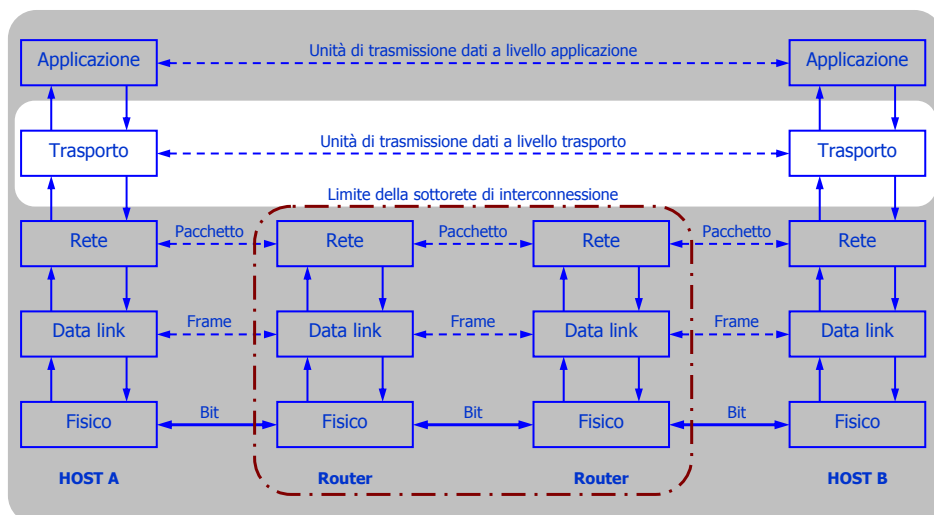


Livello Trasporto



12/03/2005

Reti di Calcolatori

1

The Transport Service

- Services Provided to the Upper Layers
- Transport Service Primitives
- Berkeley Sockets
- An Example of Socket Programming:
 - An Internet File Server

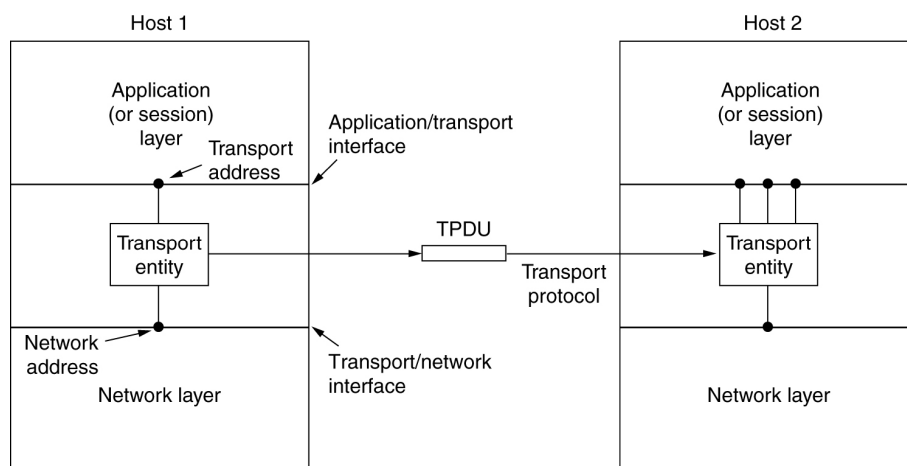
12/03/2005

Reti di Calcolatori

2

Services Provided to the Upper Layers

The network, transport, and application layers.



12/03/2005

Reti di Calcolatori

3

Transport Service Primitives

The primitives for a simple transport service.

Primitive	Packet sent	Meaning
LISTEN	(none)	Block until some process tries to connect
CONNECT	CONNECTION REQ.	Actively attempt to establish a connection
SEND	DATA	Send information
RECEIVE	(none)	Block until a DATA packet arrives
DISCONNECT	DISCONNECTION REQ.	This side wants to release the connection

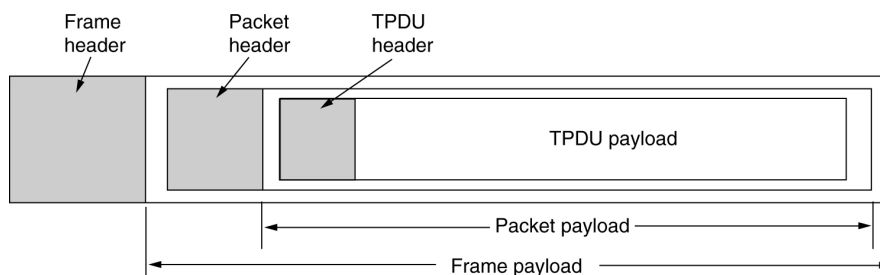
12/03/2005

Reti di Calcolatori

4

Transport Service Primitives (2)

The nesting of TPDU, packets, and frames.



