Esercitazione #3

Antonio Brogi

Dipartimento di Informatica
Università di Pisa

Sintesi

Introduzione alle reti di calcolatori

- Ritardi HA1/Q1 + ES2/Q

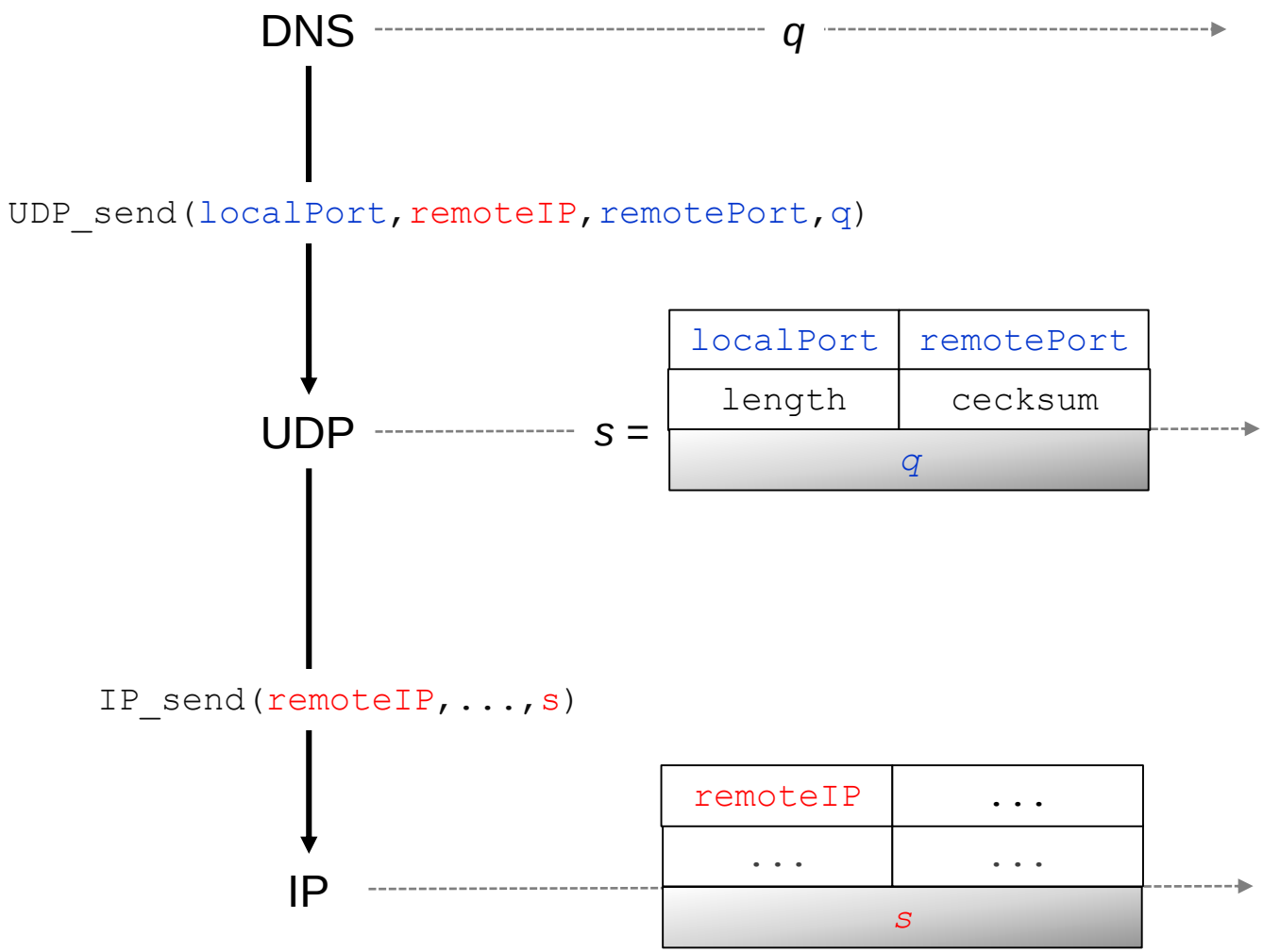
Livello application

- HTTP HA1/E1
- FTP HA1/E2 + AS1
- Email HA1/Q2 + AS2
- DNS HA2/Q3 + AS3

Livello transport

- UDP HA3/Q1
- S&W
- GBN HA3/E1
- SR HA3/Q2

Relazioni tra i livelli: esempio



Come descritto a pag. 131 del testo, UDP calcola il checksum su:

- uno pseudoheader IP,
- preambolo segmento UDP e
- parte dati segmento UDP.

