Introduction To TCP/IP
and
The Internet
Part III

DECUS Europe, Fall 1994
Friday, September 16 1994

Vladimir Sukonnik
Process Software Corporation
sukonnik@process.com

PART III

PROTOCOLS, LAYERING, AND NETWORK LEVEL SERVICES

The Problem

- Communication is difficult
- Many details to handle
  - addressing
  - routing
  - flow control
  - error detection and correction
  - out of order delivery
  - duplicated packets
  - multiple application programs
  - multiple protocols

The Solution

- Divide the problem into pieces
- Solve subproblems separately
- Combine into integrated whole

Protocol Layering

- Conceptual separation of functionality
- Each layer solves one part of the communication problem
- Intended primarily for designers
- End users may be aware of layering

Examples Of Layering

- ISO 7–layer model
- Internet reference model

ISO 7–Layer Model

- Application
- Presentation
- Session
- Transport
- Network
- Data Link
- Physical (hardware)

Internet Reference Model

- Application
- Transport
- Internet
- Network interface
- Physical (hardware)

Physical Hardware Layer

- Defines electrical signals used in communication (e.g. voltages on wires between computer and packet switch)

Network Interface Layer

- Defines communication between computer and network hardware
- Isolates details of physical (hardware) addressing
- Usually implemented by operating system

Internet Layer

- Usually implemented by experts
- Code is part of the operating system
- Provides machine to machine communication
- Offers unreliable, connectionless datagram delivery

Transport Layer

- Usually implemented by experts
- Code is part of the operating system
- Provides connection from application program to application program
- Often handles reliability, flow control

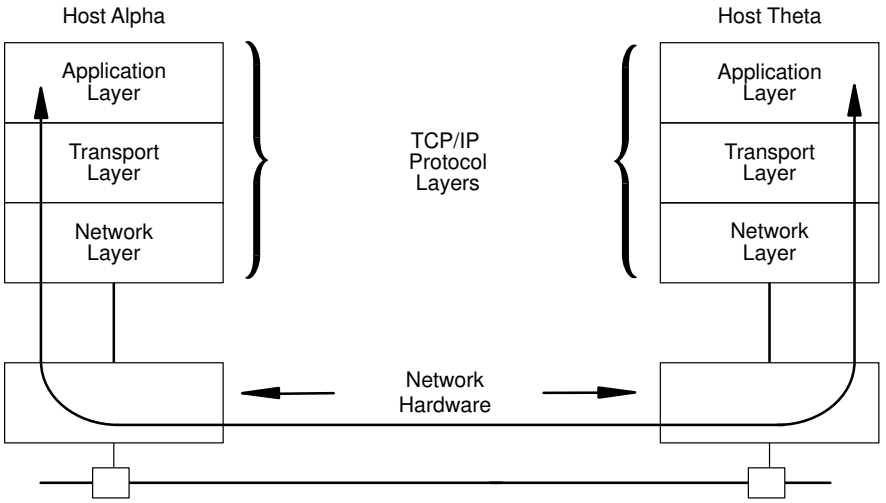
Application Layer

- Implemented by application programs (user level processes)
- Many application–specific protocols
- Built on top of transport layer

