Vincenzo Eramo

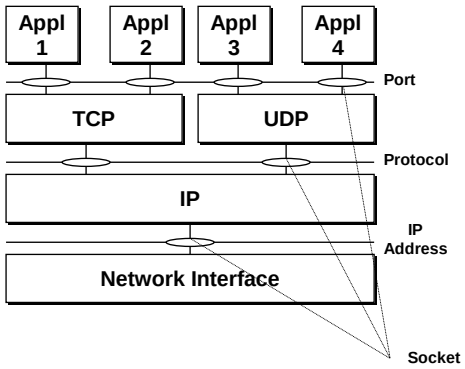
Protocolli di Trasporto in reti IP

Protocolli di Trasporto

- **Lo strato di trasporto fornisce un servizio di trasferimento allo strato applicativo conforme ai requisiti di qualità richiesti dall'applicazione**
- **User Datagram Protocol (UDP)**
  - **è utilizzato quando l'applicazione non richiede funzioni di controllo di flusso e controllo d'errore**
- **Transport Control Protocol (TCP)**
  - **è utilizzato per applicazioni che generano flussi informativi di una certa complessità che richiedono funzioni di controllo d'errore e di flusso**

INFOCOM Dept.

Indirizzamento TCP/UDP



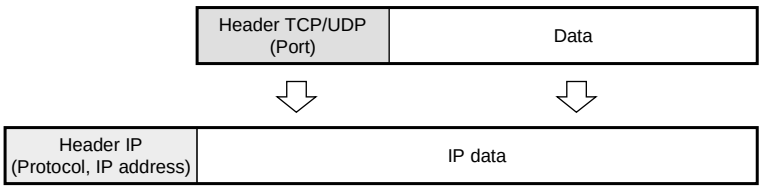
- **Distingue tra i diversi programmi applicativi (processi) che sono utenti dello stesso servizio di trasporto**
- **Port**
  - **identifica un utente dello strato di trasporto**
  - **è rappresentato da un intero (16 bit)**
- **Socket**
  - **identifica l'interfaccia tra l'applicazione ed i protocolli di comunicazione**
  - **è rappresentata dalla tripletta**

  (port; protocol; IP_Address)

INFOCOM Dept.

Indirizzamento TCP/UDP

- **La componente "Port" è contenuta nell'intestazione dell'unità dati di TCP/UDP**
- **Le componenti "Protocol" e "IP_Address" sono contenute nell'intestazione dell'unità dati di IP**



Reti di Telecomunicazioni - Vincenzo Eramo - A.A. 2004/2005

Indirizzamento TCP/UDP

- **Il numero di porta può essere**
  - **statico (Well Known port)**
    - sono identificativi staticamente associati ad applicazioni largamente utilizzate
    - sono utilizzati identificativi inferiori a 256

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

  - **dinamico (Ephemeral)**
    - sono identificativi assegnati direttamente dal sistema operativo al momento dell'apertura della connessione
    - si utilizzano valori maggiori di 1023

INFOCOM Dept.

Vincenzo Eramo

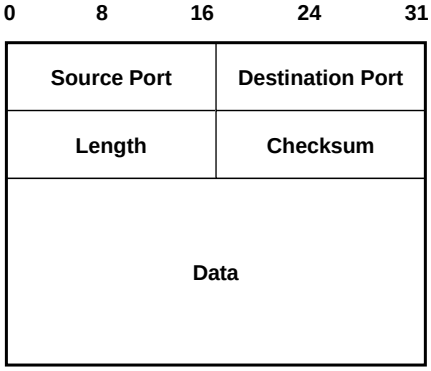
User Datagram Protocol (UDP)

Reti di Telecomunicazioni - Vincenzo Eramo - A.A. 2004/2005

User Datagram Protocol (UDP)

- **E' un protocollo senza connessione**
- **Non supporta meccanismi di riscontro e di controllo d'errore**
- **E' utilizzato per il supporto di transazioni semplici tra applicativi**
  - **interrogazioni di database**
  - **risoluzione di indirizzi**
  - **messaggi di management**