Pseudoheader	Indirizzo IP a 32 bit del mittente		
	Indirizzo IP a 32 bit del destinatario		
	Tutti 0	Protocollo (8 bit)	Lunghezza totale pacchetto UDP (16 bit)
Header	Numero di porta del mittente (16 bit)		Numero di porta del destinatario (16 bit)
	Dimensione totale del pacchetto UDP (16 bit)		Checksum (16 bit)

Ciò che il testo non menziona esplicitamente è che lo pseudoheader è solo "conceptually prefixed" (RFC 768), ovvero viene solo utilizzato per generare (e per verificare) il checksum, e viene quindi scartato.

Q: Perché UDP e TCP utilizzano uno pseudoheader IP per calcolare il checksum?

A: Per controllare anche i casi di «misrouted datagrams»

Q: L'uso dello pseudoheader IP per calcolare il checksum ha qualche svantaggio?

A: Maggior tempo necessario per generare e verificare checksum e violazione del principio di layering (UDP e TCP utilizzano informazioni del protocollo sottostante IP)

Q: Determinare per quale motivo UDP e TCP si preoccupano di controllare l'integrità di informazioni di livello network se anche IP usa un suo checksum.

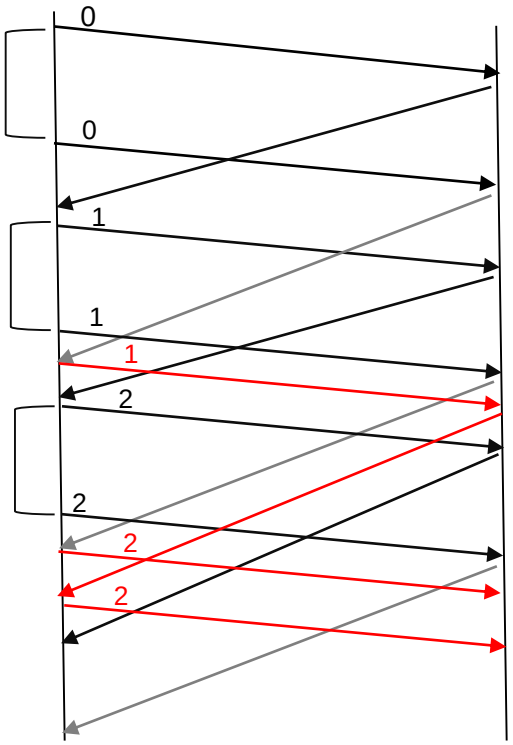
Approfondimento suggerito #6

Livello transport

Approfondimento suggerito #4

Determinare che cosa succederebbe se il mittente Stop&Wait, quando riceve un ACK corrotto o contenente un valore riscontrato diverso da quello atteso, rispedisse immediatamente il segmento.

Nel caso di timeout prematuri ogni copia «extra» causa l'invio di un riscontro «extra», che a sua volta causa l'invio di un'ulteriore copia «extra». Il numero di volte che l'n-esimo segmento viene spedito cresce al crescere di n ...

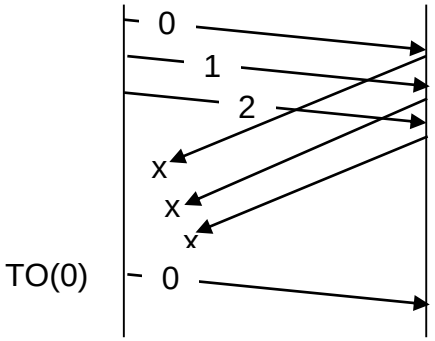


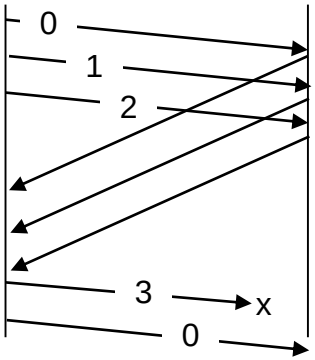
Livello transport

Approfondimento suggerito #5

Determinare quale rapporto deve esistere tra la dimensione dello spazio dei numeri di sequenza e la dimensione della finestra in SR.