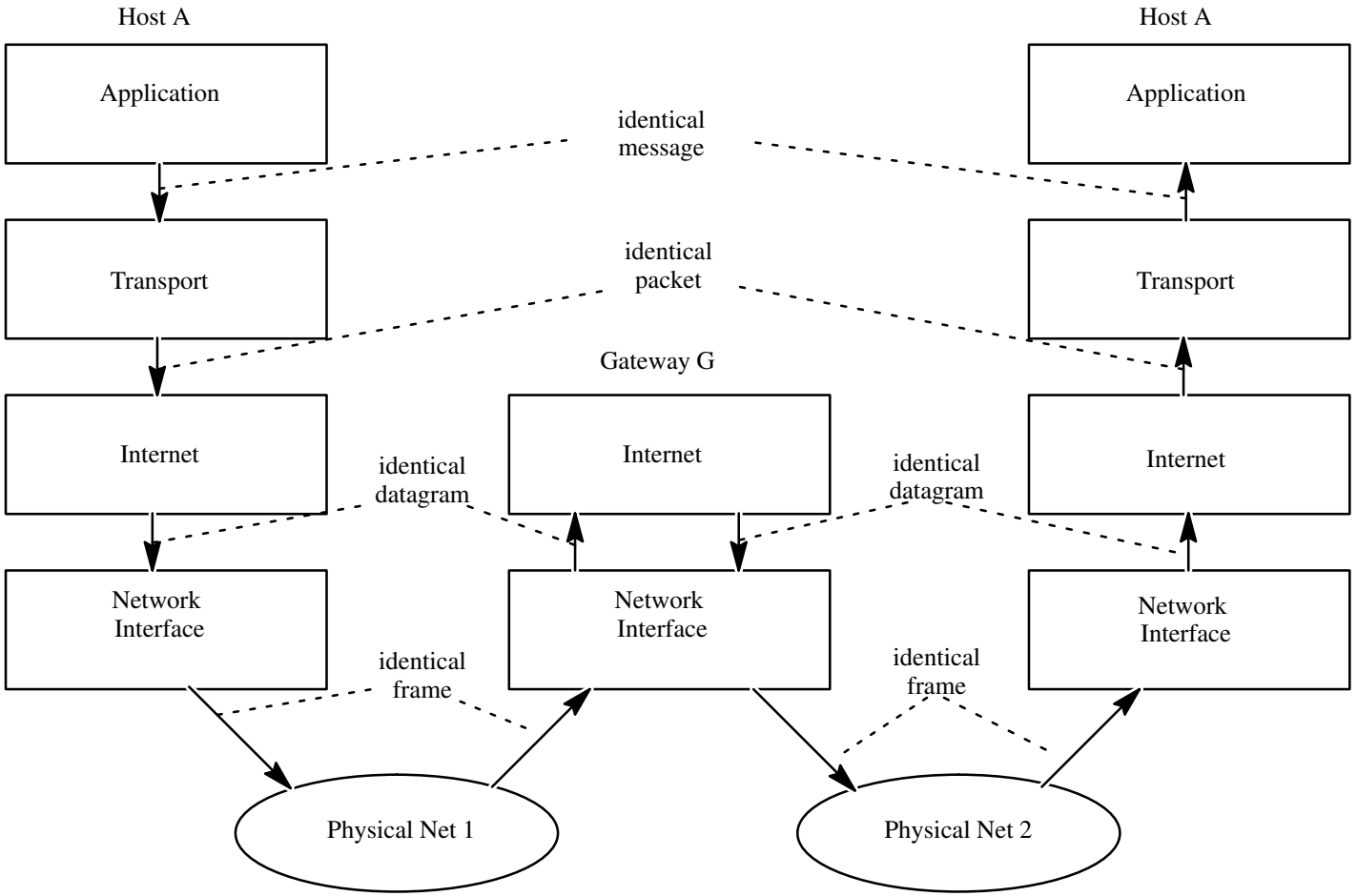
The Layering Principal

Software implementing layer n at the destination receives exactly the message sent by software implementing layer n at the source