12/03/2005

Reti di Calcolatori

5

Elements of Transport Protocols

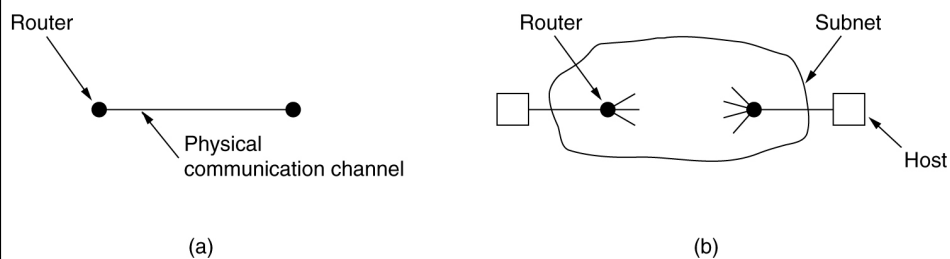
- Addressing
- Connection Establishment
- Connection Release
- Flow Control and Buffering
- Multiplexing
- Crash Recovery

12/03/2005

Reti di Calcolatori

6

Transport Protocol



(a) Environment of the data link layer.

(b) Environment of the transport layer.

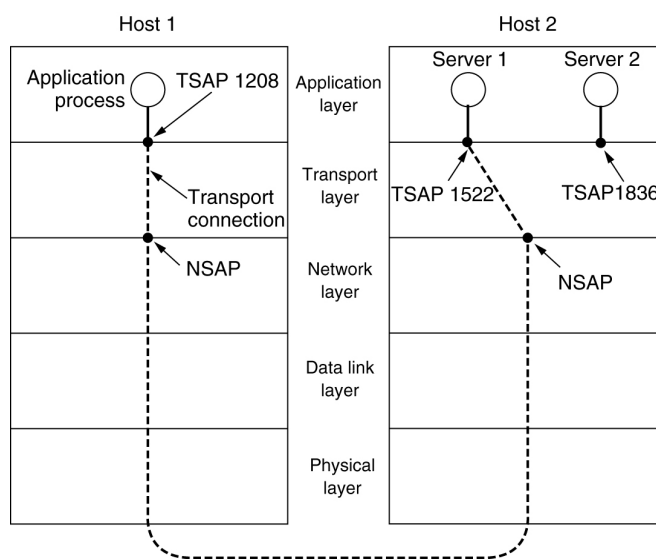
12/03/2005

Reti di Calcolatori

7

Addressing

TSAPs,
NSAPs and
transport
connections.



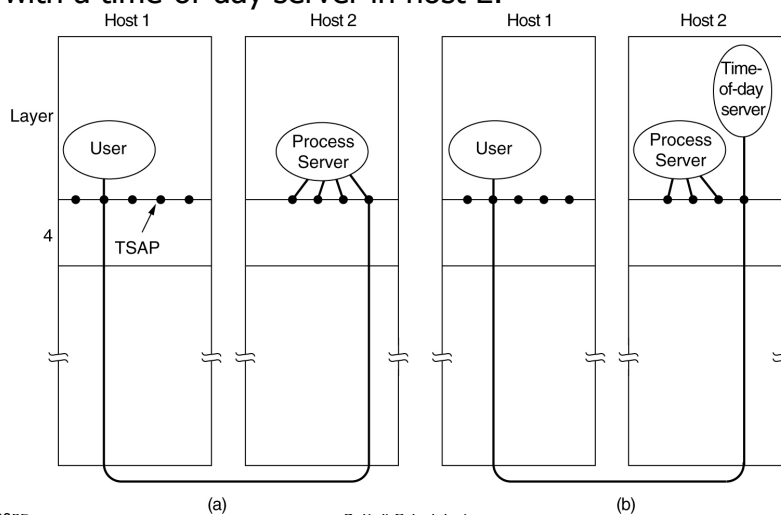
12/03/2005

Reti di Calcolatori

8

Connection Establishment

- How a user process in host 1 establishes a connection with a time-of-day server in host 2.



