



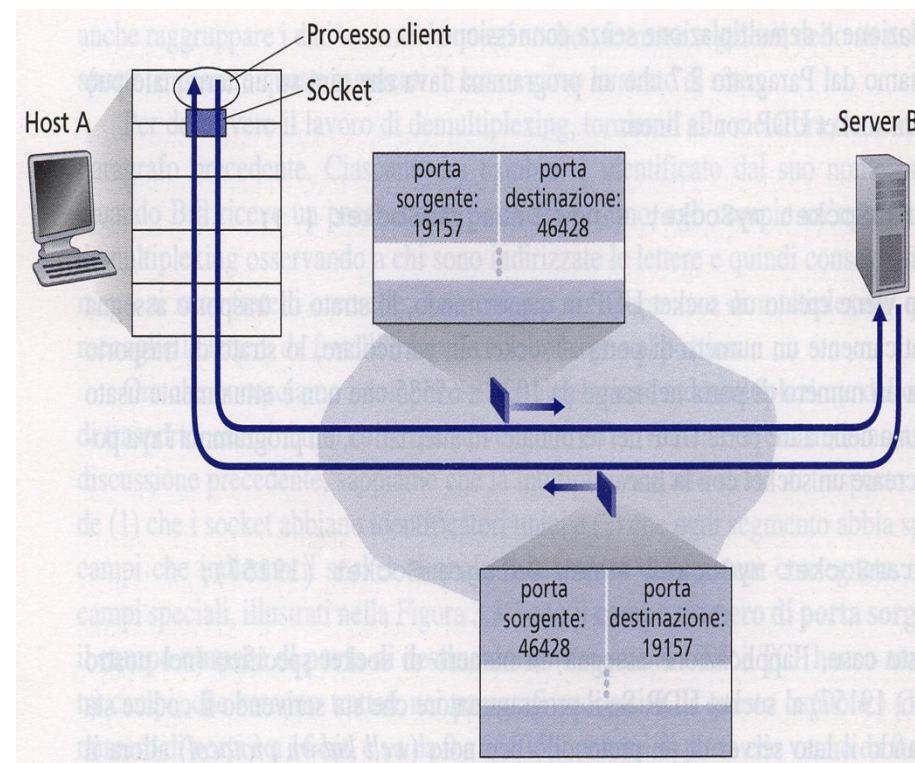
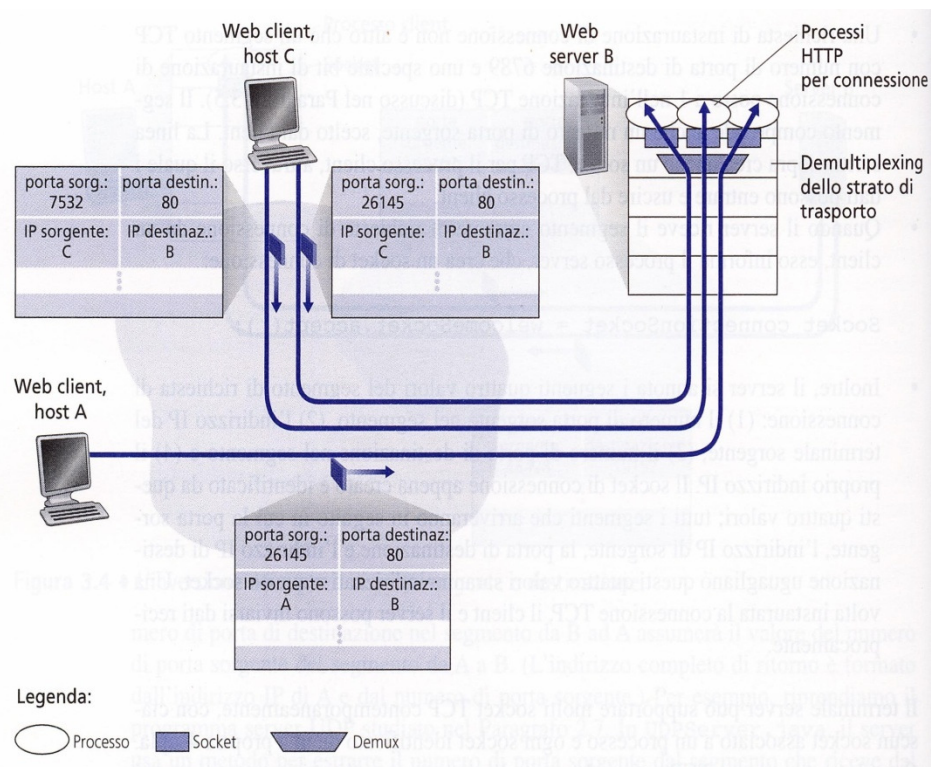
Corso di
SISTEMI TELEMATICI
a.a. 2012-2013