UDP



- **Source Port (16 bit) e Destination Port (16 bit)**
  - **identificano i processi sorgente e destinazione dei dati**
- **Datagram Length (16 bit)**
  - **è la lunghezza totale (espressa in byte) del datagramma, compreso l'header UDP**
- **Checksum (16 bit)**
  - **protegge il datagramma UDP e i campi indirizzo, protocol e datagram length dell'header IP**

Vincenzo Eramo

Transmission Control Protocol (TCP)

Transmission Control Protocol (TCP)

- **Trasferisce un flusso informativo bi-direzionale non strutturato tra due host ed effettua operazioni di multiplazione e de-multiplazione**
- **E' un protocollo con connessione**
- **Funzioni eseguite**
  - **controllo e recupero di errore**
  - **controllo di flusso**
  - **ri-ordinamento delle unità informative**
  - **indirizzamento di una specifica applicazione all'interno di un host**

INFOCOM Dept.

Unità dati TCP

- **Il TCP interpreta il flusso dati come sequenza di ottetti**
- **La sequenza di ottetti è suddivisa in segmenti**

0	4	8	16	24	31
Source Port			Destination Port		
Sequence Number					
Acknowledegment Number					
HLEN	Reserved	Code bits	Window		
Checksum			Urgent Pointer		
Options (if any)			Padding		
Data					
Data					

INFOCOM Dept.

Unità dati TCP

- **Source Port (16 bit) e Destination Port (16 bit)**
  - **identificano i processi sorgente e destinazione dei dati**
- **Sequence Number (32 bit)**
  - **numero di sequenza in trasmissione**
  - **contiene il numero di sequenza del primo byte di dati contenuti nel segmento a partire dall'inizio della sessione TCP**
- **Acknowledgement Number (32 bit)**
  - **numero di sequenza in ricezione**
  - **se ACK=1, contiene il numero di sequenza del prossimo byte che il trasmettitore del segmento si aspetta di ricevere**
  - **è possibile la modalità piggybacking di riscontro**

Reti di Telecomunicazioni - Vincenzo Eramo - A.A. 2004/2005

Unità dati TCP

- **HLEN (4 bit)**
  - **contiene il numero di parole di 32 bit contenute nell'intestazione TCP**
  - **l'intestazione TCP non supera i 60 byte ed è sempre un multiplo di 32**
- **Reserved (6 bit)**
  - **riservato per usi futuri, per ora contiene degli zeri**
- **Window (16 bit)**
  - **larghezza della finestra in byte (controllo di flusso è orientato al byte)**
  - **è il numero di byte che, ad iniziare dal valore del campo Ack Number, il trasmettitore del segmento è in grado di ricevere**
- **Checksum (16 bit)**
  - **protegge l'intero segmento più alcuni campi dell'header IP (es. indirizzi)**

INFOCOM Dept.

Unità dati TCP

- **Control bit (6 bit)**
  - **URG**
    - è uguale a 1 quando il campo urgent pointer contiene un valore significativo
  - **ACK**
    - è uguale a 1 quando il campo Ack Number contiene un valore significativo
  - **PSH**
    - è uguale a 1 se i dati devono essere consegnati all'applicazione ricevente prescindendo dal riempimento dei buffer di ricezione
  - **RST**
    - è uguale a 1 in caso di richiesta di reset della connessione
  - **SYN**
    - è uguale a 1 solo nel primo segmento inviato durante la fase di sincronizzazione fra le entità TCP
  - **FIN**
    - è uguale a 1 quando la sorgente ha esaurito i dati da trasmettere

Reti di Telecomunicazioni - Vincenzo Eramo - A.A. 2004/2005

INFOCOM Dept.

Unità dati TCP

- **Urgent Pointer (16 bit)**
  - **contiene il numero di sequenza dell'ultimo byte dei dati che devono essere consegnati urgentemente al processo ricevente**
  - **tipicamente sono messaggi di controllo (out-of-band traffic)**
- **Options (di lunghezza variabile)**
  - **sono presenti solo raramente**
  - **Esempi:**
    - Maximum Segment Size (MSS)
- **Padding (di lunghezza variabile)**
  - **impone che l'intestazione abbia una lunghezza multipla di 32 bit**

INFOCOM Dept.