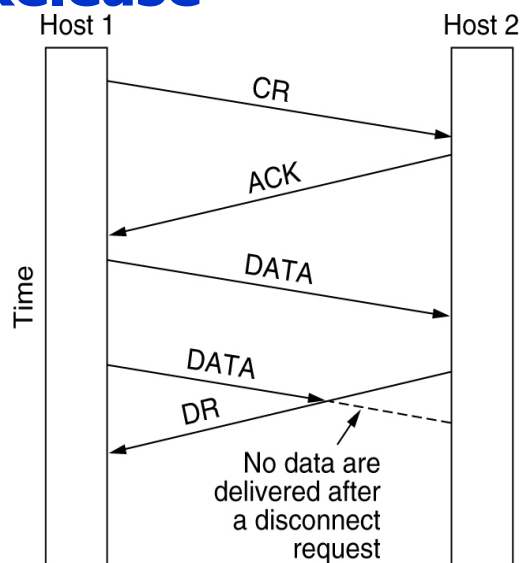
12/03/2005

Reti di Calcolatori

9

Connection Release

Abrupt
disconnection with
loss of data.

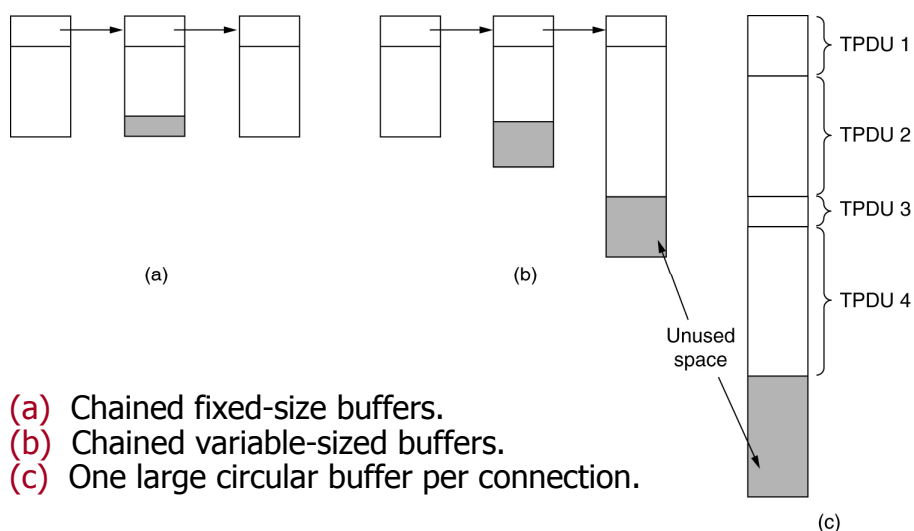


12/03/2005

Reti di Calcolatori

10

Flow Control and Buffering



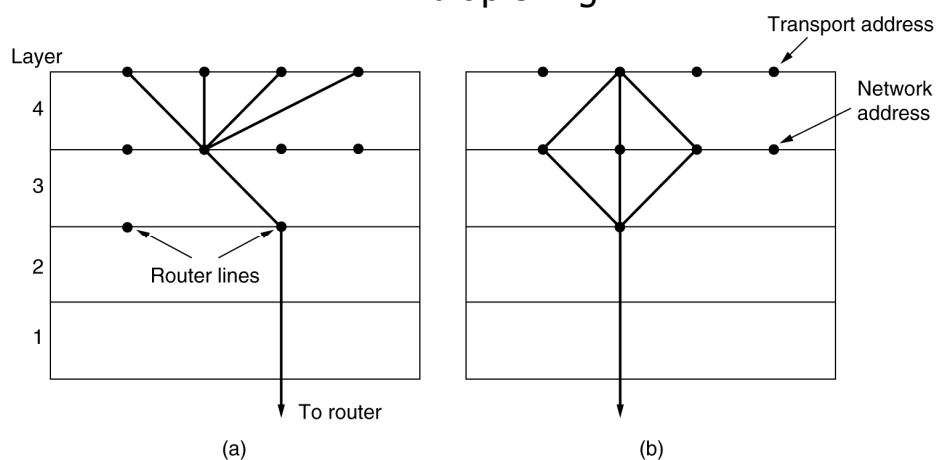
12/03/2005

Reti di Calcolatori

11

Multiplexing

(a) Upward multiplexing. (b) Downward multiplexing.