Esempio: se #sequenceNumbers=4 e N=3, il ricevente non distinguerebbe i seguenti scenari!





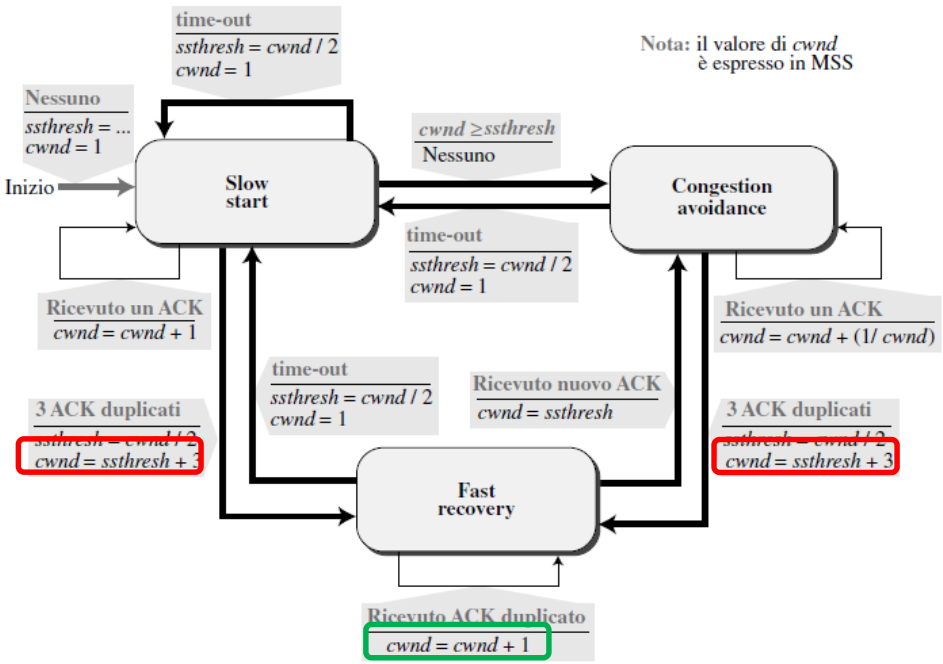
Consideriamo la situazione di massimo “disallineamento” possibile:

- mittente considera in volo i segmenti [X,X+N-1] e
- ricevente li considera invece ricevuti e attende quindi di ricevere [X+N,X+2N-1].

Per non avere problemi, l’intervallo dei numeri di sequenza usati deve essere almeno 2N ovvero la dimensione della finestra non può essere maggiore di #sequenceNumbers/2.

Livello transport

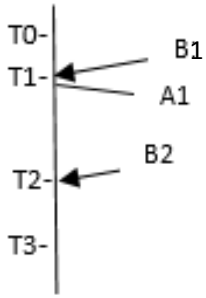
HA4/ Q1. Al tempo t0 il TCP di un host A ha già stabilito una connessione con il TCP di un altro host B, ha 3 MSS di dati in volo (spediti in tre segmenti *full-sized*), il numero di sequenza del segmento più vecchio in volo è X, e riceve un riscontro R. Subito dopo tale ricezione, il TCP si trova nello stato di fast recovery, il valore di *cwnd* è 6 MSS e quello di *ssthresh* è 2 MSS. Indicare –giustificando la risposta– quali sono i possibili valori di R.ackNum.

