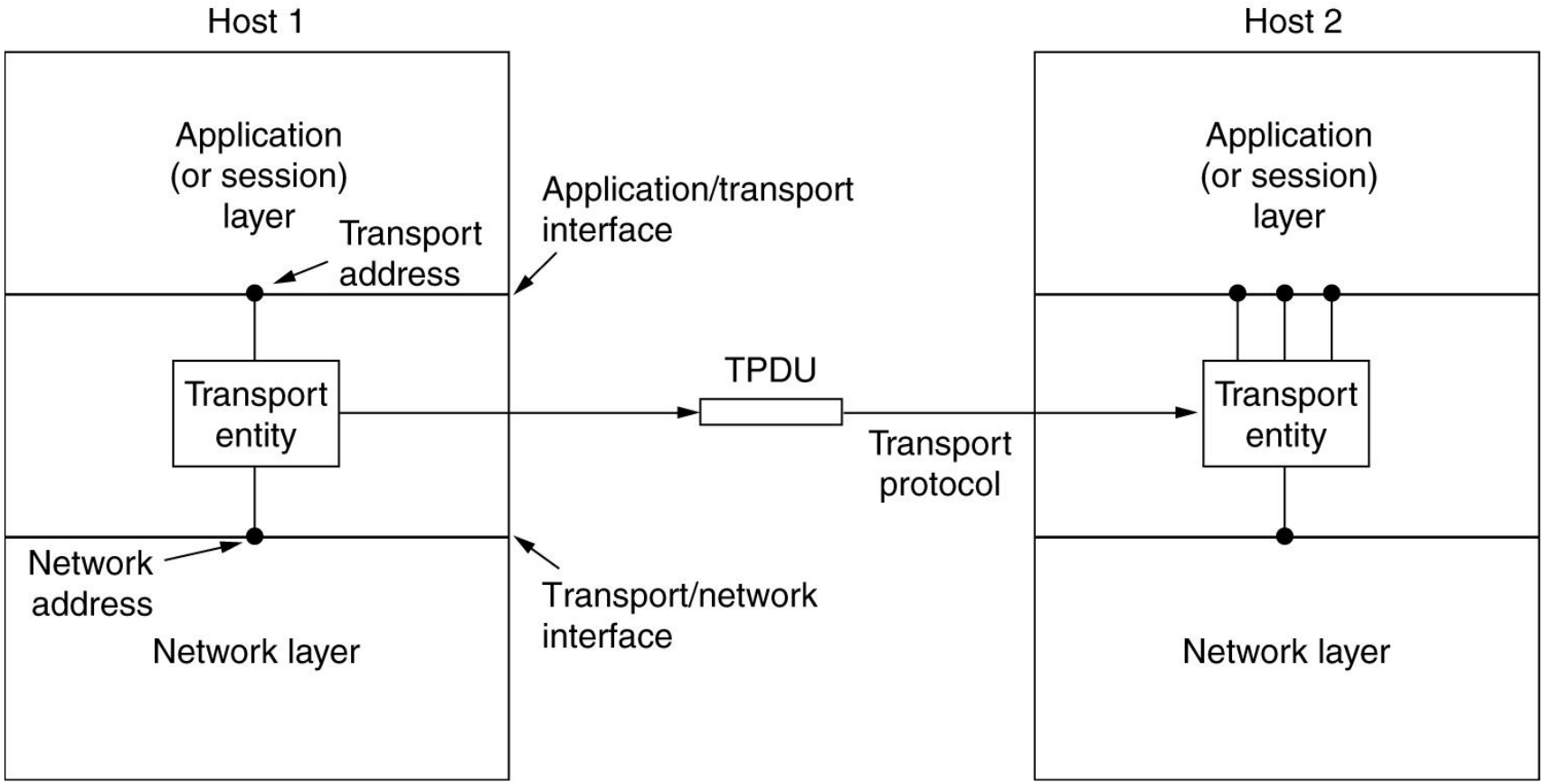

Fig. 6.1: The network, transport, and application layers.

Network Layer and Transport Layer

- The transport layer service is very similar to the network layer service.
- The **network layer** mostly runs on the **routers**, which are operated by the carrier (at least for a WAN). The users have no real control over the network layer, so they cannot solve the problem of poor service by using better routers or putting more error handling in the data link layer.
- If we want to develop applications that are independent of the particular services offered by a carrier, we'll have to devise a standard communication interface and implement that interface at the client's sites. The **transport layer** contains such implementations. The transport layer code runs entirely on the **users' machines**.
- In essence, the existence of the transport layer makes it possible for the transport service to be **more reliable** than the underlying network service.
- The transport layer fulfills the **key function** of isolating the upper layers from the technology, design, and imperfections of the subnet.

