

RETI DI CALCOLATORI – Home Assignment protocollo TCP

Prima parte

Q1. Un server Web S invia a un client C una pagina formata da tre oggetti, aventi dimensione 3,5 MSS, 2,3 MSS, e 4 MSS, rispettivamente. Supponiamo che S e C utilizzino HTTP 1.1 con connessioni TCP persistenti, e che il TCP di S si trovi nello stato di slow start con threshold uguale a 10 MSS e con valore della finestra di congestione cwnd uguale a 1 MSS. Supponiamo inoltre che tutti i segmenti siano riscontrati positivamente da C, che nessun ACK sia cumulativo, e che tali riscontri arrivino tutti correttamente, e nell'ordine con cui sono stati spediti, a S. Indicare – giustificando la risposta – la successione dei valori assunti da cwnd. Assumere per semplicità che non scada alcun timeout.

Q2. Supponiamo che il TCP di un host A abbia stabilito una connessione con un suo pari e che, trovandosi nello stato di congestion avoidance, dopo avere ricevuto un riscontro R non duplicato, aggiorni la dimensione della sua finestra di congestione a $10/3$ MSS. Indicare – giustificando la risposta– quale era la dimensione della finestra di congestione del TCP di A subito prima di ricevere R.

Q3. Al tempo t un client A invia a B un segmento S1 contenente il flag FIN attivo e subito dopo riceve da B un segmento S2 contenente anch'esso il flag FIN attivo. Indicare – giustificando la risposta – i valori dei campi seqNum e ackNum nei segmenti S1, S2 e nei successivi segmenti inviati o ricevuti da A. Supporre che il payload di S1 e S2 sia vuoto, che al tempo t l'ultimo byte di dati inviato da A a B avesse numero di sequenza w, che l'ultimo byte di dati inviato da B a A avesse numero di sequenza z. Supporre che non si verifichino perdite di pacchetti dopo t.

Q4. Al tempo t_0 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.

Q5. Un client C sta chiudendo una connessione TCP con un server S mediante chiusura a tre vie. Supponiamo che il numero di sequenza dell'ultimo segmento che C ha inviato ad S prima di iniziare la chiusura fosse 389417, mentre quello inviato da S a C sia 2017534, che C invii un altro segmento ad S all'inizio della chiusura, e che tutti i segmenti siano full sized, con dimensione massima uguale a 4096 byte, indicare – giustificando la risposta - i valori dei campi seq, ack, A ed F nei tre segmenti relativi alla chiusura.

Q6. Il TCP di un host si trova nello stato di fast recovery, con ssthresh=2,5 MSS, cwnd=6,5 MSS, rwnd=3 MSS, e ha 3 MSS di dati in volo e 2 MSS di nuovi dati da spedire. Indicare –giustificando la risposta– il numero minimo di riscontri che il TCP deve ricevere per passare prima nello stato di congestion avoidance, quindi in slow start e infine tornare in fast recovery.

Q7. Al tempo t i valori di $cwnd$ e $ssthresh$ del TCP di un host A per una connessione c sono tali che:

$$ssthresh < cwnd < ssthresh + 1 \text{ MSS}$$

Indicare – giustificando la risposta – quale è il valore assunto da $cwnd$ se al tempo t viene ricevuto un riscontro.

Q8. Sia $cwnd(i)$ l' i -esimo valore assunto dalla variabile $cwnd$ del TCP di un host H. Se $cwnd(i) = 2,5$, quali sono tutti i valori possibili per $cwnd(i-1)$, nonché gli stati relativi? Giustificare la risposta.

Q9. Sia $cwnd(i)$ l' i -esimo valore assunto dalla variabile $cwnd$ del TCP di un host H. Se $cwnd(i) = 2,5$, quali sono tutti i valori possibili per $cwnd(i+1)$, nonché gli stati relativi? Giustificare la risposta.

E1. Al tempo t_0 il TCP di un client SMTP A ha una connessione già stabilita, per la quale:

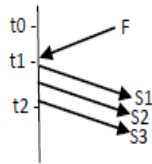
- ha 3 segmenti full-sized in volo, con $Sf=X$,
- ha 3 MSS di nuovi dati da spedire,
- ha come valori delle variabili $cwnd$ e $rwnd$ 3 MSS e della variabile $ssthresh$ 4 MSS, e
- si trova nello stato di slow start.