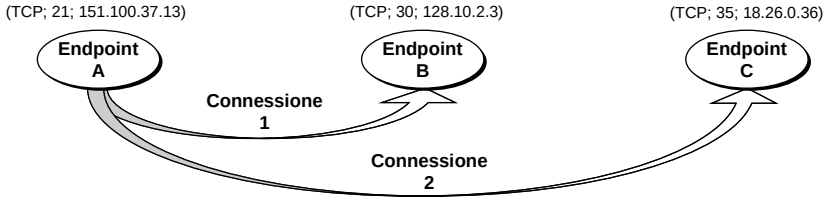
La connessione TCP

- **Il protocollo TCP è un protocollo di tipo orientato alla connessione**
- **Le due entità TCP remote si sincronizzano scambiandosi il proprio numero di sequenza iniziale, che rappresenta il numero a partire dal quale tutti i byte trasmessi saranno numerati in sequenza**

INFOCOM Dept.

La connessione TCP

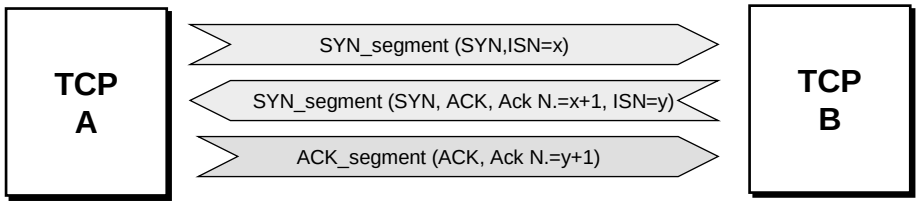
- **Una connessione TCP è identificata dalla coppia di socket associati agli endpoint tra cui vengono scambiate informazioni**
- **Un endpoint può essere impegnato allo stesso tempo in più connessioni TCP**



INFOCOM Dept.

La connessione TCP

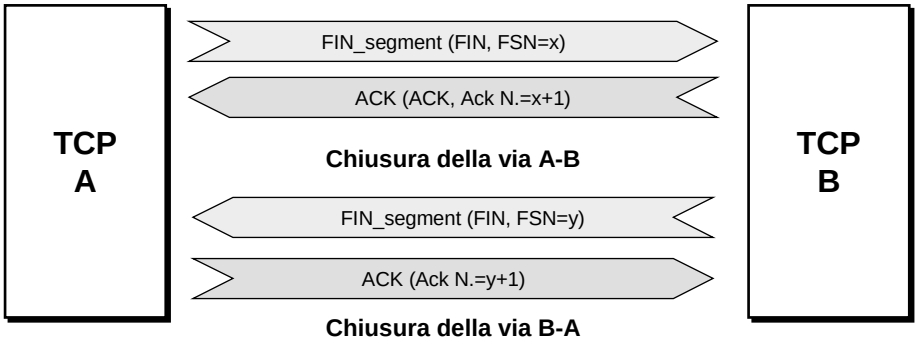
- **La sincronizzazione avviene con un meccanismo detto "three way handshaking"**



INFOCOM Dept.

La connessione TCP

- **Nella fase di rilascio le due vie sono chiuse indipendentemente**



INFOCOM Dept.

Controllo di errore

- **Il TCP prevede esclusivamente riscontri positivi (ACK)**
- **La ritrasmissione dei segmenti è innescata dalla mancata ricezione degli ACK entro un fissato tempo limite (*Timeout*)**
- **Il dimensionamento del timeout è un aspetto critico nelle prestazioni del TCP**
  - **se il suo valore è troppo piccolo, alcuni segmenti in ritardo a causa di congestione, potrebbero considerati persi e quindi ri-trasmessi con conseguente perdita di efficienza**
  - **se il suo valore è troppo grande, la risposta ad un evento di perdita sarebbe troppo lenta con conseguente perdita di efficienza**

INFOCOM Dept.