Lo strato di Trasporto



Università della Calabria D.E.I.S.

Porte TCP e UDP











Esempi di utilizzo delle porte

Well Known Port



Sono associate agli applicativi principali

Servizio	Porta	TCP	UDP
FTP	21		
Telnet	23		
SMTP	25		
TFTP	69		
DNS	53		
HTTP	80		
SNMP	161		



Il protocollo UDP

(User Datagram Protocol)

RFC 768

Protocollo UDP: funzioni



- E' il protocollo di trasporto più semplice in grado di usare il servizio di comunicazione e le funzionalità di IP facendo colloquiare processi remoti
- UDP aggiunge due funzionalità a quelle di IP:
 - ✗ l'indirizzamento delle applicazioni, cioè il de/multiplexing delle informazioni tra le varie applicazioni tramite il concetto di porta
 - ✗ un blando controllo d'errore sull'header dei messaggi, cioè una checksum (opzionale) per verificare l'integrità dei dati

Protocollo UDP: servizio



- UDP è un protocollo che fornisce un servizio di tipo datagram, che non garantisce la consegna e che non esercita nessun controllo sul flusso e riordinamento delle unità informative emesse dall'applicazione:
 - ✗ connectionless (pacchetti fuori sequenza)
 - ✗ non affidabile (pacchetti persi)
 - ✗ senza controllo di flusso (saturazione del ricevitore)
- Non prevede meccanismi di recupero da errore, es. ritrasmissioni in caso di errori/perdite
 - ✗ eventuali meccanismi di ritrasmissione (se necessari) vengono gestiti direttamente dall'applicazione