Al tempo t_1 il TCP di A riceve un riscontro in conseguenza del quale non invia nuovi dati, e la stessa cosa avviene al tempo t_2 . Al tempo t_3 riceve un riscontro R non duplicato, il cui campo $rwnd$ ha valore 2MSS, e subito dopo avere ricevuto R il TCP di A invia al tempo t_4 due segmenti, contenenti 1 MSS e 0,5 MSS di dati rispettivamente. Indicare – giustificando la risposta – lo stato e il valore di $cwnd$ del TCP di A al tempo t_4 e il valore del campo $ackNum$ in R. Supporre che nell'intervallo $[t_0, t_4]$ non scada alcun timeout.

E2. Al tempo t_0 il TCP di un host A ha una connessione già stabilita, per la quale ha 2 segmenti full sized in volo e 2 MSS di nuovi dati da spedire, si trova nello stato di congestion avoidance, con $Sf=Y$, $ssthresh=4$ MSS, $cwnd=17/4$ MSS, e $rwnd=2$ MSS. Al tempo t_1 riceve un riscontro duplicato per la terza volta (ovvero riceve per la quarta volta lo stesso riscontro), al tempo t_2 riceve di nuovo un riscontro duplicato, al tempo t_3 scatta un timeout, mentre ai tempi t_4 e t_5 vengono ricevuti due riscontri non duplicati. Indicare – giustificando la risposta – lo stato del TCP e i valori di $ssthresh$, $cwnd$ e $rwnd$ subito dopo i tempi t_1 , t_2 , t_3 e t_4 , supponendo che nell'intervallo $[t_0, t_5]$ il TCP invii solo un segmento S contenente nuovi dati, subito dopo t_2 e che nell'intervallo $[t_0, t_5]$ scatti un solo timeout (in t_3).

E3. Al tempo t_0 il TCP di un server A ha una connessione già stabilita, per la quale:

- ha 2 segmenti full-sized in volo, con $Sf=X$,
- ha 3 MSS di nuovi dati da spedire,
- ha come valori delle variabili $cwnd$ e $rwnd$ 3 MSS e 2 MSS, rispettivamente, e
- si trova nello stato di slow start.



Al tempo t_1 il TCP di A riceve un segmento F con flag FIN a 1 e quindi spedisce in sequenza 3 segmenti S1, S2, S3, l'ultimo dei quali subito prima del tempo t_2 . Supponendo che nell'intervallo $[t_0, t_2]$ non scatti alcun timeout indicare –giustificando la risposta– i valori delle variabili $cwnd$ e $rwnd$ e lo stato del TCP di A al tempo t_2 .

E4. Al tempo t il TCP di un processo applicativo A non ha dati in volo e il valore della sua variabile S_f è X . Sapendo che:

- B1 contiene il flag FIN a 1, $seqNum=Z$ e non contiene dati
- A1 contiene il flag ACK a 1, il flag FIN a 0 e non contiene dati
- A2, A3 e A4 non sono risedizioni di A1 e hanno i flag ACK e FIN a 0
- A5 contiene il flag FIN a 1, il flag ACK a 0 e non contiene dati

indicare – giustificando la risposta – i possibili valori del campo $seqNum$ dei segmenti A2, A3, A4, A5 e del campo $ackNum$ dei segmenti A1, B2 e B3.



Q1. Un server Web S invia a un client C una pagina formata da tre oggetti, aventi dimensione 3,5 MSS, 2,3 MSS, e 4 MSS, rispettivamente. Supponiamo che S e C utilizzino HTTP 1.1 con connessioni TCP persistenti, e che il TCP di S si trovi nello stato di slow start con threshold uguale a 10 MSS e con valore della finestra di congestione cwnd uguale a 1 MSS. Supponiamo inoltre che tutti i segmenti siano riscontrati positivamente da C, e che tali riscontri arrivino tutti correttamente, e nell'ordine con cui sono stati spediti, a S. Indicare – giustificando la risposta – la successione dei valori assunti da cwnd. Assumere per semplicità che non scada alcun timeout.