Controllo di errore

- **Il *Retransmission TimeOut* (RTO) è determinato con uno schema adattativo**
- **Il TCP misura dinamicamente il *Round Trip Time* (RTT)**
  - **RTT = ritardo tra l'invio di un segmento e la ricezione del relativo ACK**
- **Il valore di RTO è scelto maggiore del valore medio osservato del RTT**
- **La misura del RTT è affetta dai seguenti errori**
  - **l'emissione degli ACK da parte del ricevente può essere non immediata**
  - **se è stata effettuata una ritrasmissione è impossibile distinguere se l'ACK si riferisce alla trasmissione iniziale o alla ritrasmissione**
  - **lo stato di congestione della rete può cambiare molto rapidamente**

INFOCOM Dept.

Calcolo del Retransmission TimeOut (RTO)

- **Il Round Trip Time è misurato segmento per segmento**
  - **il RTT di un segmento è misurato come l'intervallo di tempo tra l'istante di emissione del segmento e quello di ricezione del relativo ACK**
- **Nella specifica originale si utilizza una media pesata di RTT, denominata *Smoothed Round Trip Estimate* (SRTT)**
  - **il valore di SRTT al passo k è dato da (il valore raccomandato per il parametro α è 0.9)**

$$SRTT(k+1) = \alpha \ SRTT(k) + (1-\alpha) \ RTT(k+1)$$

$$SRTT(1) = RTT(1)$$

- **Il valore di RTO è dato da (il valore raccomandato per il parametro β detto "delay variance factor" è 2)**

$$RTO(k) = \beta \ SRTT(k)$$

Reti di Telecomunicazioni - Vincenzo Eramo - A.A. 2004/2005

INFOCOM Dept.

Exponential RTO Backoff

- **Determina il valore di RTO associato ad un segmento ritrasmesso**
  - **è consigliabile variare RTO sui segmenti ritrasmessi perché l'esaurimento del timeout è dovuto a congestione in rete**
- **Una sorgente TCP aumenta il valore di RTO per ogni ritrasmissione (*exponential backoff process*) (normalmente q=2)**

$$RTO_{i+1} = q \cdot RTO_i$$

INFOCOM Dept.

Karn Algorithm

- **In caso di ritrasmisssione TCP non distingue se il riscontro si riferisce (*retransmission ambiguity problem*)**
  - **alla prima trasmissione del segmento, timeout troppo elevato con perdita di efficienza e inutili ritardi**
  - **alla ritrasmissione del segmento, timeout troppo breve e quindi ritrasmissioni eccessive e nuovi errori di misura**
- **L'algoritmo di Karn stabilisce di**
  - **non considerare il RTT dei segmenti ritrasmessi**
  - **usare come RTO il valore dato dalla procedura di exponential backoff**
  - **ricalcolare il nuovo valore di RTO solo al momento della ricezione di un ACK di un segmento non ritrasmesso**

INFOCOM Dept.

Controllo di Flusso e di Congestione

- **Il controllo di flusso ha lo scopo di limitare il tasso di generazione dei dati da parte di un host**
  - **tale meccanismo è indispensabile in Internet dove sono presenti host di potenzialità molto diverse**
- **Il controllo della congestione ha lo scopo di recuperare situazioni di sovraccarico nella rete**

INFOCOM Dept.

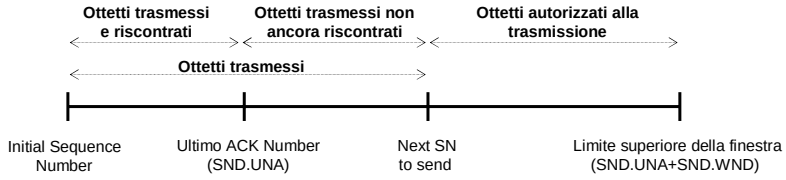
Controllo di Flusso

- **TCP utilizza un controllo di flusso a finestra basato su finestra scorrevole di ampiezza variabile**
- **Il controllo di flusso opera a livello di ottetti (byte)**
- **Gli ottetti sono numerati sequenzialmente a partire dal numero scelto durante il 3-way handshaking**
- **Un riscontro (ACK Number=X e Window=W) significa che**
  - **sono riscontrati tutti gli ottetti ricevuti fino a quello numerato con X-1**
  - **il trasmittente è autorizzato a trasmettere fino a ulteriori w ottetti, ovvero fino all'ottetto numerato con X+W-1**

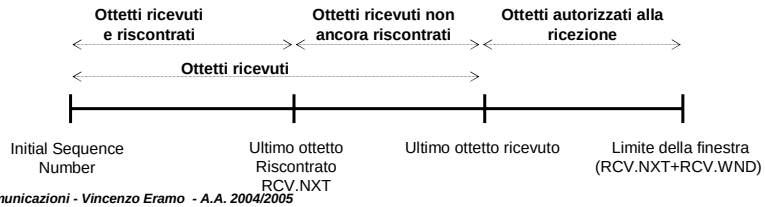
INFOCOM Dept.

