Introduction to Network Security

Chapter 2

Network Protocols

Topics

- Protocol Specifications
- Protocol Addresses
- Protocol Headers

A TCP/IP Tutorial

Status of this Memo

This RFC is a tutorial on the TCP/IP protocol suite, focusing particularly on the steps in forwarding an IP datagram from source host to destination host through a router. It does not specify an Internet standard. Distribution of this memo is unlimited.

Table of Contents

1. Introduction

This tutorial contains only one view of the salient points of TCP/IP, and therefore it is the "bare bones" of TCP/IP technology. It omits the history of development and funding, the business case for its use, and its future as compared to ISO OSI. Indeed, a great deal of technical information is also omitted. What remains is a minimum of information that must be understood by the professional working in a TCP/IP environment. These professionals include the systems administrator, the systems programmer, and the network manager.

This tutorial uses examples from the UNIX TCP/IP environment, however the main points apply across all implementations of TCP/IP.

Note that the purpose of this memo is explanation, not definition. If any question arises about the correct specification of a protocol, please refer to the actual standards defining RFC.

Workgroup:	Internet Engineering Task Force
Internet-Draft:	draft-ietf-tcpm-rfc793bis-25
Obsoletes:	793, 879, 2873, 6093, 6429, 6528, 6691 (if approved)
Updates:	5961, 1122 (if approved)
Published:	7 September 2021
Intended Status:	Standards Track
Expires:	11 March 2022

Author:	W. Eddy, Ed.
	MTI Systems

Transmission Control Protocol (TCP) Specification

Abstract

This document specifies the Transmission Control Protocol (TCP). TCP is an important transport layer protocol in the Internet protocol stack, and has continuously evolved over decades of use and growth of the Internet. Over this time, a number of changes have been made to TCP as it was specified in RFC 793, though these have only been documented in a piecemeal fashion. This document collects and brings those changes together with the protocol specification from RFC 793. This document obsoletes RFC 793, as well as RFCs 879, 2873, 6093, 6429, 6528, and 6691 that updated parts of RFC 793. It updates RFC 1122, and should be considered as a replacement for the portions of that document dealing with TCP requirements. It also updates RFC 5961 by adding a small clarification in reset handling while in the SYN-RECEIVED state. The TCP header control bits from RFC 793 have also been updated based on RFC 3168.¶

RFC EDITOR NOTE: If approved for publication as an RFC, this should be marked additionally as "STD: 7" and replace RFC 793 in that role.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

Protocol Specifications

- Open vs. Closed
- Specification methods
  - –English descriptions
  - –Flow & timing diagrams
  - –Open to interpretation
- Implementation flaws

Network Standards

- Specifies
  - –Services provided
  - –Services expected
  - –Functions provided
  - –Protocol and packet formats
  - –Timing and sequence of the packets

TABLE OF CONTENTS

Workgroup:	Internet Engineering Task Force
Internet-Draft:	draft-ietf-tcpm-rfc793bis-25
Obsoletes:	793, 879, 2873, 6093, 6429, 6528, 6691 (if approved)
Updates:	5961, 1122 (if approved)
Published:	7 September 2021
Intended Status:	Standards Track
Expires:	11 March 2022

Author:	W. Eddy, Ed.
	MTI Systems

Transmission Control Protocol (TCP) Specification

Abstract

This document specifies the Transmission Control Protocol (TCP). TCP is an important transport layer protocol in the Internet protocol stack, and has continuously evolved over decades of use and growth of the Internet. Over this time, a number of changes have been made to TCP as it was specified in RFC 793, though these have only been documented in a piecemeal fashion. This document collects and brings those changes together with the protocol specification from RFC 793. This document obsoletes RFC 793, as well as RFCs 879, 2873, 6093, 6429, 6528, and 6691 that updated parts of RFC 793. It updates RFC 1122, and should be considered as a replacement for the portions of that document dealing with TCP requirements. It also updates RFC 5961 by adding a small clarification in reset handling while in the SYN-RECEIVED state. The TCP header control bits from RFC 793 have also been updated based on RFC 3168.