Q1. S invierà a C i tre oggetti di dimensione 3,5 MSS, 2,3 MSS e 4 MSS in 4, 3 e 4 segmenti, rispettivamente. Il TCP di S incrementerà di 1 MSS la dimensione di cwnd ogni volta che riceverà un ACK di uno dei primi 9 segmenti, ottenendo così $cwnd=10$ MSS. Passato nello stato di congestion avoidance, il TCP di S aggiornerà quindi cwnd a $(10 + 1/10)$ MSS = 101/10 MSS quando riceverà l'ACK del decimo segmento e a $(101/10 + 10/101)$ MSS = 10301/1010 MSS quando riceverà l'ACK dell'undicesimo segmento.

Q2. Supponiamo che il TCP di un host A abbia stabilito una connessione con un suo pari e che, trovandosi nello stato di congestion avoidance, dopo avere ricevuto un riscontro R non duplicato aggiorni la dimensione della sua finestra di congestione a 10/3 MSS. Indicare –giustificando la risposta– quale era la dimensione della finestra di congestione del TCP di A subito prima di ricevere R.

Q2. Sappiamo che, a seguito della ricezione di R, il nuovo valore di CongWin (CW nel seguito) viene determinato applicando la formula: $CW' = CW + 1/CW$ Se quindi CongWin= n MSS abbiamo: $n + 1/n = 10/3$ quindi $n^2 - 10/3n + 1 = 0$ ovvero $n=3$ oppure $n=1/3$. La dimensione della finestra di congestione subito prima della ricezione di R era quindi 3 MSS (CongWin non scende mai sotto 1 MSS).

Q3. Al tempo t un client A invia a B un segmento S1 contenente il flag FIN attivo e subito dopo riceve da B un segmento S2 contenente anch'esso il flag FIN attivo. Indicare – giustificando la risposta – i valori dei campi seqNum e ackNum nei segmenti S1, S2 e nei successivi segmenti inviati o ricevuti da A. Supporre che il payload di S1 e S2 sia vuoto, che al tempo t l'ultimo byte di dati inviato da A a B avesse numero di sequenza w, che l'ultimo byte di dati inviato da B a A avesse numero di sequenza z. Supporre che non si verifichino perdite di pacchetti dopo t.

Q3. Poiché in t il valore di Sequence number di A è w+1 e quello di B è z+1, S1 contiene seqNum=w+1 e ackNum=z+1 mentre S2 contiene seqNum=z+1 e ackNum=w+2. Dopo avere ricevuto S2, A invia a B un segmento con flag ACK attivo, seqNum=w+2 e ackNum=z+2 poiché i segmenti "FIN" che non trasportano dati consumano un numero di sequenza.

Q4. 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

Q4. Se subito dopo la ricezione di R il TCP si trova nello stato di fast recovery e 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 $R.ackNum \leq X$.

Q5. Un client C sta chiudendo una connessione TCP con un server S mediante chiusura a tre vie. Supponiamo che il numero di sequenza dell'ultimo segmento che C ha inviato ad S prima di iniziare la chiusura fosse 389417, mentre quello inviato da S a C sia 2017534, che C invii un altro segmento ad S all'inizio della chiusura, e che tutti i segmenti siano full sized, con dimensione massima uguale a 4096 byte, indicare – giustificando la risposta - i valori dei campi seq, ack, A ed F nei tre segmenti relativi alla chiusura.

**Q5. Da C ad S: seq=393513; ack= 2021630; A=1; F=1 (segmento FIN).
Da S a C: seq=2021630; ack= 393514; A=1; F=1 (segmento FIN+ACK).
Da C ad S: seq=393514; ack= 2021631; A=1; F=0 (segmento ACK).**

Q6. Il TCP di un host si trova nello stato di fast recovery, con ssthresh=2,5 MSS, cwnd=6,5 MSS, rwnd=3 MSS, e ha 3 MSS di dati in volo e 2 MSS di nuovi dati da spedire. Indicare –giustificando la risposta– il numero minimo di riscontri che il TCP deve ricevere per passare prima nello stato di congestion avoidance, quindi in slow start e infine tornare in fast recovery.

Q6. TCP passa da fast recovery a congestion avoidance se riceve un riscontro non duplicato. Se l'evento successivo è lo scadere di un timeout passa in slow start. Se a questo punto riceve tre riscontri duplicati torna in fast recovery. Quindi il numero minimo di riscontri che TCP deve ricevere per passare prima in congestion avoidance, quindi in slow start e infine tornare in fast recovery è 4.