12/03/2005

Reti di Calcolatori

12

Crash Recovery

Different combinations of client and server strategy.

Strategy used by sending host	Strategy used by receiving host					
	First ACK, then write			First write, then ACK		
	AC(W)	AWC	C(AW)	C(WA)	W AC	WC(A)
Always retransmit	OK	DUP	OK	OK	DUP	DUP
Never retransmit	LOST	OK	LOST	LOST	OK	OK
Retransmit in S0	OK	DUP	LOST	LOST	DUP	OK
Retransmit in S1	LOST	OK	OK	OK	OK	DUP

OK = Protocol functions correctly
 DUP = Protocol generates a duplicate message
 LOST = Protocol loses a message

12/03/2005

Reti di Calcolatori

13

The Internet Transport Protocols: UDP

- Introduction to UDP
- Remote Procedure Call
- The Real-Time Transport Protocol

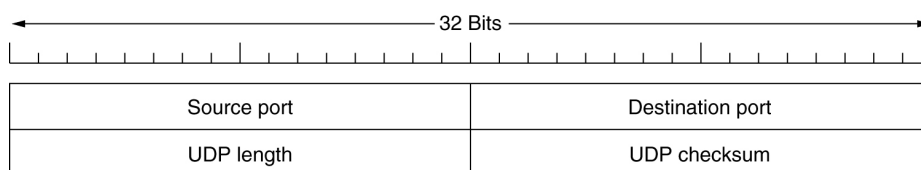
12/03/2005

Reti di Calcolatori

14

Introduction to UDP

The UDP header.



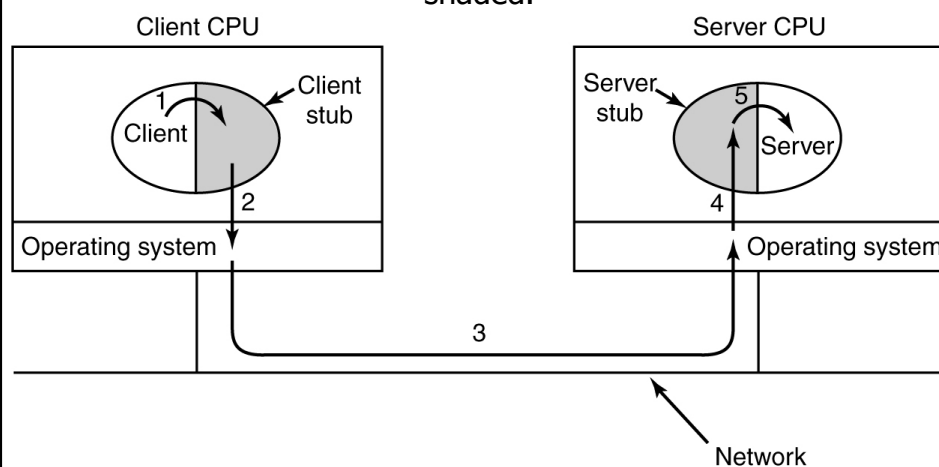
12/03/2005

Reti di Calcolatori

15

Remote Procedure Call

Steps in making a remote procedure call. The stubs are shaded.



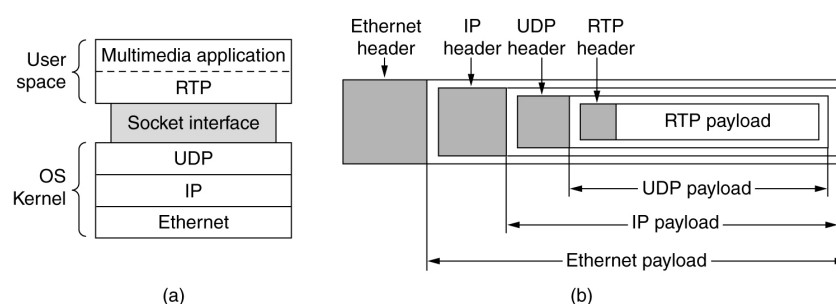
12/03/2005

Reti di Calcolatori

16

The Real-Time Transport Protocol

- (a) The position of RTP in the protocol stack.
 (b) Packet nesting.



