3.1.9 Appello del 03/09/2003

ESERCIZIO 1: La Figura 1.1 illustra la struttura interna di un autonomous

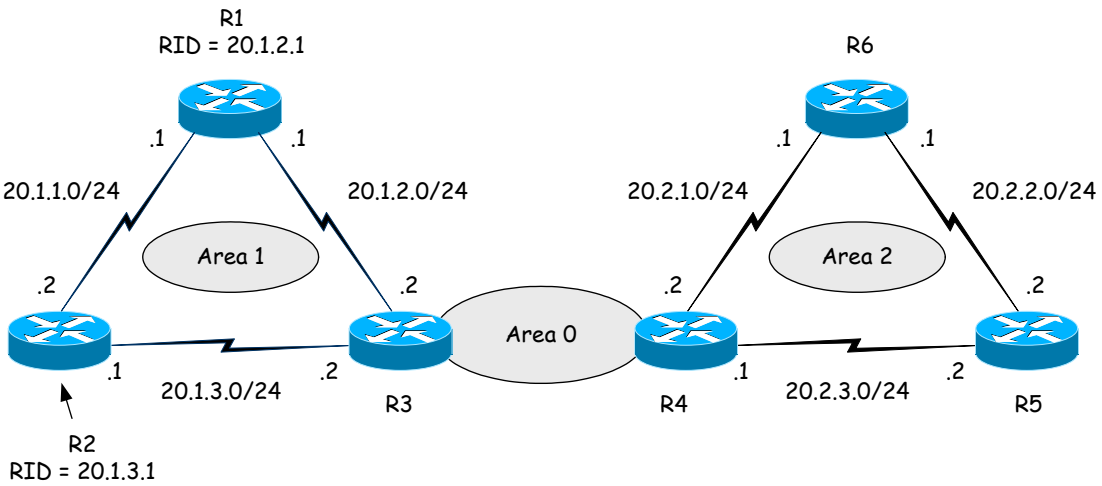


Figura 1.1: Struttura in aree dell'autonomous system

system isolato (cioè non collegato ad Internet). Le LIS dell'Area 1 e dell'Area 2 sono visibili mentre non lo sono sono quelle dell'Area 0. Per conoscere la struttura interna dell'Area 0, l'operatore di rete stampa il link state database di R3 e, relativamente all'Area 0, ottiene il risultato riportato in Figura 1.2. Il candidato:

LS Type	Link ID	Advertising Router
Router	20.64.0.1	20.64.0.1
Router	20.64.0.2	20.64.0.2
Network	20.64.0.2	20.64.0.2

Figura 1.2: Link state database di R3 relativamente all'Area 0

1. specifichi la(le) LIS contenuta(e) nell'Area 0 e i relativi router, indicando, per ciascuno router il *Router ID* e, laddove possibile, l'indirizzo IP dell'interfaccia di collegamento;
2. completi la struttura del link state database di R3 relativamente all'Area 0 e specifichi la struttura del link state database di R3 relativamente all'Area 1;

3. specifichi la struttura dell'LSA che R3 inoltra nell'Area 0 per annunciare la LIS 20.1.2.0/24;
4. specifichi la struttura della *Routing Table* di R3;

Supponiamo adesso che venga fatta l'aggregazione dei prefissi nell'Area 1 e dell'Area 2. Il candidato:

5. specifichi di nuovo la struttura del link state database di R3 relativamente all'Area 0 all'Area 1;
6. risponda alla domanda precedente supponendo che le due aree vengano configurate come *Stub Area*.

NOTA. Il candidato supponga che tutte le LIS (anche quelle contenute nell'Area 0) dell'autonomous system di Figura 1.1 abbiano come prefisso / 24.

RISOLUZIONE

1. Poiché il link state database di R3 relativamente all'Area 0 contiene due *Router LSA* ed un *Network LSA*, dalla struttura degli entries si può dedurre che l'*Area* 0 contiene una sola broadcast LIS con due router (R3 ed R4) ad essa collegati come come illustrato, ad esempio, in Figura 1.3. Dai *Router*