RFC EDITOR NOTE: If approved for publication as an RFC, this should be marked additionally as "STD: 7" and replace RFC 793 in that role.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

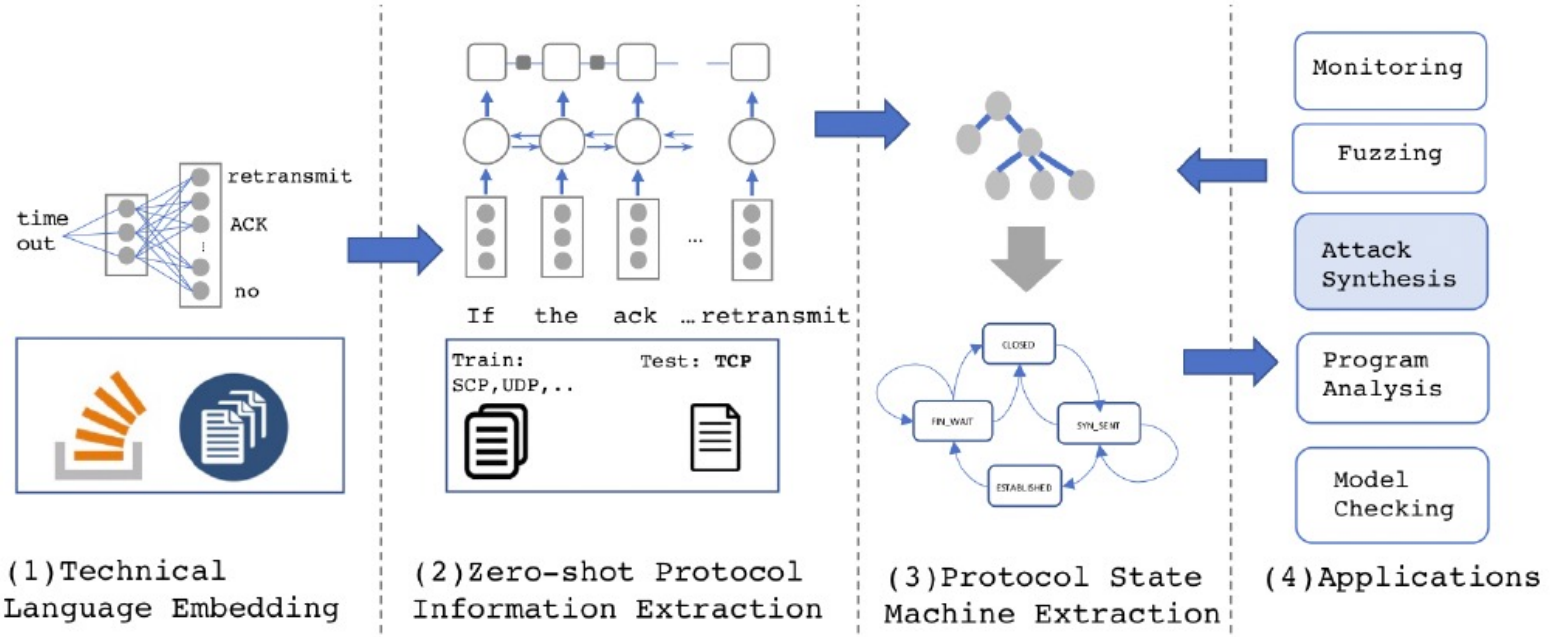
Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or

Table of Contents

Fragmentation, transmission and reassembly across a local network which is invisible to the internet protocol module is called internet fragmentation and may be used [6].

The internet fragmentation and reassembly procedure needs to be able to break a datagram into an almost arbitrary number of pieces that can be later reassembled. The receiver of the fragments uses the identification field to ensure that fragments of different datagrams are not mixed. The fragment offset field tells the receiver the position of a fragment in the original datagram. The fragment offset and length determine the portion of the original datagram covered by this fragment. The more-fragments flag indicates (by being reset) the last fragment. These fields provide sufficient information to reassemble datagrams.

If the network receives a DETACH REQUEST message before the ongoing identification procedure has been completed, the network shall *abort the identification procedure* and shall process the detach procedure