12/03/2005

Reti di Calcolatori

17

The Internet Transport Protocols: TCP

- Introduction to TCP
- The TCP Service Model
- The TCP Protocol
- The TCP Segment Header
- TCP Connection Establishment
- TCP Connection Release
- TCP Connection Management Modeling
- TCP Transmission Policy
- TCP Congestion Control
- TCP Timer Management
- Wireless TCP and UDP
- Transactional TCP

12/03/2005

Reti di Calcolatori

18

The Transmission Control Protocol (TCP)

- The TCP layer creates a connection between sender and receiver using port numbers.
- The port number identifies a particular application on a particular device (IP address).
- TCP can multiplex multiple connections (using port numbers) over a single IP line.
- The TCP layer can ensure that the receiver is not overrun with data (end-to-end flow control) using the Window field.
- TCP can perform end-to-end error correction (Checksum).
- TCP allows for the sending of high priority data (Urgent Pointer).

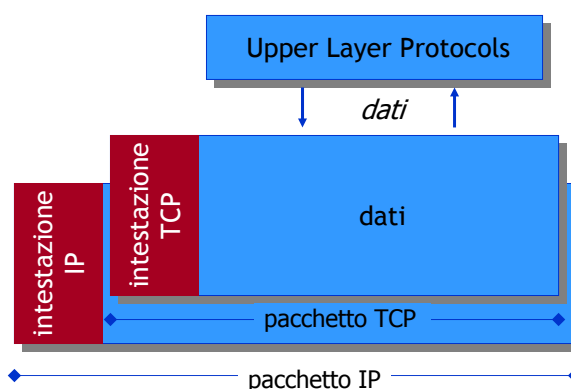
12/03/2005

Reti di Calcolatori

19

Dal livello 3 al livello 4

- Da IP a TCP, mediante incapsulamento dei dati



12/03/2005

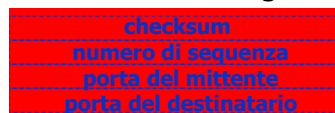
Reti di Calcolatori

20

La logica di TCP

- TCP fornisce, "sopra a IP", il controllo degli errori e la sequenzializzazione nella trasmissione dei dati degli ULP

La struttura (semplificata)
dell'intestazione TCP:



Due nodi comunicano al livello TCP mediante "porte", ognuna identificata da un numero a 16 bit, generalmente scritto in notazione decimale (p.es. 212.239.33.115:80 è la porta 80 del nodo con indirizzo IP 212.239.33.115)

Uno stesso nodo può mantenere comunicazioni diverse attive contemporaneamente, ognuna su una diversa porta

12/03/2005

Reti di Calcolatori

21

I livelli di indirizzamento

livello TCP	porta TCP	80
livello IP	indirizzo IP	212.239.33.115
livello Ethernet	indirizzo MAC	00:10:A4:0C:6E:2A

In trasmissione:

nel pacchetto TCP:
porta TCP del destinatario

incapsulato nel:

nel pacchetto IP:
indirizzo IP del destinatario

risolto mediante ARP nel:

indirizzo MAC del destinatario

In ricezione:

de-incapsulamenti successivi,
per ottenere l'indirizzo IP
e quindi la porta TCP
a cui la trasmissione è rivolta

12/03/2005

Reti di Calcolatori

22

TCP/IP: indirizzamento

➤ Schema di indirizzamento generale su due livelli: **indirizzo IP + porta TCP**

• Indirizzo IP

- indirizzo associato a ogni calcolatore collegato a una sottorete;
- si tratta di un indirizzo **Internet** globale unico, utilizzato da IP per l'instradamento e la consegna dei pacchetti.