Controllo di Flusso

- **Puntatori per il controllo a finestra lato emittente**

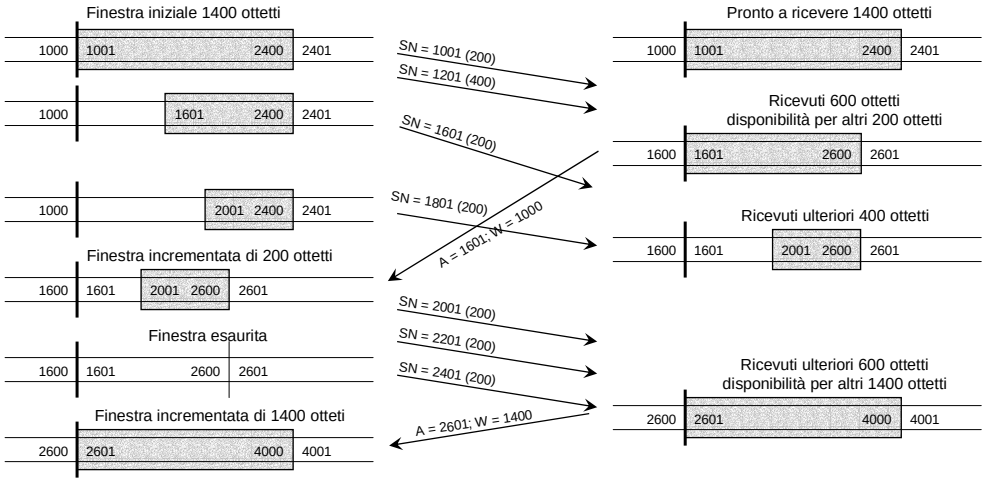


- **Puntatori per il controllo a finestra lato ricevente**



INFOCOM Dept.

Controllo di Flusso



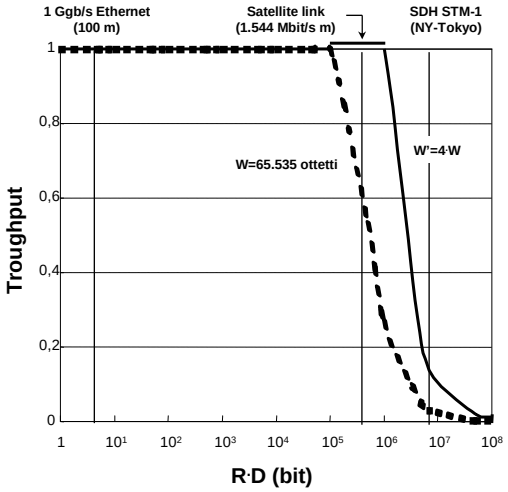
Reti di Telecomunicazioni - Vincenzo Eramo - A.A. 2004/2005

INFOCOM Dept.

Controllo di Flusso

- **Il throughput (S) di una connessione TCP dipende da:**
  - **dimensione della finestra (W)**
  - **ritardo di propagazione (D)**
  - **bit rate (R)**

$$S = \begin{cases} 1 & \text{se} \quad W > RD/4 \\ \dfrac{4 \cdot W}{R \cdot D} & \text{se} \quad W < RD/4 \end{cases}$$



INFOCOM Dept.

Controllo di congestione

- **Ha lo scopo di recuperare situazioni di sovraccarico nella rete limitando il traffico offerto alla rete**
- **Difficoltà:**
  - **il protocollo IP (protocollo di rete) non possiede alcun meccanismo per rivelare e controllare la congestione**
  - **il TCP è un protocollo end-to-end e può rivelare e controllare la congestione solo in modo indiretto**
  - **la rete non coopera con gli host per il controllo della congestione**
  - **la conoscenza dello stato della rete da parte delle entità TCP è imperfetta a causa dei ritardi di rete**
  - **le entità TCP che usano la rete non cooperano tra loro, anzi competono per l'uso delle risorse distribuite**

INFOCOM Dept.