Q7. Al tempo t i valori di cwnd e ssthresh del TCP di un host A per una connessione c sono tali che:

$$ssthresh < cwnd < ssthresh + 1 \text{ MSS}$$

Indicare – giustificando la risposta – quale è il valore assunto da cwnd se al tempo t viene ricevuto un riscontro.

Q7. Al tempo t il TCP di A si trova nello stato congestion avoidance dato che non può trovarsi né nello stato slow start (poiché $ssthresh < cwnd$) né nello stato fast recovery (poiché $cwnd < ssthresh + 1 \text{ MSS}$). Quindi:

- se il riscontro ricevuto è un riscontro non duplicato allora cwnd è posto a $cwnd+1/cwnd$,
- se il riscontro ricevuto è un riscontro duplicato ricevuto per la terza allora cwnd è posto a $cwnd/2 + 3 \text{ MSS}$,
- se il riscontro ricevuto è un riscontro duplicato ricevuto non per la terza allora il valore cwnd non viene modificato.

Q8. Sia cwnd(i) l'i-esimo valore assunto dalla variabile cwnd del TCP di un host H. Se cwnd(i) =2,5, quali sono tutti i valori possibili per cwnd(i-1), nonché gli stati relativi? Giustificare la risposta.

Q8. Siccome cwnd è sempre intero in slow start, e in fast recovery cwnd è > 3, il TCP si trova in congestion avoidance. Il valore di cwnd(i-1) poteva essere 2 ed aver ricevuto un ack non duplicato, sempre nello stato di congestion avoidance, oppure poteva essere in fast recovery, con ssthresh=2.5 ed aver ricevuto un nuovo ack: in questo caso il valore di cwnd(i-1) poteva essere X.5 con X intero e maggiore di 4.

Q9. Sia $cwnd(i)$ l' i -esimo valore assunto dalla variabile $cwnd$ del TCP di un host H. Se $cwnd(i) = 2,5$, quali sono tutti i valori possibili per $cwnd(i+1)$, nonché gli stati relativi? Giustificare la risposta.

Q9. Il valore di $cwnd(i+1)$ può essere $5/2 + 2/5 = 2.9$ se rimane in congestion avoidance e riceve un ack non duplicato, oppure 1 se scade il time out e passa in slow start, oppure $5/4 + 3 = 17/4$ se riceve 3 ack duplicati e passa in fast recovery.

E1. Al tempo t_0 il TCP di un client SMTP A ha una connessione già stabilita, per la quale: - ha 3 segmenti full-sized in volo, con $Sf=X$, - 3 MSS di nuovi dati da spedire, - ha come valori delle variabili $cwnd$ e $rwnd$ 3 MSS e della variabile $ssthresh$ 4 MSS, e - si trova nello stato di slow start. Al tempo t_1 il TCP di A riceve un riscontro in conseguenza del quale non invia nuovi dati, e la stessa cosa avviene al tempo t_2 . Al tempo t_3 riceve un riscontro R non duplicato, il cui campo $rwnd$ ha valore 2MSS, e subito dopo avere ricevuto R il TCP di A invia al tempo t_4 due segmenti, contenenti 1 MSS e 0,5 MSS di dati rispettivamente. Indicare – giustificando la risposta – lo stato e il valore di $cwnd$ del TCP di A al tempo t_4 e il valore del campo $ackNum$ in R. Supporre che nell'intervallo $[t_0, t_4]$ non scada alcun timeout.

E1. Siano R1 e R2 i riscontri ricevuti in t_1 e t_2 e analizziamo i vari casi possibili.