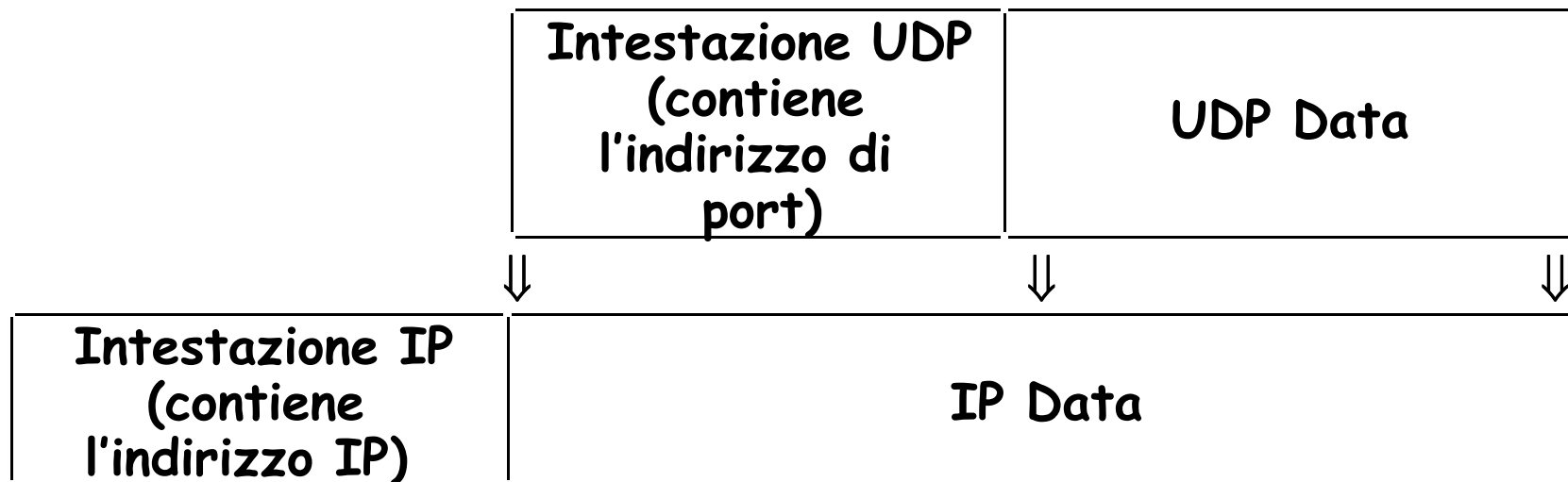


Protocollo UDP: applicazioni

- UDP è usato da quegli applicativi che non necessitano di un trasferimento affidabile e per i quali l'overhead dovuto alla fase di apertura di un servizio di trasporto orientato alla connessione non sarebbe giustificato. Tra questi: DNS, NFS, SNMP, RIP, ecc.
- Inoltre, UDP è usato dai servizi che non possono tollerare il controllo di flusso del TCP. Tra questi, i servizi di trasporto di flussi *stream* come voce o video. In questo caso, di solito, alle funzionalità di UDP vengono aggiunte quelle del protocollo RTP (Real Time Protocol), che ha come compito principale quello di aggiungere all'header UDP le funzionalità di numerazione dei pacchetti e di time-stamp

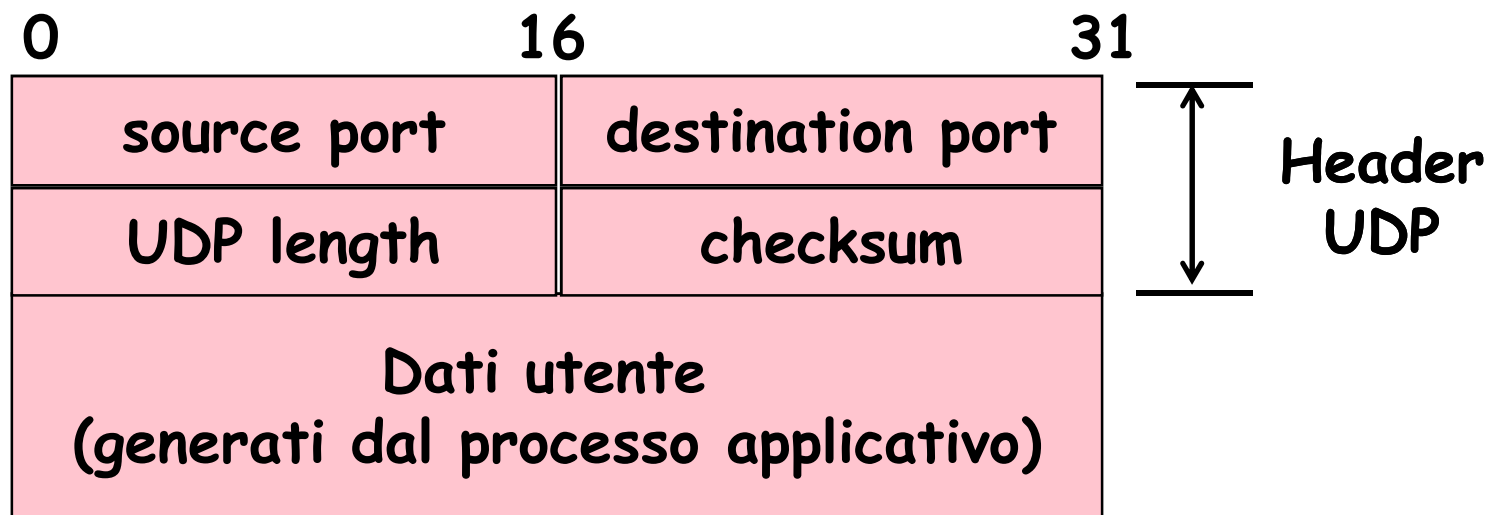


UDP accetta dal livello superiore dati senza vincoli sulla loro lunghezza, eventualmente li frammenta e li invia in datagrammi IP distinti