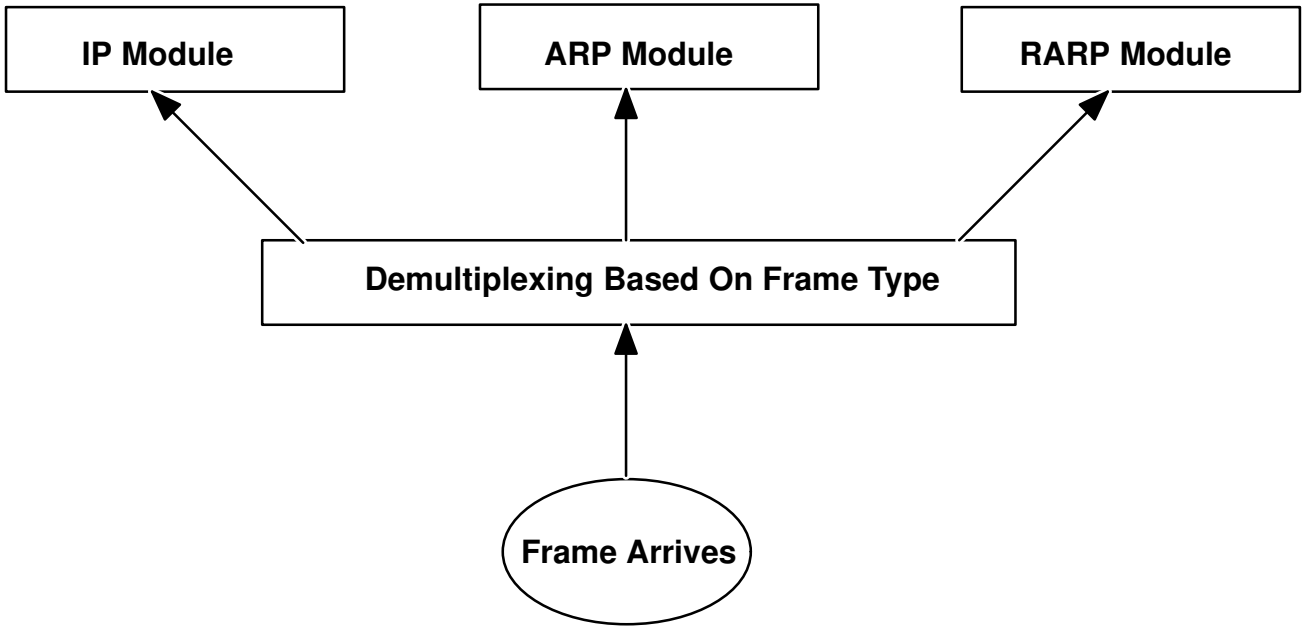
How Data Passes Through Protocol Layers



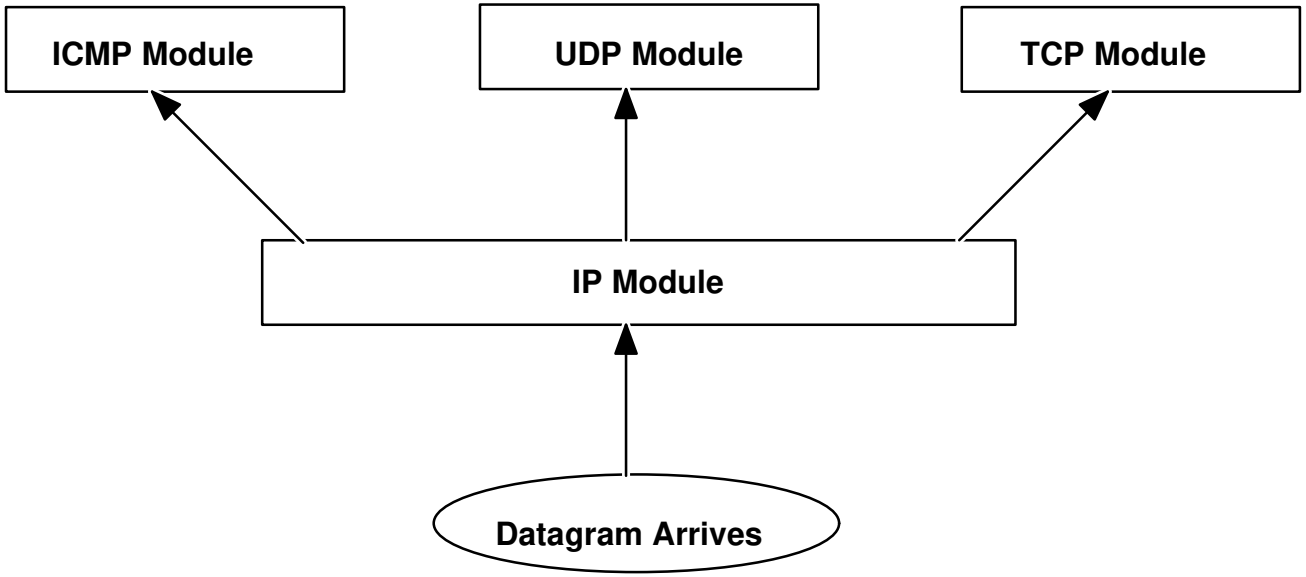
Layering In An Internet



Demultiplexing Among Software Modules At Each Layer



Demultiplexing Among Software Modules At Each Layer (continued)



Interface Between Internet Protocol Software In the Operating System and Application Programs