- R1 non può essere un riscontro non duplicato. (Se lo fosse, dopo averlo ricevuto TCP passerebbe in congestion avoidance con $cwnd=4$ MSS e, dopo avere ricevuto R2 e R, $cwnd \geq 4$ MSS e quindi in t_4 $\min(cwnd, rwnd) = 2$ MSS e TCP non invierebbe in alcun caso solo 1,5 MSS di nuovi dati.)
- Se R1 è un riscontro duplicato ricevuto per la terza volta allora dopo averlo ricevuto TCP passa in fast recovery con $ssthresh=1,5$ MSS e $cwnd=4,5$ MSS. A questo punto:
 - Se R2 è un riscontro duplicato ricevuto non per la terza volta, allora dopo averlo ricevuto TCP rimane in fast recovery e $cwnd$ diventa 5,5 MSS. Dopo avere ricevuto R TCP passa quindi in congestion avoidance, $cwnd$ diventa 1,5 MSS e $R.ackNum$ deve quindi avere come valore $X+3$ MSS.
 - R2 non può essere un riscontro non duplicato. (Se lo fosse, allora dopo averlo ricevuto TCP passerebbe in congestion avoidance con $cwnd=1,5$ MSS e, dopo avere ricevuto R, $cwnd$ assumerebbe un valore maggiore di 1,5 MSS e TCP non invierebbe in alcun caso solo 1,5 MSS di nuovi dati.)
- Se R1 è un riscontro duplicato ricevuto non per la terza volta:
 - R2 non può essere un riscontro duplicato ricevuto non per la terza volta. (Se lo fosse allora dopo avere ricevuto R $cwnd=4$ MSS e $\min(cwnd, rwnd) = 2$ MSS e quindi TCP non invierebbe solo 1,5 MSS di nuovi dati.)
 - Se R2 è un riscontro duplicato ricevuto per la terza volta, allora dopo averlo ricevuto TCP passa in fast recovery con $ssthresh=1,5$ MSS e $cwnd=4,5$ MSS. Quindi, dopo avere ricevuto R, TCP passa in congestion avoidance con $cwnd=1,5$ MSS e $R.ackNum$ deve quindi avere valore $X+3$ MSS.
 - R2 non può essere un riscontro non duplicato. (Se lo fosse allora dopo avere ricevuto R2 il TCP passerebbe in congestion avoidance con $cwnd=4$ MSS e dopo avere ricevuto R $cwnd > 4$ MSS e $\min(cwnd, rwnd) = 2$ MSS e quindi TCP non invierebbe solo 1,5 MSS di nuovi dati.)

E2. Al tempo t_0 il TCP di un host A ha una connessione già stabilita, per la quale ha 2 segmenti full sized in volo e 2 MSS di nuovi dati da spedire, si trova nello stato di congestion avoidance, con $S_f=Y$, $ssthresh=4$ MSS, $cwnd=17/4$ MSS, e $rwnd=2$ MSS. Al tempo t_1 riceve un riscontro duplicato per la terza volta (ovvero riceve per la quarta volta lo stesso riscontro), al tempo t_2 riceve di nuovo un riscontro duplicato, al tempo t_3 scatta un timeout, mentre ai tempi t_4 e t_5 vengono ricevuti due riscontri non duplicati. Indicare – giustificando la risposta – lo stato del TCP e i valori di $ssthresh$, $cwnd$ e $rwnd$ subito dopo i tempi t_1 , t_2 , t_3 e t_4 , supponendo che nell'intervallo $[t_0,t_5]$ il TCP invii solo un segmento S contenente nuovi dati, subito dopo t_2 e che nell'intervallo $[t_0,t_5]$ scatti un solo timeout (in t_3).

E2. Subito dopo t_1 il TCP di A passa nello stato di fast recovery, ponendo $ssthresh$ a $17/8$ MSS e $cwnd$ a $41/8$ MSS, mentre $rwnd$ non può essere maggiore di $2MSS$ dato che non vengono spediti nuovi dati. Subito dopo t_2 il TCP pone $cwnd=49/8$ MSS, mentre $rwnd$ non può essere maggiore di $2MSS+dimDati(S)$ dato che viene spedito solo il segmento S contenente nuovi dati. Subito dopo t_3 il TCP passa nello stato di slow start, ponendo $ssthresh$ a $49/16$ MSS e $cwnd$ a 1 MSS, mentre $rwnd$ non può essere maggiore di $2MSS+dimDati(S)$ dato che non vengono spediti nuovi dati. Infine subito dopo t_4 il TCP pone $cwnd$ a 2 MSS, mentre $rwnd$ non può essere superiore a 1 MSS + $dimDati(S)$ o a $dimDati(S)$, a seconda del riscontro ricevuto (*l'ultima riga dell'ultima colonna della tabella sotto è sbagliata e non va considerata: a t_5 si riceve un riscontro nuovo e quindi il riscontro ricevuto a t_4 non può riscontare tutta la finestra - se lo facesse non esisterebbe il riscontro ricevuto a t_5*).