TPDUs, Packets, and Frames

- Messages sent by clients are encapsulated as **TPDUs** (exchanged by the transport layer).
- TPDUs are contained in **packets** (exchanged by the network layer).
- Packets are contained in **frames** (exchanged by the data link layer).

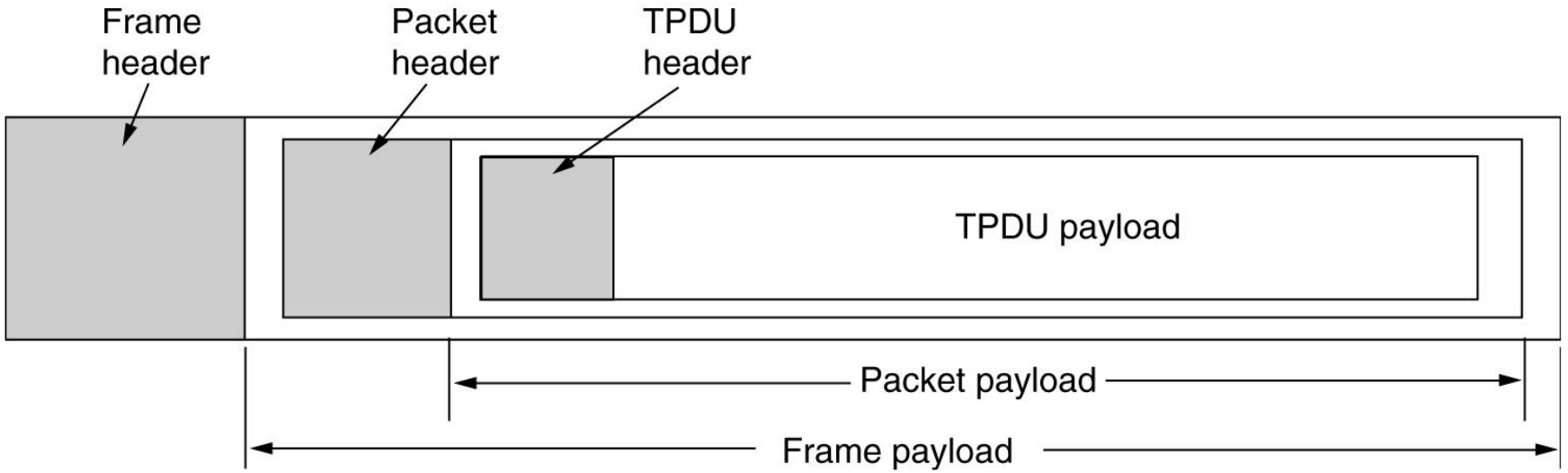

Fig. 6.2: The nesting of TPDUs, packets, and frames.

PART
02

The Internet Transport Protocols: UDP

6.2 The Internet Transport Protocols: UDP

- The Internet has **two main protocols** in the transport layer:
  - An **unreliable connectionless** protocol: User Datagram Protocol **(UDP).**
  - A **reliable connection-oriented** protocol: Transmission Control Protocol **(TCP).**
- **UDP** is basically just IP with a short header.
- UDP transmits **segments** consisting of an 8-byte header followed by the payload.
- UDP **does not do** flow control, error control, or retransmissions. All it **does** is provide an interface to the IP protocol with the added feature of demultiplexing multiple processes using the ports.
- **Application:** client-server interactions; **DNS** (the Domain Name system).

UDP Header

- **Source/Destination port:** identify the end points within the source and destination machines. With the port fields, the transport layer can deliver segments correctly.
- **UDP length:** includes the 8-byte header and the data.
- **UDP checksum** is **optional** and stored as 0 if not computed (a true computed 0 is stored as all 1s).