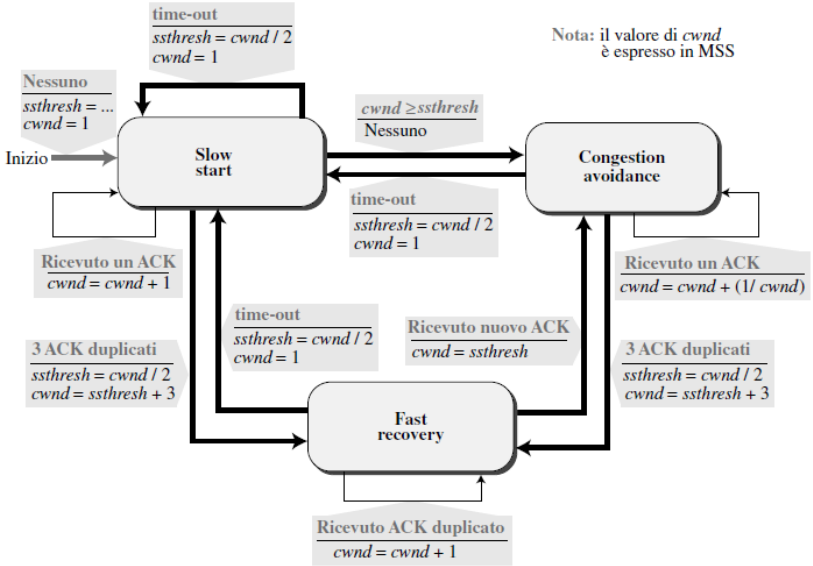


Se subito dopo la ricezione di R il TCP si trova nello stato di *fast recovery* con *cwnd*=6 MSS e *ssthresh*=2 MSS, ciò implica che il TCP si trovava già in *fast recovery* in t0 (altrimenti *cwnd* dovrebbe essere *ssthresh*+3 MSS dopo la ricezione di R).

Poiché dopo la ricezione di R il TCP rimane in *fast recovery*, R è necessariamente un riscontro duplicato, ovvero R.ackNum≤X.

HA4/E1. Al tempo T0 il TCP di un host A ha già stabilito una connessione con il TCP di un altro host B, ha 2 MSS di dati in volo (spediti in due segmenti *full-sized*), 2 MSS di dati non ancora spediti, il numero di sequenza del segmento più vecchio in volo è X, la dimensione della finestra di congestione *cwnd* è 6 MSS e il valore di *ssthresh* è 8 MSS. Al tempo T1>T0 riceve da B un riscontro B1 e in conseguenza ciò invia un segmento A1 contenente 1 MSS di dati. Al tempo T2>T1 riceve da B un altro riscontro B2. Sapendo che al tempo T3>T2 *cwnd=ssthresh* e che nell'intervallo [T0,T3] non si verifica alcun altro evento oltre quelli sopra menzionati, indicare ‑giustificando la risposta‑ i possibili valori dei campi *ackNum* e *rwnd* contenuti in B1 e B2.



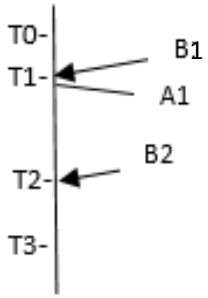


$cwnd_{t0} < ssthresh_{t0} \rightarrow$ TCP in SS in t0.

(1) Se B1 riscontro non duplicato allora dopo avere ricevuto B1: cwnd=7 MSS e TCP rimane in SS. Affinché $cwnd_{t3}=ssthresh_{t3}$ anche B2 riscontro non duplicato.

- Se **B1.ackNum=X+1 MSS** allora dopo avere ricevuto B1: **min(cwnd,B1.rwnd)=2 MSS** dato che viene spedito 1 solo MSS di nuovi dati. Quindi **B1.rwnd=2MSS**.
  - Se **B2.ackNum=X+2 MSS** allora dopo avere ricevuto B2: **min(cwnd,B2.rwnd)≤1 MSS** dato che non vengono spediti nuovi dati. Quindi **B2.rwnd ≤ 1MSS**.
  - Se **B2.ackNum=X+3 MSS** allora dopo avere ricevuto B2: **min(cwnd,B2.rwnd)=0 MSS** e **B2.rwnd=0 MSS**.

HA4/E1. Al tempo T0 il TCP di un host A ha già stabilito una connessione con il TCP di un altro host B, ha 2 MSS di dati in volo (spediti in due segmenti *full-sized*), 2 MSS di dati non ancora spediti, il numero di sequenza del segmento più vecchio in volo è X, la dimensione della finestra di congestione *cwnd* è 6 MSS e il valore di *ssthresh* è 8 MSS. Al tempo T1>T0 riceve da B un riscontro B1 e in conseguenza ciò invia un segmento A1 contenente 1 MSS di dati. Al tempo T2>T1 riceve da B un altro riscontro B2. Sapendo che al tempo T3>T2 *cwnd=ssthresh* e che nell'intervallo [T0,T3] non si verifica alcun altro evento oltre quelli sopra menzionati, indicare -giustificando la risposta- i possibili valori dei campi *ackNum* e *rwnd* contenuti in B1 e B2.