Controllo di congestione

- **In caso di congestione, il controllo di flusso a finestra protegge implicitamente, oltre al destinatario, anche la rete**
  - **se la rete è congestionata arriveranno meno riscontri e quindi saranno emessi un numero minore di segmenti**
  - **il meccanismo adattativo di timeout evita ritrasmissioni che porterebbero ad un aumento della congestione invece che ad una sua diminuzione**

INFOCOM Dept.

Controllo di congestione

- **Sono definiti dei meccanismi addizionali**
  - **TCP utilizza la stima di RTT come misura di congestione, lo scadere del timeout di ritrasmissione è considerato un sintomo di congestione**
- **Esistono varie implementazioni di TCP**
  - **Berkeley**
  - **Tahoe**
  - **Reno**

	Meccanismo	TCP Berkeley	TCP Tahoe	TCP Reno
Window	Slow Start	◆	◆	◆
	Congestion Avoidance	◆	◆	◆
	Fast retransmit		◆	◆
	Fast recovery			◆

Reti di Telecomunicazioni - Vincenzo Eramo - A.A. 2004/2005

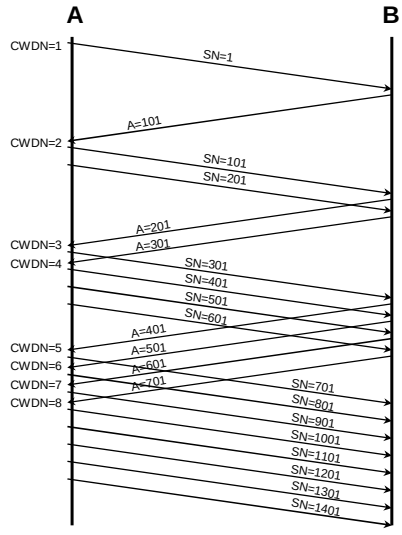
INFOCOM Dept.

Slow Start

- **Tende ad evitare l'insorgere di congestione durante la fase di avvio di una connessione**
- **Regola l'emissione dei segmenti all'inizio di una connessione e ha lo scopo di raggiungere il ritmo di emissione a regime senza causare congestione**
- **Si definisce una *Congestion Window* (cwdn) (misurata in segmenti) che tende ad aumentare progressivamente**
- **La congestion window limita il valore della finestra fino a che questo non sia fissato dalla ricezione degli ACK**

INFOCOM Dept.

Slow Start



- **L'ampiezza della finestra (awnd) in segmenti è**

$$awdn = \min[credit, cwdn]$$

  - **credit:**
    - numero di crediti (in segmenti) concessi nell'ultimo ACK
  - **cwdn:**
    - congestion window (in segmenti)
  - **per il primo segmento**
    - cwdn=1
  - **per ogni segmento riscontrato**
    - cwdn=cwdn+1

Reti di Telecomunicazioni - Vincenzo Eramo - A.A. 2004/2005

Congestion Avoidance

- **Regola l'ampiezza della finestra in caso di congestione di rete che non permette di arrivare al valore di finestra indicato dal ricevitore**
- **Una procedura identica a quella di slow start è troppo aggressiva in caso di congestione**
- **Il meccanismo di congestion avoidance**
  - **è innescato in caso di esaurimento del timeout e quindi di ritrasmissione di un segmento**
  - **richiede la definizione di un parametro detto *Slow Strat Threshold Size* (ssthresh)**