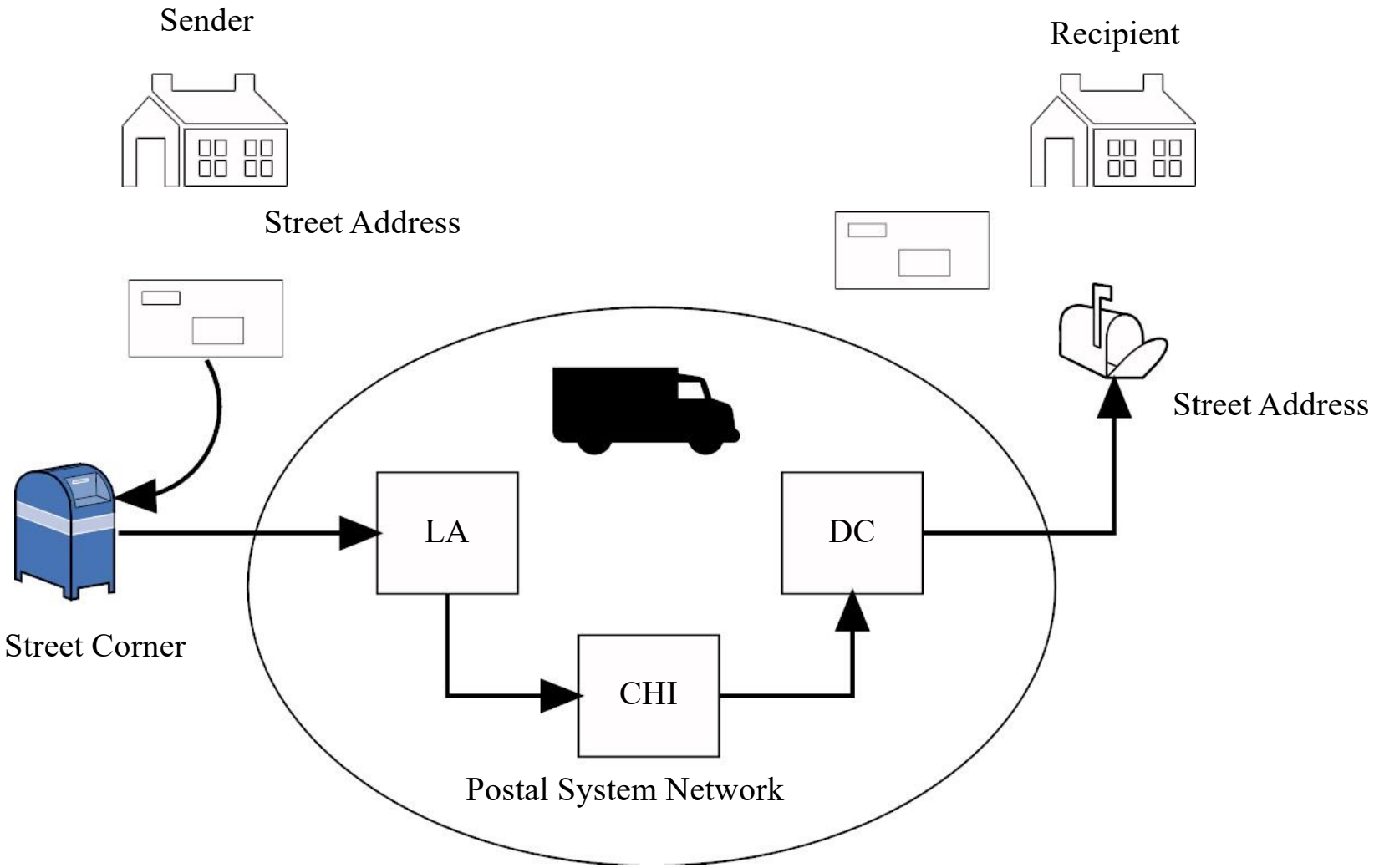
Automated Attack Synthesis by Extracting Finite State Machines from Protocol Specification Documents

Maria Leonor Pacheco*, Max von Hippel†, Ben Weintraub†, Dan Goldwasser*, Cristina Nita-Rotaru†
*Purdue University, West Lafeyette, IN, USA, {pachecog,dgoldwas}@purdue.edu
†Northeastern University, Boston, MA, USA, {vonhippel.m,weintraub.b,c.nitarotaru}@northeastern.edu