| Subito dopo t_1 | Subito dopo t_2 | Subito dopo t_3 | Subito dopo t_4 |
|-----------------------|----------------------------|-------------------------------|---|
| stato=FR | stato=FR | stato=SS | stato=SS |
| $ssthresh = 17/8$ MSS | $ssthresh = 17/8$ MSS | $ssthresh = 49/16$ MSS | $ssthresh = 49/16$ MSS |
| $cwnd = 41/8$ MSS | $cwnd = 49/8$ MSS | $cwnd = 1$ MSS | $cwnd = 2$ MSS |
| $rwnd \leq 2MSS$ | $rwnd = 2MSS + dimDati(S)$ | $rwnd \leq 2MSS + dimDati(S)$ | $R.ack = Y + 1MSS \rightarrow rwnd \leq 1MSS + dimDati(S)$ $R.ack = Y + 2MSS \rightarrow rwnd \leq dimDati(S)$ $R.ack = Y + 2MSS + dimDati(S) \rightarrow rwnd = 0$ |

E3. Al tempo t_0 il TCP di un server A ha una connessione già stabilita, per la quale: - ha 2 segmenti full-sized in volo, con $S_f=X$, - ha 3 MSS di nuovi dati da spedire, - ha come valori delle variabili $cwnd$ e $rwnd$ 3 MSS e 2 MSS, rispettivamente, e - si trova nello stato di slow start. Al tempo t_1 il TCP di A riceve un segmento F con flag FIN a 1 e quindi spedisce in sequenza 3 segmenti S_1 , S_2 , S_3 , l'ultimo dei quali subito prima del tempo t_2 . Supponendo che nell'intervallo $[t_0,t_2]$ non scatti alcun timeout indicare –giustificando la risposta– i valori delle variabili $cwnd$ e $rwnd$ e lo stato del TCP di A al tempo t_2 .

E3. Dato che il TCP di S invia 3 segmenti dopo avere ricevuto il segmento F, quest'ultimo deve contenere un riscontro non duplicato. A seguito della ricezione di F il TCP di S aggiorna $cwnd$ a 4 MSS e quindi:

- se $F.ackNum = X + 1$ MSS allora $F.rwnd \geq 4$ MSS, dato che il TCP di S invia 3 MSS di nuovi dati
- analogamente, se $F.ackNum = X + 2$ MSS allora $F.rwnd \geq 3$ MSS. Al tempo t_2 il valore di $cwnd$ sarà quindi 4 MSS, il valore di $rwnd$ sarà 4 MSS oppure 3 MSS e il TCP di S si troverà nello stato slow start se $ssthresh > 4$ MSS e nello stato congestion avoidance altrimenti.

E4. Al tempo t il TCP di un processo applicativo A non ha dati in volo e il valore della sua variabile Sf è X . Sapendo che: \square $B1$ contiene il flag FIN a 1, $seqNum=Z$ e non contiene dati \square $A1$ contiene il flag ACK a 1, il flag FIN a 0 e non contiene dati \square $A2, A3$ e $A4$ non sono rispediti di $A1$ e hanno i flag ACK e FIN a 0 \square $A5$ contiene il flag FIN a 1, il flag ACK a 0 e non contiene dati indicare – giustificando la risposta – i possibili valori del campo $seqNum$ dei segmenti $A2, A3, A4, A5$ e del campo $ackNum$ dei segmenti $A1, B2$ e $B3$.

E4. Osserviamo che $A1$ è il riscontro di $B1$ quindi $A1.ackNum=Z+1$, $A2$ trasporta $N2$ byte di nuovi dati, con $A2.seqNum=X$, e $A3$ trasporta altri $N3$ byte di nuovi dati, con $A3.seqNum=X+N2$. A questo punto:

- se $B2.ackNum=X+N2+N3$ oppure se $B2.ackNum=X+N2$ e non scatta alcun timeout, allora $A4$ trasporta $N4$ byte di nuovi dati, con $A4.seqNum=X+N2+N3$, e $B3.ackNum=X+N2+N3+N4=A5.seqNum$;
- se $B2.ackNum=X+N2$ e scatta un timeout allora $A4$ è una ripesizione di $A3$, ovvero $A4.seqNum=X+N2$, e $B3.ackNum=X+N2+N3=A5.seqNum$.