- L'unità dati UDP (datagramma utente) ha lunghezza variabile, viene imbustata in IP ed indirizzata con il campo Protocol pari a 17
- L'intestazione di UDP è lunga 8 byte, contro i 20 byte dell'intestazione TCP

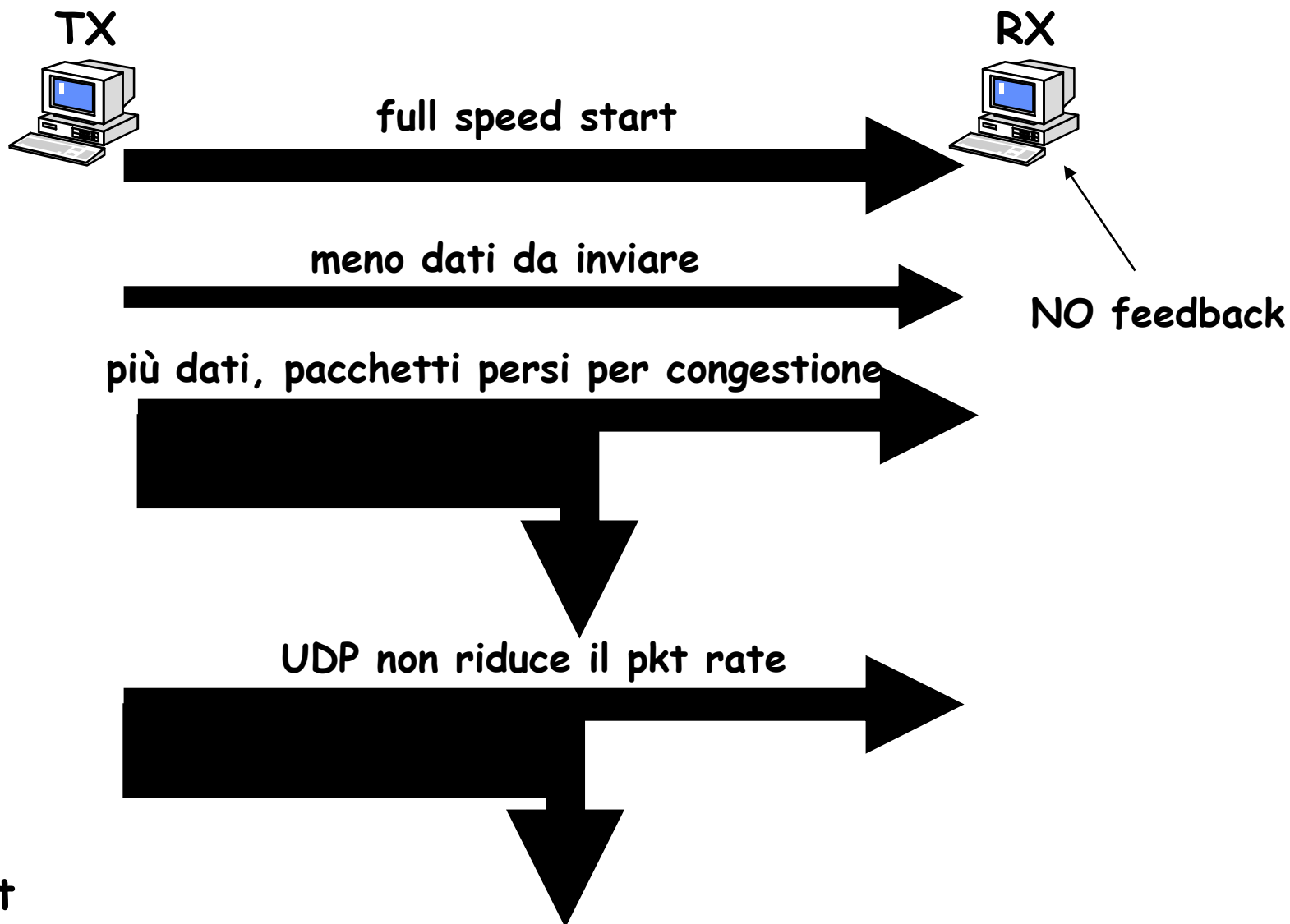




Pacchetto UDP

- **Port number (16 bit)** indirizzi delle porte sorgente e destinazione
- **Length (16 bit)** è la lunghezza in byte del datagramma UDP (header + dati); il minimo valore è di 8 byte, quando la parte dati è vuota
 - ✗ l'informazione è ridondante, visto che l'header UDP ha lunghezza fissa di 8 byte, la lunghezza della parte dati potrebbe essere ricavata sottraendo 8 byte al contenuto del campo length dell'header IP
- **Checksum (16 bit)**, campo opzionale per il controllo di errore; quando non si usa (in reti altamente affidabili) si riduce il carico di processamento di un datagramma; però dato che IP non fa controllo di errore, la checksum è l'unico strumento per verificare che i dati siano giunti a destinazione correttamente

UDP: congestione