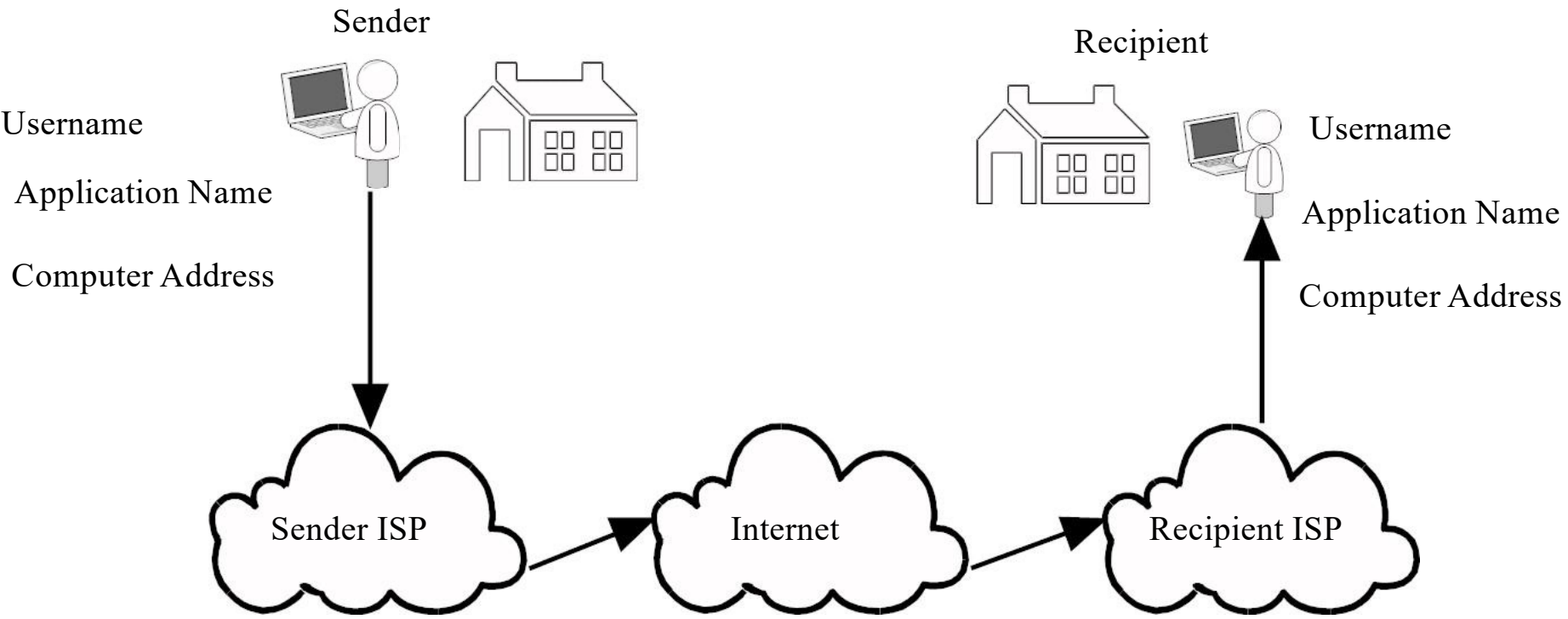
Standards Organizations

- American National Standards Institute (ANSI): ANSI is a private organization whose membership is made up of professional societies, government groups and other associations. They develop standards that help groups complete in the global market.
- Institute of Electrical and Electronics Engineers (IEEE): IEEE is an international professional society that creates international standards in many different areas.
- International Standards Organization (ISO): A group whose membership is standards committees from across the world. ANSI represents the United States on ISO.
- International Telecommunications Union-Telecommunications Standards Sector (ITU-T): A group created by the United Nations that creates standards primarily for the phone system.
- Internet Engineering Task Force (IEFT): This group develops standards for the Internet and consists of members from various organizations and is open to any person that has an interest.