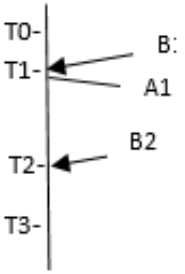


E2. Osserviamo che in TO il TCP si trova nello stato di slow start dato che cwnd < ssthresh. Analizziamo i vari casi possibili.

(1) Se B1 è un riscontro non duplicato allora, subito dopo avere ricevuto B1, TCP rimane in slow start e cwnd=7MSS. Affinchè in T3 cwnd=ssthresh, anche B2 deve essere un riscontro non duplicato; in tal caso infatti, subito dopo avere ricevuto B2, cwnd=8MSS.

- Se B1.ackNum=X+1MSS allora, dopo avere ricevuto B1, min(cwnd,B1.rwnd)=2MSS dato che viene spedito 1 solo MSS di nuovi dati, quindi B1.rwnd=2MSS.
  - Se B2.ackNum=X+2MSS allora, dopo avere ricevuto B2, min(cwnd,B2.rwnd)≤1MSS dato che non vengono spediti nuovi dati, quindi B2.rwnd≤1MSS.
  - Analogamente, se B2.ackNum=X+3MSS allora, dopo avere ricevuto B2, min(cwnd,B2.rwnd)=0 MSS e B2.rwnd=0MSS.
- Se B1.ackNum=X+2MSS allora, dopo avere ricevuto B1, min(cwnd,B1.rwnd)=1MSS dato che viene spedito 1 solo MSS di nuovi dati, quindi B1.rwnd=1MSS. Poiché B2.ackNum=X+3MSS allora, dopo avere ricevuto B2, min(cwnd,B2.rwnd)=0 MSS dato che non vengono spediti nuovi dati, quindi B2.rwnd=0MSS.

HA4/ **E1.** Al tempo T0 il TCP di un host A ha già stabilito una connessione con il TCP di un altro host B, ha 2 MSS di dati in volo (spediti in due segmenti *full-sized*), 2 MSS di dati non ancora spediti, il numero di sequenza del segmento più vecchio in volo è X, la dimensione della finestra di congestione *cwnd* è 6 MSS e il valore di *ssthresh* è 8 MSS. Al tempo T1>T0 riceve da B un riscontro B1 e in conseguenza ciò invia un segmento A1 contenente 1 MSS di dati. Al tempo T2>T1 riceve da B un altro riscontro B2. Sapendo che al tempo T3>T2 *cwnd=ssthresh* e che nell'intervallo [T0,T3] non si verifica alcun altro evento oltre quelli sopra menzionati, indicare - giustificando la risposta- i possibili valori dei campi *ackNum* e *rwnd* contenuti in B1 e B2.



(2) Se B1 è un riscontro duplicato ricevuto per la terza volta allora, subito dopo avere ricevuto B1, TCP passa in fast recovery, ssthresh=3MSS e cwnd=6MSS. Affinchè in T3 cwnd=ssthresh, B2 deve essere un riscontro non duplicato; in tal caso infatti, subito dopo avere ricevuto B2, cwnd=3MSS.

Osserviamo che B1.ackNum=X e il valore di B1.rwnd è ininfluente (dato che scatta il meccanismo di fast retransmit).

- Se B2.ackNum=X+1MSS allora, dopo avere ricevuto B2, min(cwnd,B2.rwnd)=1MSS dato che non vengono spediti nuovi dati, quindi B2.rwnd=1MSS.

Analogamente, se B2.ackNum=X+2MSS allora, dopo avere ricevuto B2, min(cwnd,B2.rwnd)=0MSS e B2.rwnd=0MSS.

Sintesi

Introduzione alle reti di calcolatori

- Ritardi HA1/Q1 + ES2/Q
- Livelli ES3/esempio

Livello application

- HTTP HA1/E1
- FTP HA1/E2 + AS1
- Email HA1/Q2 + AS2
- DNS HA2/Q3 + AS3

Livello transport

- UDP HA3/Q1
- S&W AS4
- GBN HA3/E1
- SR HA3/Q2 + AS5
- TCP HA4/Q1 + HA4/E1

HA5