- Operating system specific
- Defines exactly how application programs interact with protocol software
- Called the socket interface in many systems
- System may have library of procedures that simulate socket interface

User Datagram Protocol (UDP)

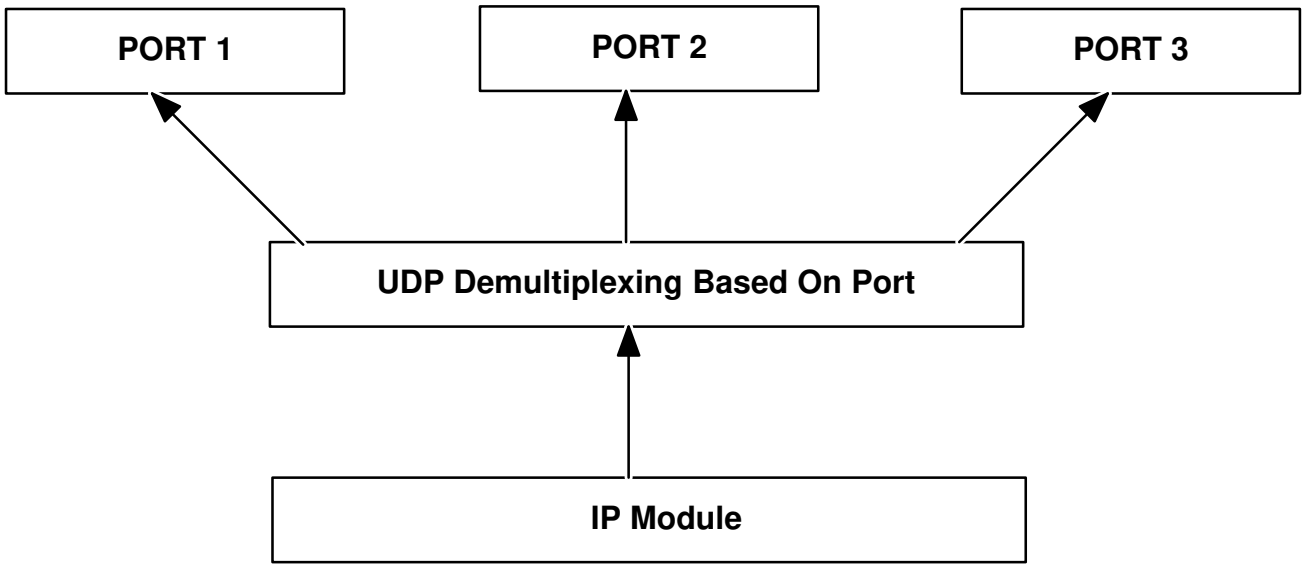
- Provides application programs with ability to send and receive datagrams
- Provides for multiple, concurrent application programs on a single machine
- Includes (optional) checksum for data transferred
- Offers unreliable, connectionless transport

User Datagram Protocol Details

- Defines format of messages application programs send
- Each message includes small integer protocol port numbers to distinguish among multiple application programs
- Messages are encapsulated in IP datagrams for delivery

UDP Message Format

Demultiplexing Based On Protocol Port Number



Transmission Control Protocol

- Major transport service in the Internet
- Reliable stream transfer
- Full duplex connection
- Flow control
- Uses IP for datagram transmission

Transmission Control Protocol Details

- Allows sender to generate a stream of bytes in convenient chunks
- Divides stream into small segments for transmissions
- Sends each segment in IP datagram
- Receiving TCP returns acknowledgement upon successful receipt of data
- Sender starts timer after segment sent and retransmits unless positive acknowledgement arrives

TCP Retransmission

- Designed for Internet environment
  - delays on one connection vary over time
  - delays vary widely between connections
- Fixed value for timeout will fail
  - waiting too long introduces unnecessary delay
  - not waiting long enough wastes network bandwidth with unnecessary retransmission
- Retransmission strategy must be adaptive

Adaptive Retransmission

- TCP keeps estimate of round –trip time on each connection
- Round–trip estimate derived from observed delay between sending segment and receiving acknowledgement
- Timeout for retransmission based on current round–trip estimate
- Heuristics cam sometimes fail (e.g. round–trip delay changes quickly)