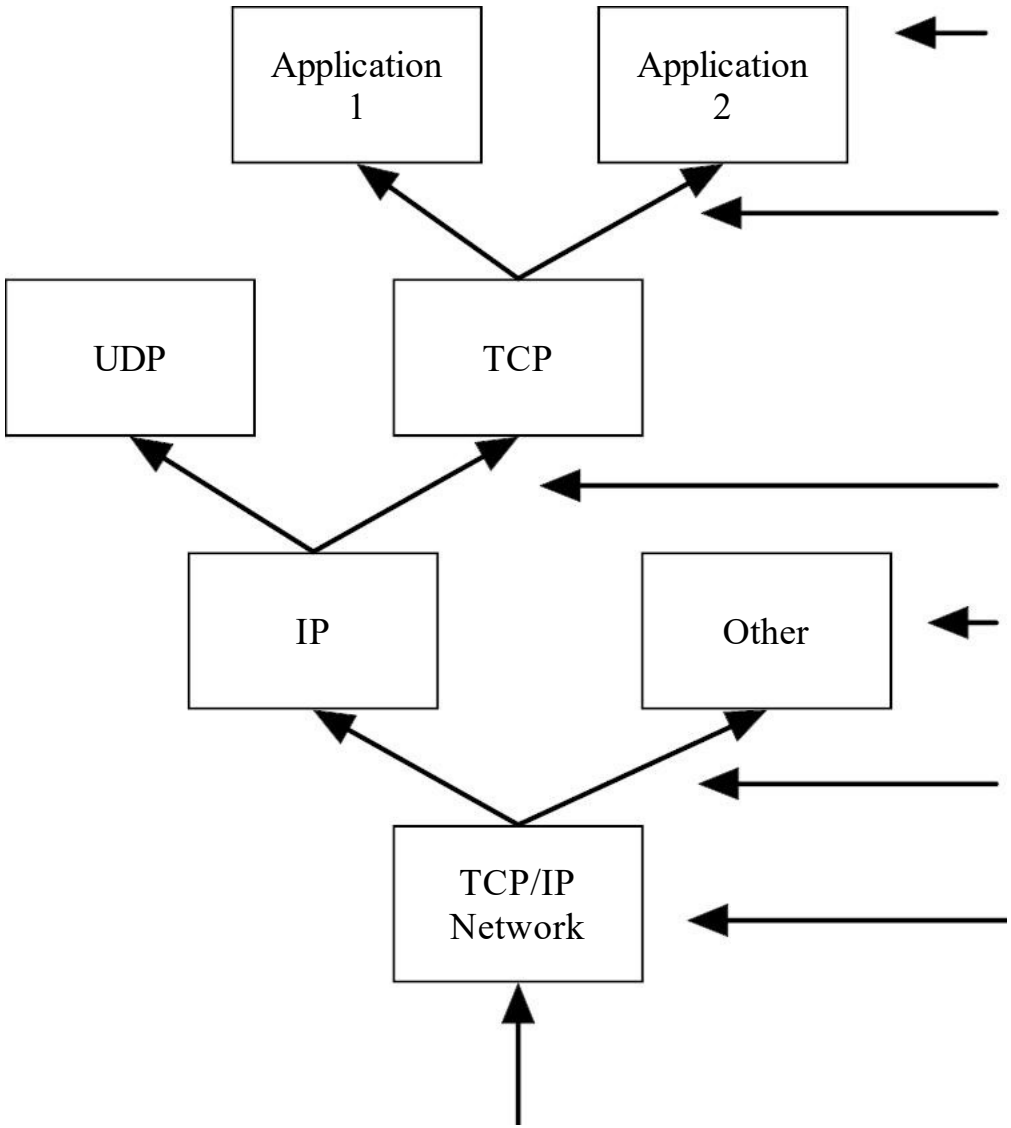
Addressing


Sender
Recipient
Street Address
Street Address
LA
DC
CHI
Street Corner
Postal System Network

Addressing


Sender
Recipient
Username
Application Name
Computer Address
Username
Application Name
Computer Address
Sender ISP
Internet
Recipient ISP

Addresses



Application user address:
identifies items within the application

Application port number:
Identifies the instance of application

Transport Protocol type:
Identifies the transport layer used

Global Network Layer Address:
Identifies the device

Protocol type:
Identifies which protocol is carried in the packet

Hardware Address:
Identifies which device on the physical network

Address Assignment - How

- Static
  - –Configuration
  - –Built in
- Dynamic
  - –Protocol discovery
  - –User provided

Hardware address assignment

- Hardware
  - –Vendor assigned
  - –Address used as a filter
  - –Address can be changed

IP address assignment

- Global address allocation
- Address assignment
  - –Protocol based (DHCP)
  - –Static
  - –Locally controlled
- Addresses can be changed

Application address assignment

- Port Number (much less control)
  - –Well know ports
  - –Protocol based discovery
  - –Configuration based
  - –User input based

Hostname assignment

- Often political and/or commercially driven
- Assignment via central authority
- Protocol to find the IP address given a name (DNS)

Protocol Headers

- Fixed packet type
  - –Easy to parse
  - –Limited functionality
- Freeform type
  - –Harder to parse
  - –Easy to extend

Fixed packet header

Fixed:
- Addresses (Layer addresses and payload type)
- Control data
- Header data
-

Options:
- Extended fixed data
- Optional control data
- Optional Payload control

Payload: Content is not a concern of the header

Trailer:
Optional field often used for error control

Transport Control Protocol (TCP)

- Each connection is uniquely identified by the 4–tuple:
  - **(SrcPort, SrcIPAddr, DsrPort, DstIPAddr)**
- Sliding window and flow control
  - **acknowledgment, SequenceNum, AdvertisedWindow**

TCP Header

Offsets Octet		0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0				NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size														
16	128	Checksum																Urgent pointer (if URG set)															
20 ...	160 ...	Options (if Data Offset > 5, padded at the end with "0" bytes if necessary) ...																															

Freeform header

<Start Header>
<Data type = application 7>
<Data length = 400>
<Data encoding = ASCII>
</End Header>
<Start Data>
(the data)
</End Data>