• Porta TCP

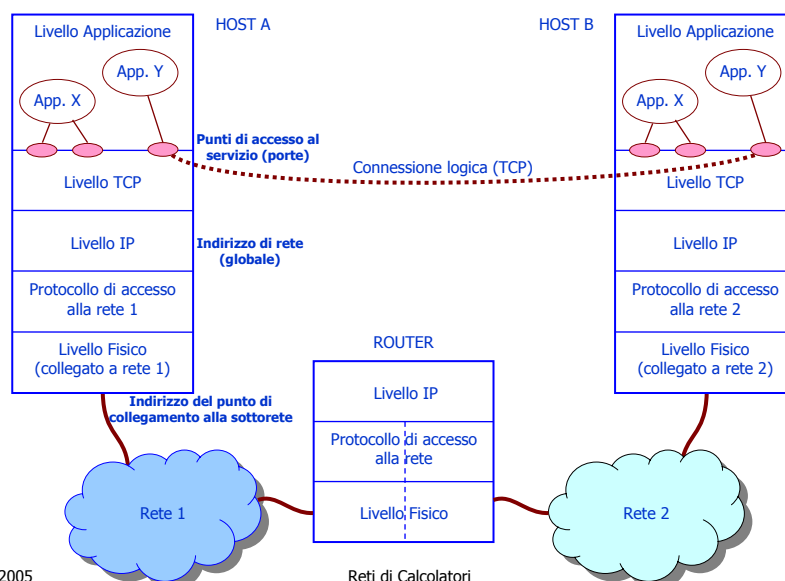
- indirizzo unico all'interno dell'host che individua un processo attivo sull'host;
- utilizzato da TCP per consegnare i dati al processo giusto;
- TCP aggiunge altre informazioni di controllo/servizio:
 - il **numero d'ordine** nella sequenza (riordinare i messaggi dopo il loro arrivo a destinazione);
 - **codici di controllo della correttezza (checksum)**, che permettono al destinatario di verificare l'assenza di errori;
 - ...

12/03/2005

Reti di Calcolatori

23

Indirizzi TCP/IP



12/03/2005

Reti di Calcolatori

24

Affidabilità su TCP/IP

- IP è un protocollo **connectionless** (non orientato alla connessione)
 - frammenta il messaggio in datagrammi;
 - ogni datagramma viene inviato a destinazione lungo percorsi indipendenti;
 - il controllo (*checksum*) consente soltanto la verifica dell'integrità dell'intestazione, ma non dei dati;
 - attenzione:
 - non c'è garanzia che tutti i pezzi arrivino a destinazione né che arrivino "in ordine"
 - la correttezza e l'ordine di ricezione dei dati devono essere assicurati da protocolli di livello più elevato.
- TCP è un protocollo **connection oriented** (orientato alla connessione)
 - garantisce la consegna di un messaggio completo di tutte le sue parti e ordinato correttamente,
 - il controllo consente la valutazione della correttezza sia dell'intestazione TCP che dei dati.
- La combinazione delle due modalità permette di ottenere sia una buona efficienza di trasmissione che una elevata affidabilità:
 - OK per applicazioni **client-server**;
 - KO laddove l'affidabilità non è un requisito essenziale. In questi casi TCP può essere sostituito con altri protocolli (e.g. UDP - User Datagram Protocol).

12/03/2005

Reti di Calcolatori

25

Some assigned TCP-ports

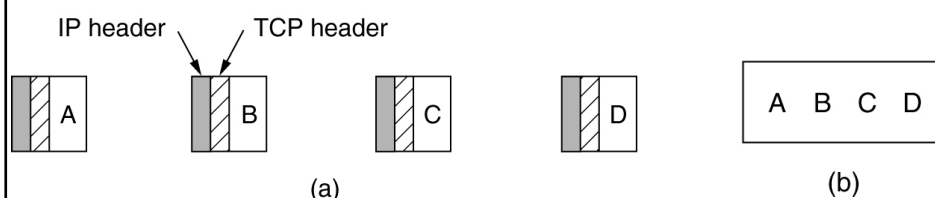
Port	Protocol	Use
21	FTP	File transfer
23	Telnet	Remote login
25	SMTP	e-mail
69	TFTP	Trivial File Transfer Protocol
79	Finger	Lookup info about a user
80	HTTP	World Wide Web
110	POP-3	Remote e-mail
119	NNTP	USENET news

12/03/2005

Reti di Calcolatori

26

The TCP Service Model (2)



- (a) Four 512-byte segments sent as separate IP datagrams.
- (b) The 2048 bytes of data delivered to the application in a single READ CALL.