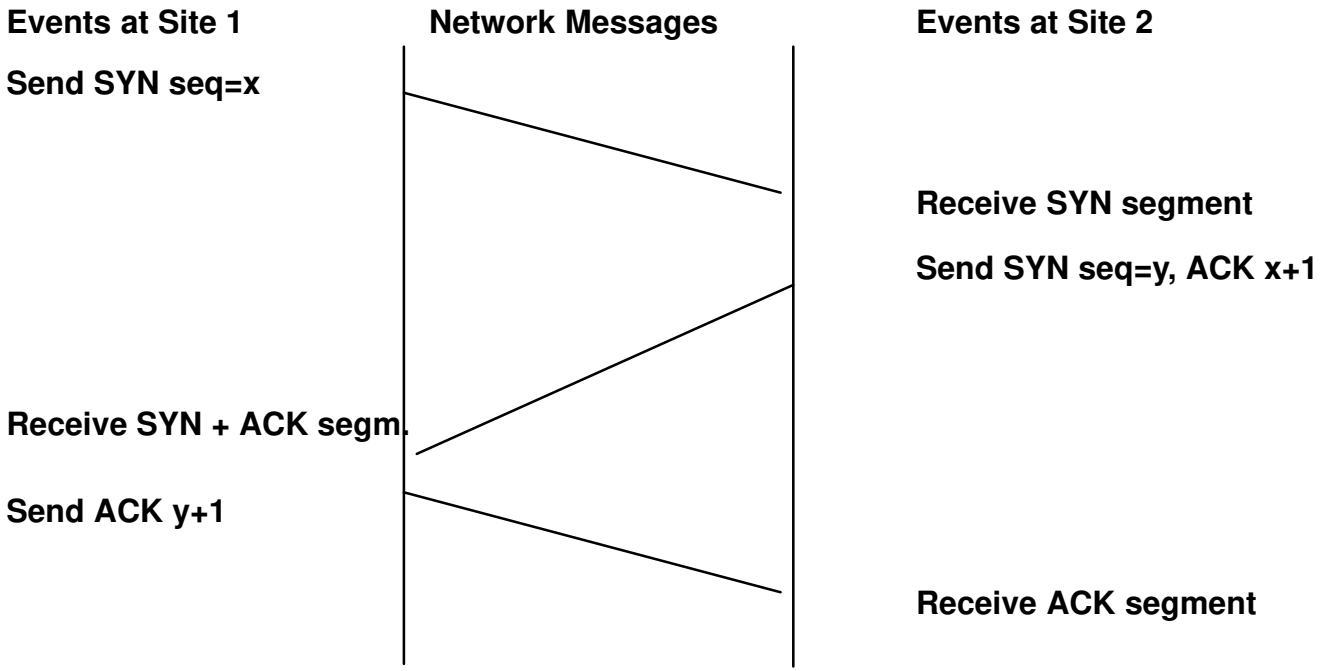
TCP Details

- Segment contains checksum for data being sent
- Receiver acknowledges highest byte received, not each specific segment
- Protocol port numbers used to distinguish among multiple application programs