Congestion Avoidance

- **Procedura**
  - **Si innesca quando c'è la scadenza di un time-out**
  - **il valore iniziale di ssthresh è dato da**

$$ssthresh = \left\lfloor \frac{cwdn}{2} \right\rfloor$$

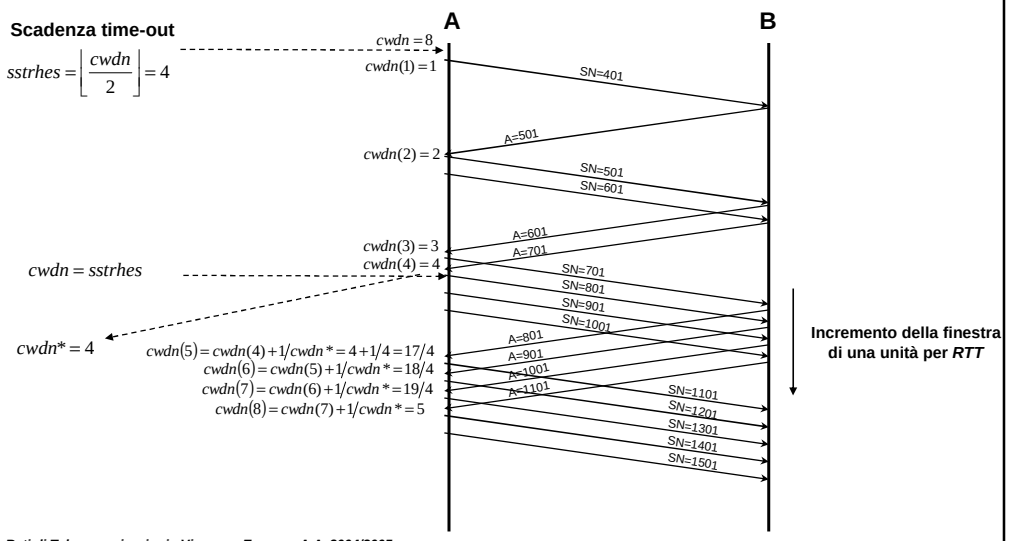
  - **Il valore di cwnd viene posto ad 1**
  - **si esegue la procedura slow start fino a che**

$$cwdn \leq ssthresh$$

  - **se cwdn ≥ sstresh, cwdn è incrementato di 1/cwdn* (bytes) ad ogni ricezione di ACK,essendo cwdn* il valore di finestra al ciclo precedente**
  - **lo scopo è quello di aumentare la finestra al più di un segmento ogni round trip time**

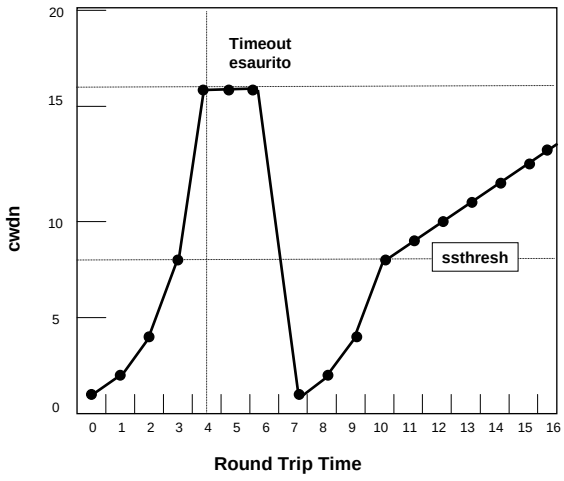
INFOCOM Dept.

Congestion Avoidance (1) (Evoluzione della finestra)



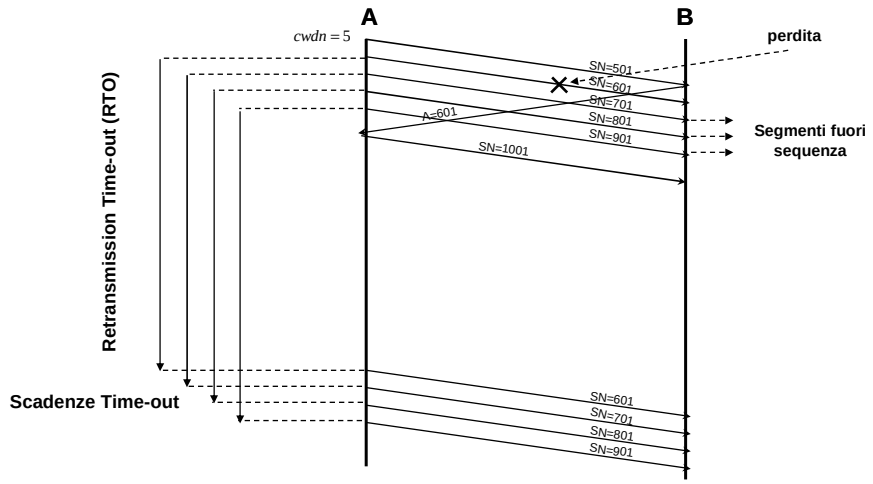
INFOCOM Dept.