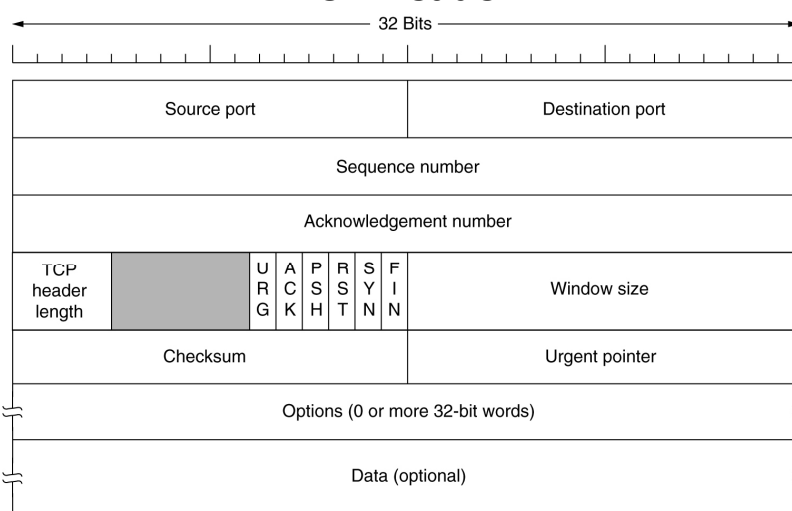
12/03/2005

Reti di Calcolatori

27

The TCP Segment Header

TCP Header.



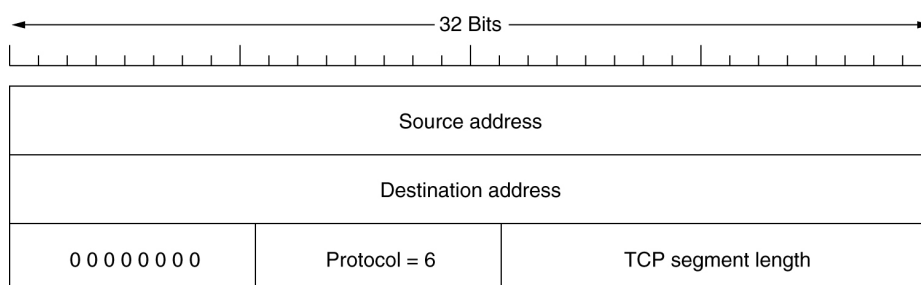
12/03/2005

Reti di Calcolatori

28

The TCP Segment Header (2)

The pseudoheader included in the TCP checksum.



12/03/2005

Reti di Calcolatori

29

Network Performance Measurement

- The basic loop for improving network performance.
 - Measure relevant network parameters, performance.
 - Try to understand what is going on.
 - Change one parameter.

12/03/2005

Reti di Calcolatori

30

System Design for Better Performance

Rules:

- CPU speed is more important than network speed.
- Reduce packet count to reduce software overhead.
- Minimize context switches.
- Minimize copying.
- You can buy more bandwidth but not lower delay.
- Avoiding congestion is better than recovering from it.
- Avoid timeouts.

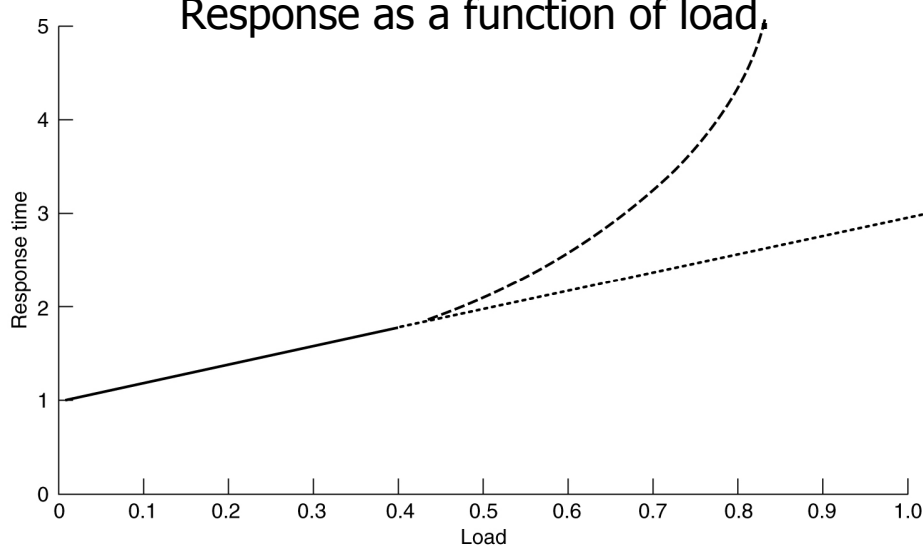
12/03/2005

Reti di Calcolatori

31

System Design for Better Performance (2)

Response as a function of load



12/03/2005

Reti di Calcolatori

32