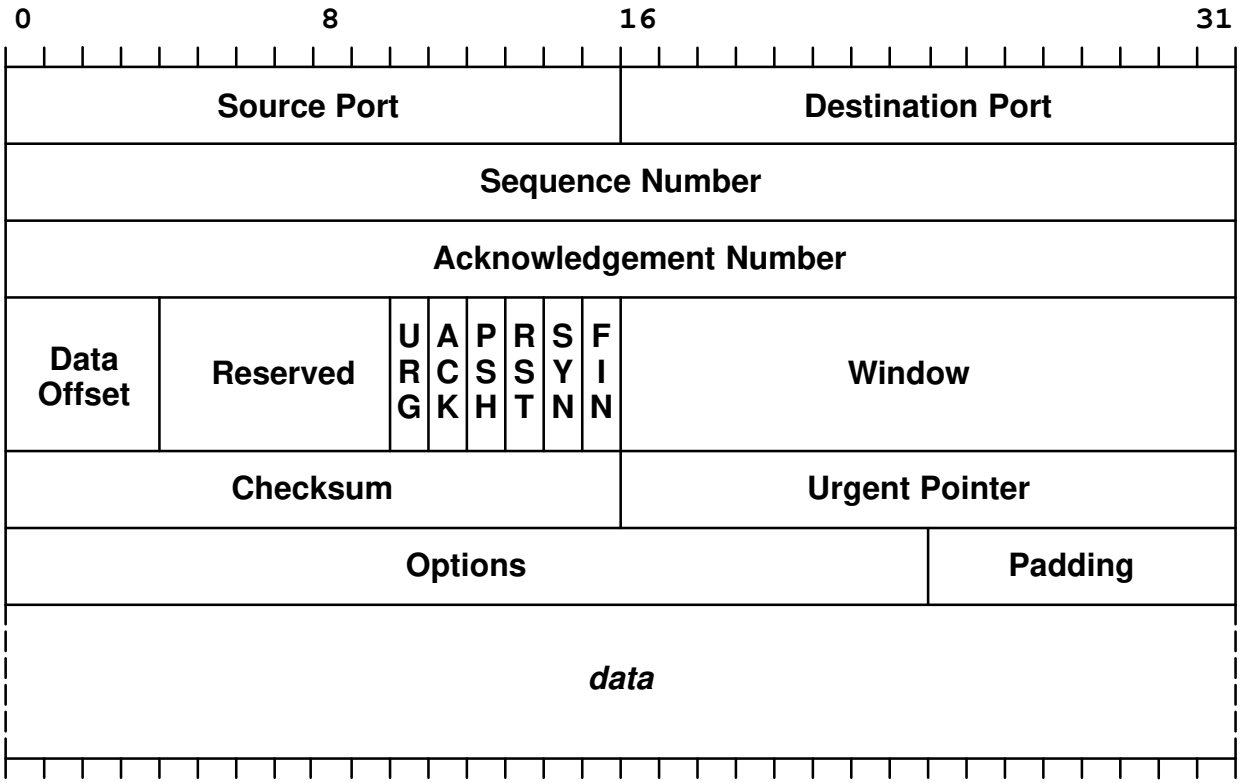
TCP Details (continued)

- Receiver controls flow by telling sender size of currently available buffer
- Call window *advertisement*
- Each segment contains advertisement , including data segments
- Receiver can send additional acknowledgements whenever buffer space becomes available
- Data flow may be shut down in one direction
- Connection established (and broken) with 3–way handshake