Il protocollo TCP (Transmission Control Protocol)

Definizione: RFC 793

Identificazione di bug: RFC 1122

Estensioni : RFC 1323



Transmission Control Protocol (TCP)

- TCP è un protocollo “con connessione” che fornisce un servizio affidabile “end-to-end” tra coppie di processi su host remoti
- TCP effettua funzioni di:
 - × indirizzamento di uno specifico utente all’interno di un host (multiplazione e de-multiplazione delle informazioni)
 - × trasferimento di un flusso informativo continuo e bi-direzionale (byte stream), ma non strutturato, di dati tra host remoti (funzione di segmentation & reassembly)
 - × gestione delle connessioni
 - × controllo e recupero di errore
 - × controllo di flusso
 - × controllo di congestione
 - × riordinamento delle unità informative

Indirizzamento TCP



- **Multiplexing:**

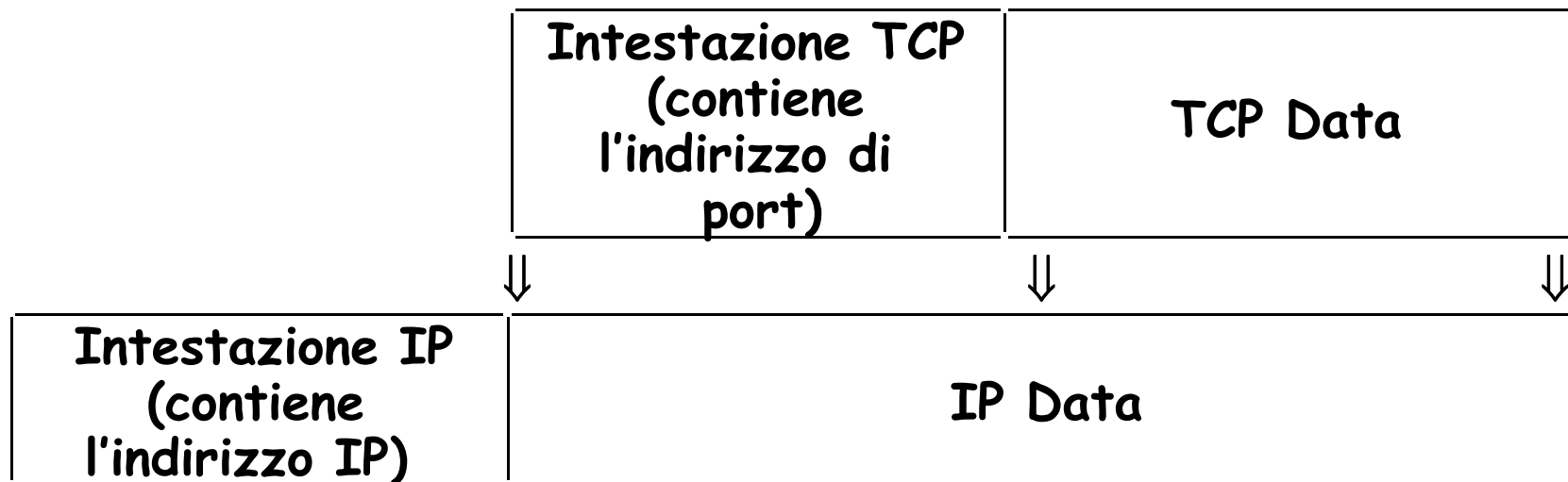
per permettere ai processi in un Host di usare contemporaneamente le facilità di comunicazione di TCP, il TCP fornisce un set di indirizzi o porte in ogni host