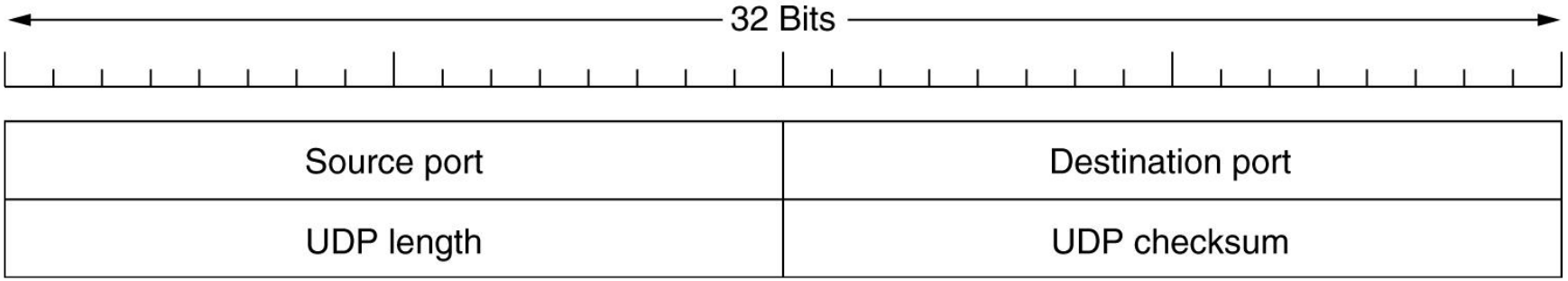

Fig. 6.3: The UDP header

PART 03

The Internet Transport Protocols: TCP

6.3 The Internet Transport Protocols: TCP

- Provides a **reliable connection-oriented byte stream** over an unreliable internetwork. Message boundaries are not preserved end to end.
- TCP can dynamically adapt to properties of the internetwork and be robust in the face of many kinds of failures.
- All TCP connections are **full duplex** and **point–to–point.** No support for multicasting or broadcasting.
- A TCP **TPDU** is called a **segment**, consisting of a fixed **20-byte header** (plus an optional part) followed by data bytes. Two limits restrict the **segment size:**
  - Maximum total length (TCP header + data): 65,515 bytes.
  - A segment is fragmented by the network layer when it is larger than the network's **maximum transfer unit (MTU)—**in practice 1500 bytes.

The TCP Service Model

- TCP service is obtained by both the sender and receiver creating end points, called **sockets**.
- Each socket has a socket number (address, 48 bits) consisting of the IP address of the host and **16-bit number** local to that host, called a **port**.
- Port numbers below 1024 are called **well-known ports** and are reserved for standard service

Port	Protocol	Use
21	FTP	File transfer
23	Telnet	Remote login
25	SMTP	E-mail
69	TFTP	Trivial File Transfer Protocol
79	Finger	Lookup info about a user
80	HTTP	World Wide Web
110	POP-3	Remote e-mail access
119	NNTP	USENET news

Fig. 6.4: Some assigned ports.

The TCP Segment Header

- After header options, up to 65535-20 (IP header) -20 (TCP header)=**65495** data bytes may follow. Segments without any data are legal and are used for ACK and control massages.
- **Acknowledgment number:** specifies the next byte expected; ACK=1 valid; ACK=0 ignored.
- **SYN:** synchronization; is for connection setup. **FIN:** finish; is for connection release. **RST:** reset a connection.

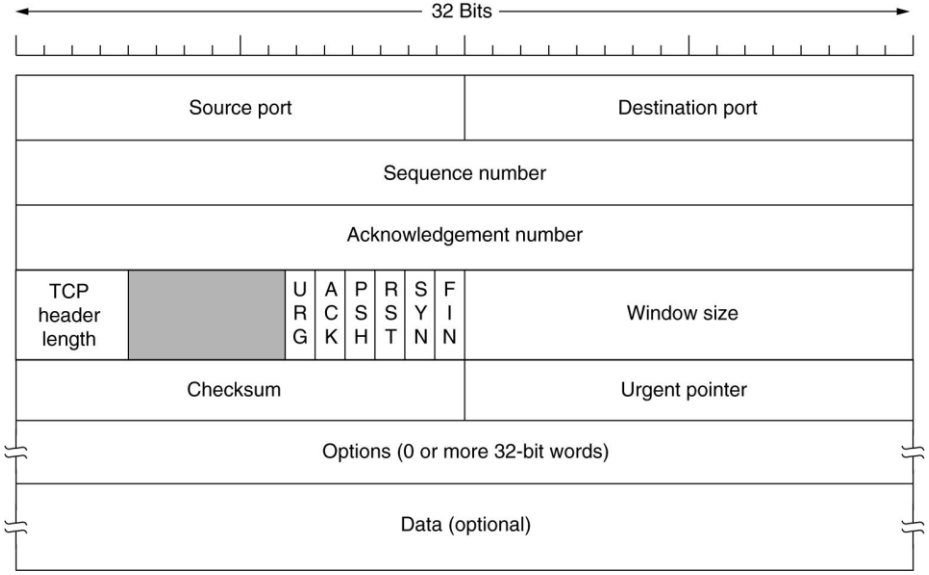

Fig. 6.5: TCP Header.

谢谢大家！

Thank you!