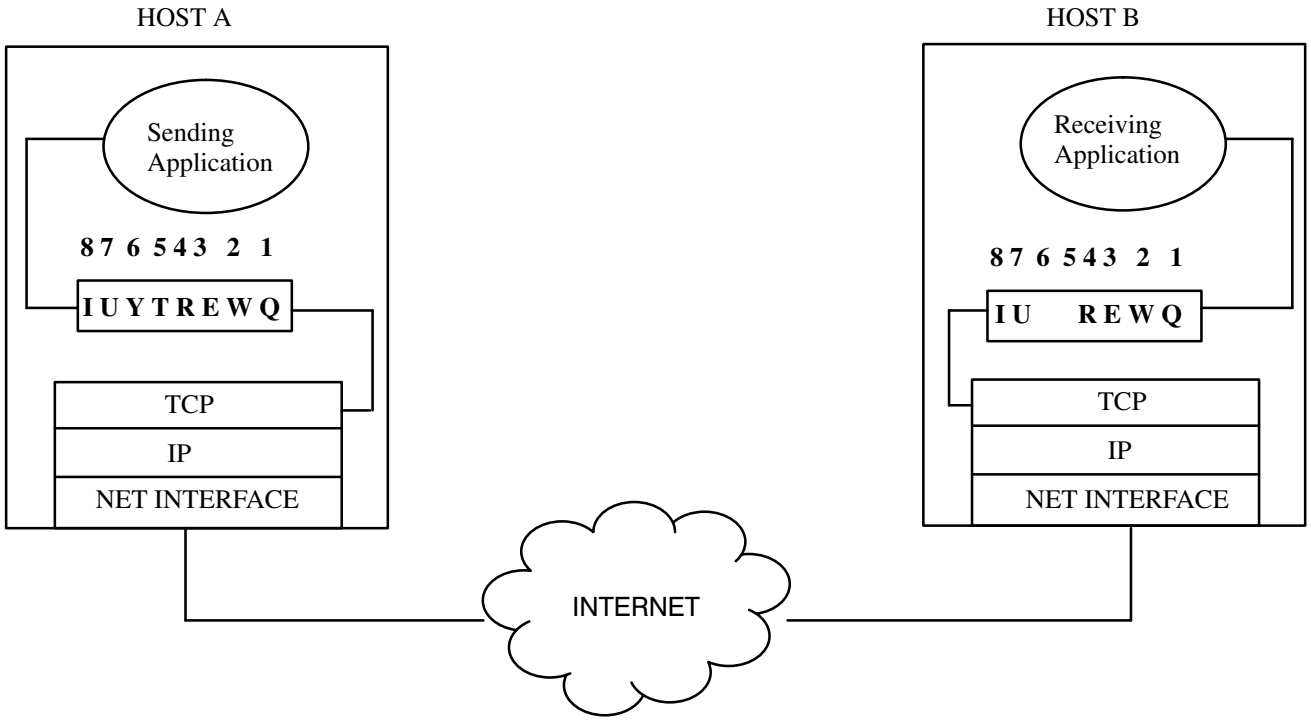
3–Way Handshake



TCP Segment Format

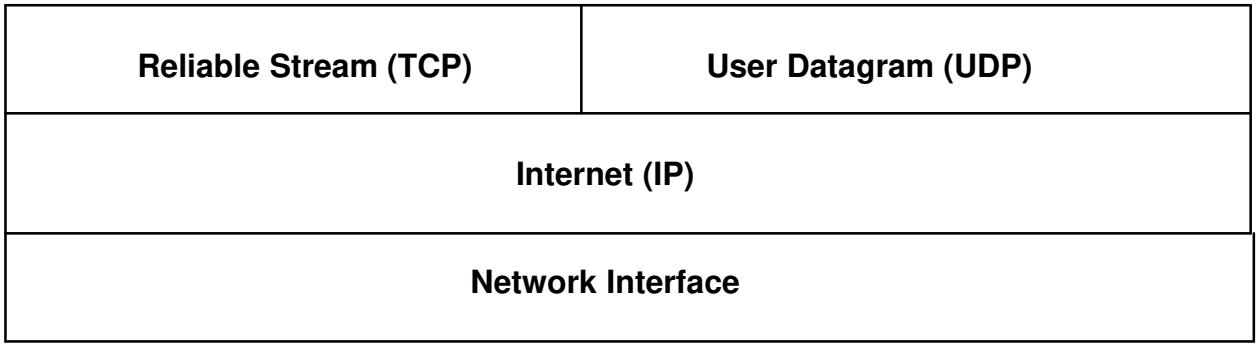


TCP Acknowledgement Example



- Assume octets 5 & 6 lost
- Sender transmits octets 7 & 8
- Receiver acknowledges octets 1–4

Conceptual Layering



Assignment Of Protocol Ports

- Need both globally fixed ports and dynamically allocated ports
- Accommodate with two port types
  - Statically assigned ports
  - Dynamically assigned ports

Statically Assigned Ports

- Called ”well known”
- Used for services like mail
- Fixed by central authority
- Use ”small” values (<256)
- Values less than 1024 reserved for privileged ports

Dynamically Assigned Ports

- Available for user applications
- Operating system chooses when application begins
- Programmer responsible for devising mechanism to inform other programs
- Use ”large” values

Program Interface To Port Assignment

- Literal constants make programs difficult to read
- Most systems provide a database of service names and library routines to map names into protocol port numbers
- Can add local definitions to the database

Example Database For Service Mapping

```
echo            7/tcp
echo            7/udp
ftp             21/tcp
telnet          23/tcp
smtp            25/tcp
time            37/tcp
time            37/udp
nameserver          53/tcp
nameserver          53/udp
foobar          2001/udp
```

Summary

- Protocol layering divides the complex problem of communication into manageable pieces
- Internet Protocol (IP) is machine–to–machine
- IP defines a connectionless, unreliable datagram delivery service
- IP defines the format of IP datagrams, the unit of Internet transfer analogous to a physical network packet

Summary (continued)

- Transmission Control Protocol (TCP) is an end–to–end protocol
- TCP provides reliable, flow–controlled, full–duplex stream delivery service
- User Datagram Protocol (UDP) is an end–to–end datagram protocol
- UDP gives application programs access to connectionless delivery service