- L'indirizzo completo TCP/IP è costituito dall'insieme di indirizzo IP e numero di porta e identifica univocamente un processo in esecuzione su un host
- Tale indirizzo viene spesso indicato con il nome di “socket”;

Indirizzamento TCP



La componente "port" è contenuta nell'intestazione dell'unità dati di TCP, mentre la componente IP_Address è contenuta nell'intestazione dell'unità dati di IP



Indirizzamento TCP



- Questo significa che tutte le connessioni in atto tra due specifici host usano gli stessi indirizzi IP di sorgente e di destinazione. Saranno perciò distinte solo a livello TCP
- Ne segue che queste connessioni possono essere viste come multiplate su un unico indirizzo IP, ovvero su un unico “canale” IP di comunicazione (non su una connessione IP, dato che IP è connectionless)

Indirizzamento TCP



- Una connessione TCP è identificata dalla coppia di socket (sorgente e destinazione) associata ai due processi (end-point) che hanno stabilito la connessione
- Un numero di porta può essere usato per più connessioni, ma l'insieme dei socket di sorgente e destinazione identifica univocamente una connessione
 - ad es., un server web può avere più connessioni contemporaneamente attive sulla porta 80 e le distingue sulla base dell'indirizzo IP e del numero di porta dei client

Indirizzamento TCP



- Un end-point può essere impegnato allo stesso tempo in più connessioni TCP

(21; 151.100.37.13) (21; 128.10.2.3)

(21; 18.26.0.36)

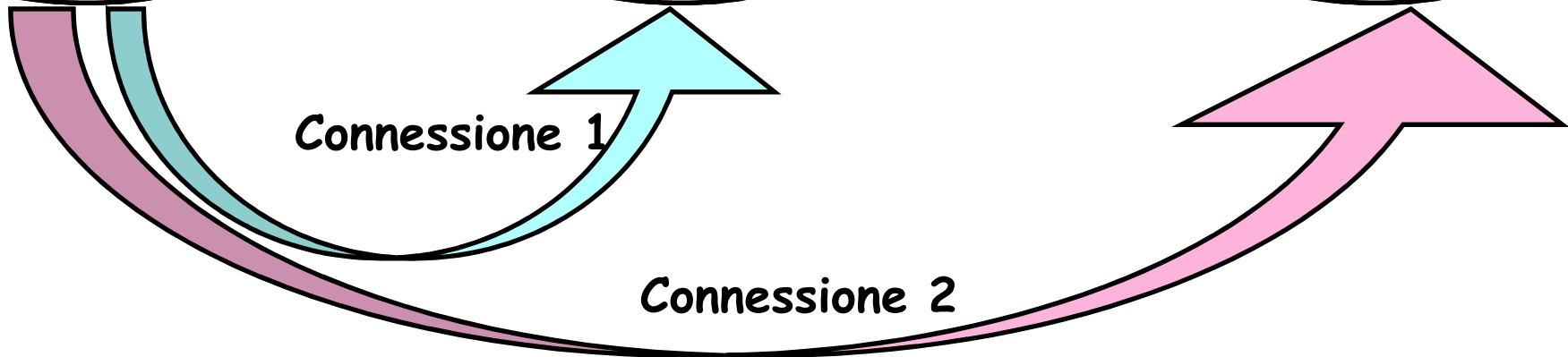
End-point
A

End-point
B

End-point
C

Connessione 1

Connessione 2



Indirizzamento TCP



L'assegnazione del numero di porta può essere:

- **Statico**

- l'identificativo è staticamente associato all'applicazione
- sono utilizzati identificativi inferiori a 255

Numero	Applicazione	Numero	Applicazione
7	Echo	37	Time
21	FTP	53	Domain Name Server
23	TELNET	103	X400 Mail Service
25	SMTP	119	NNTP (USENET New Transfer Prot.)