Congestion Avoidance (2) (Evoluzione della finestra)



INFOCOM Dept.

Fast Retransmit (1)



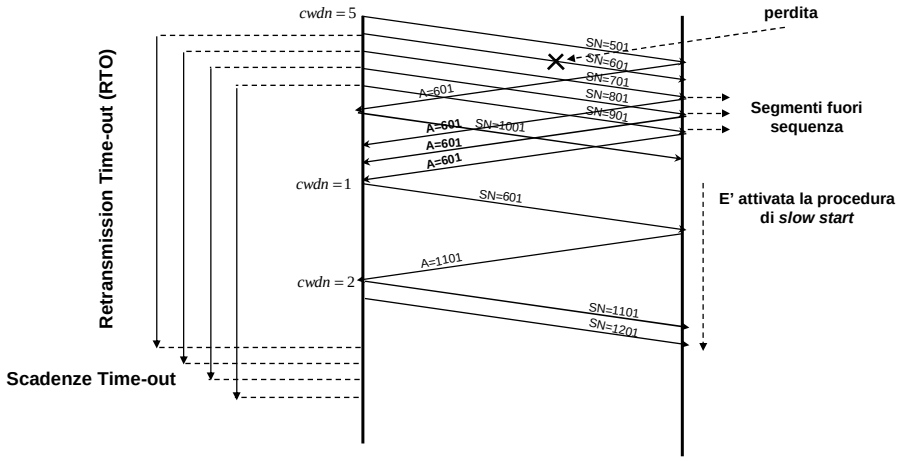
- La perdita occassionale di un pacchetto, dovuta ad una congestione di rete lieve, può causare la ri-trasmissione di un consistente numero di pacchetti

INFOCOM Dept.

Fast Retransmit (2)

- **Migliora le prestazioni in caso di perdita di un singolo segmento**
  - **velocizza la ritrasmissione del segmento perso**
  - **evita la ritrasmissione dei segmenti successivi**
- **Procedura:**
  - **il ricevitore emette un ACK non appena rivela un fuori sequenza ed emette un ACK per ogni segmento successivo fuori sequenza**
  - **la ricezione di tre ACK duplicati è considerato sintomo di un segmento perso**
    - la scelta di tre ACK tende ad evitare il caso in cui il segmento successivo a quello riscontrato abbia subito un ritardo così elevato da aver causato un fuori sequenza
  - **la ritrasmissione del segmento inizia non appena sono ricevuti quattro ACK del segmento precedente anche se il timeout non è scaduto**

Fast Retransmit (3)



Fast Recovery

- **Evita l'innesco della procedura standard di congestion avoidance associata alla procedura di fast retransmission**
  - **l'arrivo di ACK multipli assicura che i segmenti ricevuti sono stati ricevuti e quindi la congestione è stata superata**
- **Rispetto alla procedura di congestion avoidance**
  - **il valore iniziale di cwdn è maggiore**
  - **l'incremento di cwdn è sempre lineare**
  - **si evita la fase iniziale di aumento esponenziale di cwdn (slow start)**

Fast Recovery

La procedura è la seguente

- **quando sono stati ricevuti tre ACK duplicati**
  - si pone

    $$sstresh = \lfloor cwdn/2 \rfloor$$

  - viene ritrasmesso il segmento perduto
  - per tener conto dei segmenti già ricevuti si pone

    $$cwdn = sstresh + 3$$

- **ogni volta che arriva un ACK duplicato, il valore di cwdn viene incrementato di uno e trasmesso (se possibile) un segmento**
- **quando viene ricevuto un ACK non duplicato (riscontro cumulativo)**
  - si pone

    $$cwdn = ssthresh$$

  - la finestra è aggiornata come nella procedura di congestion avoidance