- **Dinamico:** l'identificativo è assegnato direttamente dal sistema operativo al momento dell'apertura della connessione



Trasferimento dati TCP

- TCP accetta dal livello applicativo un flusso continuo di dati non strutturati (byte stream), li frammenta e li invia in unità dati distinte, detti segmenti
- La lunghezza massima dei segmenti viene negoziata durante la fase di apertura della connessione (MSS – Maximum Segment Size) e comunque è inferiore a 64 kbyte, dato che il segmento deve poter utilizzare il payload di un unico pacchetto IP, e dipende dall'MTU



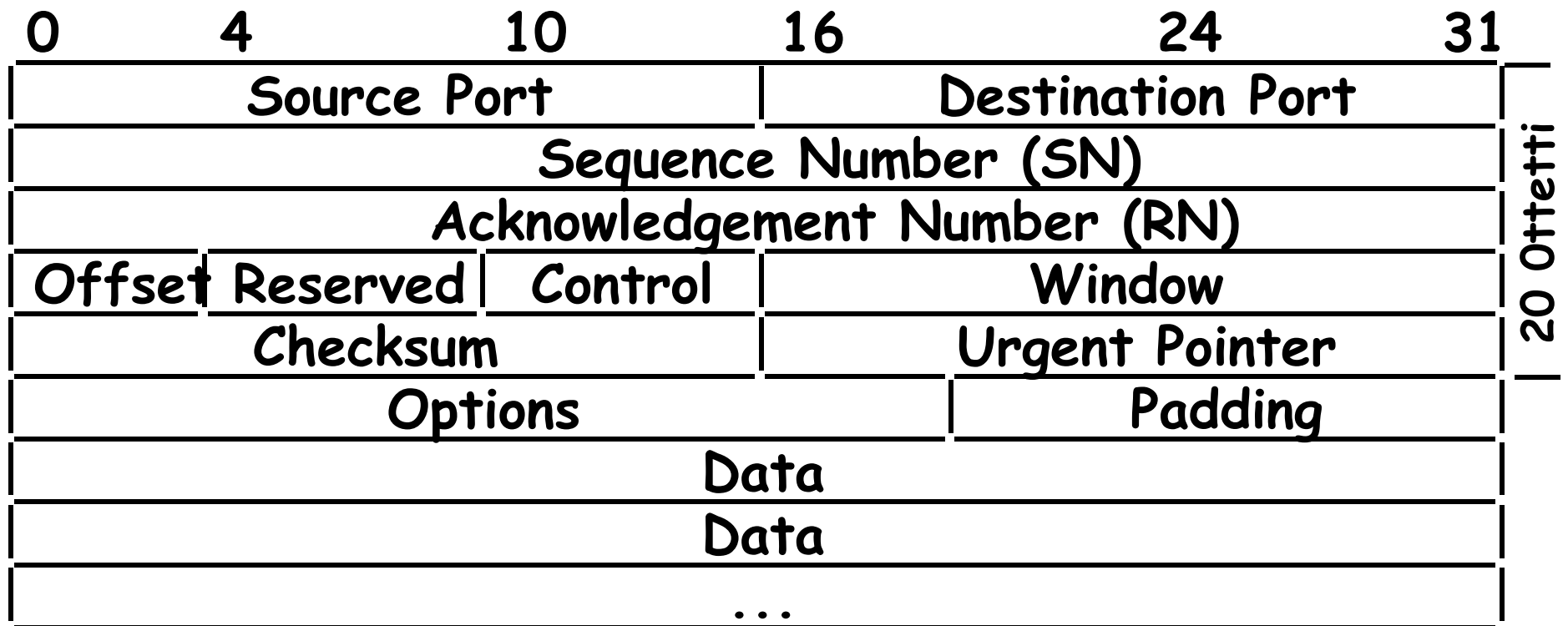
Trasferimento dati TCP

- Le operazioni di segmentazione e riassetramento possono avvenire in parallelo, cioè il TCP è un protocollo full-duplex punto-punto
- Il trasferimento dati TCP avviene secondo un meccanismo di sliding window: all'invio di un segmento il sender fa partire un timer, quando il segmento arriva a destinazione il receiver invia un segmento con il riscontro del segmento ricevuto; se il timer del sender scade prima della ricezione del riscontro il sender ritrasmette il segmento
- Il TCP utilizza un unico formato di trame sia per la trasmissione di informazione d'utente che per l'informazione di servizio (apertura e chiusura della connessione, messaggi per il controllo d'errore e quello di flusso)



Formato dell'unità dati TCP

- L'header è lungo 20 byte se il campo opzioni non viene utilizzato. Il campo dati può essere vuoto (trame di acknowledgment, connessione, ecc.).





Formato dell'unità dati

- **Source Port (16 bit):** definisce l'indirizzo logico del processo sorgente dei dati
- **Destination Port (16 bit):** definisce l'indirizzo logico del processo destinatario dei dati
- **Sequence Number (32 bit):** numero di sequenza in trasmissione (SN); contiene il numero di sequenza del primo byte di dati contenuti nel segmento a partire dall'inizio della sessione (se $SN=m$ ed il segmento contiene n byte il prossimo SN sarà pari a $m+n$)
- **Acknowledgement Number (32 bit):** numero di sequenza in ricezione (RN); nei segmenti in cui il bit ACK è 1, contiene il numero di sequenza del prossimo byte che il ricevitore si aspetta di ricevere. Il meccanismo di riscontro usato è cumulativo, così l'ack di un SN X indica che sono stati ricevuti tutti i byte fino a X escluso X . In caso di connessioni interattive bidirezionali gli ack sono inviati in piggybacking (nei segmenti di risposta contenenti dati di utente). I numeri SN e RN vengono utilizzati per il controllo d'errore e di flusso



Formato dell'unità dati: SN e Ack

- I numeri di sequenza e gli ack rendono affidabile la trasmissione TCP. Ad ogni byte di dati si assegna un numero di sequenza. In ogni segmento TCP si inserisce il numero di sequenza del primo byte di dati contenuto nel segmento (SN)
- I segmenti in direzione opposta portano anche un numero di acknowledgment che è il numero di sequenza del successivo byte di dati da trasmettere atteso dal ricevitore
- Quando il TCP trasmette un segmento dati, ne conserva una copia in una coda di ritrasmissione e fa partire un timer; quando riceve l'ack per quei dati, allora cancella il segmento dalla coda. Se l'ack non è ricevuto prima della scadenza del time-out, il segmento viene ritrasmesso
- Così il TCP mantiene la corretta sequenza dei segmenti in ricezione; cioè prima di inviare una nuova sequenza di byte aspetta che la sequenza precedente venga riscontrata



Formato dell'unità dati

- **Offset (4 bit):** contiene il numero di parole di 32 bit contenute nell'intestazione TCP (da Source port a Padding). L'intestazione TCP non supera i 64 byte ed è un multiplo di 32
- **Reserved (6 bit):** riservato per usi futuri, contiene zeri
- **Control bit (6 bit):** i bit di controllo sono 6:
 - ✗ **URG:** vale 1 quando il campo Urgent Pointer contiene un valore significativo; è usato per indicare dati urgenti che vengono trasmessi al di fuori del controllo di flusso con un meccanismo di segnalazione end-to-end tra processi remoti
 - ✗ **ACK:** vale 1 quando il campo Acknowledgement Number contiene un valore significativo
 - ✗ **PSH:** vale 1 quando l'applicazione esige che i dati forniti vengano trasmessi e consegnati all'applicazione ricevente prescindendo dal riempimento dei buffer allocati fra applicazione e TCP e viceversa (di solito il riempimento dei buffer scandisce la trasmissione e la consegna dei dati)



Formato dell'unità dati

- **RST**: vale 1 quando un malfunzionamento impone il reset della connessione
- **SYN**: indica l'inizio della connessione; vale 1 solo nel primo segmento inviato durante la fase di sincronizzazione fra le entità TCP (3-way handshaking)
- **FIN**: indica la fine della connessione; vale 1 quando la sorgente ha esaurito i dati da trasmettere
- ✖ **Window (16 bit)**: larghezza della finestra per il controllo di flusso; il TCP ricevente riporta al TCP trasmittente il valore di una "window", che contiene il numero di byte che, a cominciare dal valore del campo Acknowledgement Number, il TCP ricevente è disposto a ricevere. Il controllo di flusso è orientato al byte
- ✖ **Checksum (16 bit)**: contiene la sequenza che permette al TCP ricevente di verificare la correttezza del segmento;