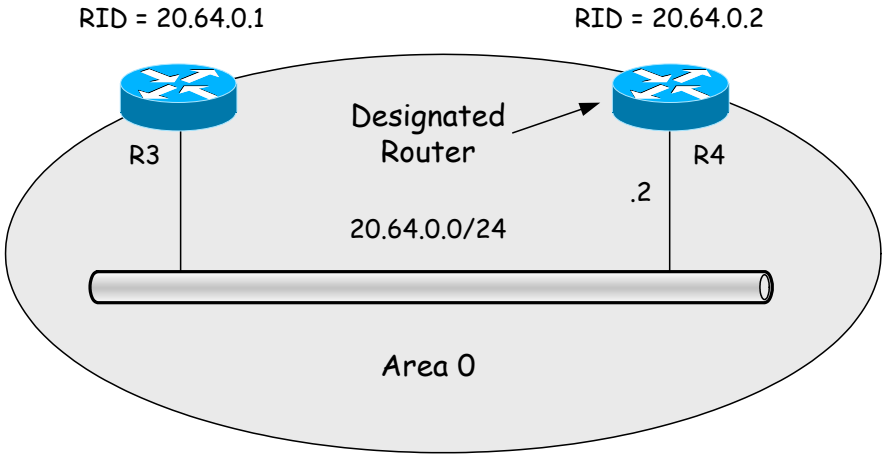


Figura 1.3: Configurazione della LIS contenuta nell'Area 0

LSA si deducono i *Router ID* dei due router che hanno effettuato l'annuncio. Dalle informazioni contenute in Figura 1.2 non è però possibile stabilire una corrispondenza biunivoca tra i due *Router ID* ed R3/R4. Di seguito faremo riferimento alla configurazione illustrata in Figura 1.3. Dal *Network LSA* si

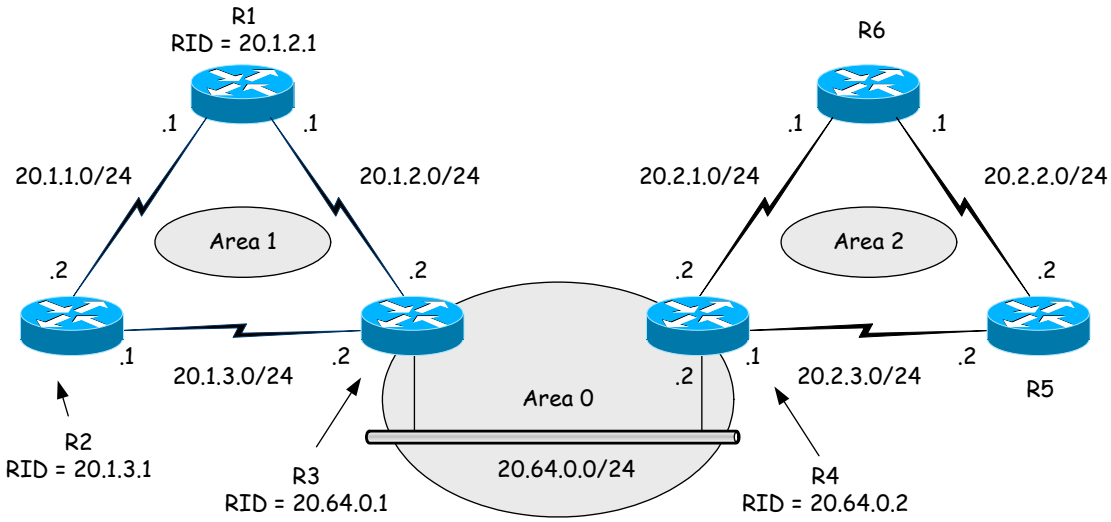


Figura 1.4: Struttura dell'autonomous system

deduce che R4 è il *Designated Router* e che l'IP address della relativa interfaccia è 20.64.0.2. Di conseguenza la configurazione dettagliata dell'autonomous system è illustrata in Figura 1.4.

2. La Figura 1.5 e la Figura 1.6 illustrano il link state database di R3 relati-

LS Type	Link ID	Advertising Router
Router	20.64.0.1	20.64.0.1
Router	20.64.0.2	20.64.0.2
Network	20.64.0.2	20.64.0.2
ABR Summary	20.1.1.0	20.64.0.1
ABR Summary	20.1.2.0	20.64.0.1
ABR Summary	20.1.3.0	20.64.0.1
ABR Summary	20.2.1.0	20.64.0.2
ABR Summary	20.2.2.0	20.64.0.2
ABR Summary	20.2.3.0	20.64.0.2

Figura 1.5: Link state database di R3 relativamente all'Area 0

vamente all'Area 0 e all'Area 1 rispettivamente.

LS Type	Link ID	Advertising Router
Router	20.1.2.1	20.1.2.1
Router	20.1.3.1	20.1.3.1
Router	20.64.0.1	20.64.0.1
ABR Summay	20.2.1.0	20.64.0.1
ABR Summay	20.2.2.0	20.64.0.1
ABR Summay	20.2.3.0	20.64.0.1
ABR Summay	20.64.0.0	20.64.0.1

Figura 1.6: Link state database di R3 relativamente all'Area 1

3. L'LSA richiesto dall'esercizio è un *Router LSA* la cui struttura è illustrata in Figura 1.7.

4. La struttura della *Routing Table* di R3 è riportata in Figura 1.8.

5. La Figura 1.9 e la Figura 1.10 illustrano la struttura del link state database di R3 relativamente all'*Area* 0 all'*Area* 1 rispettivamente.

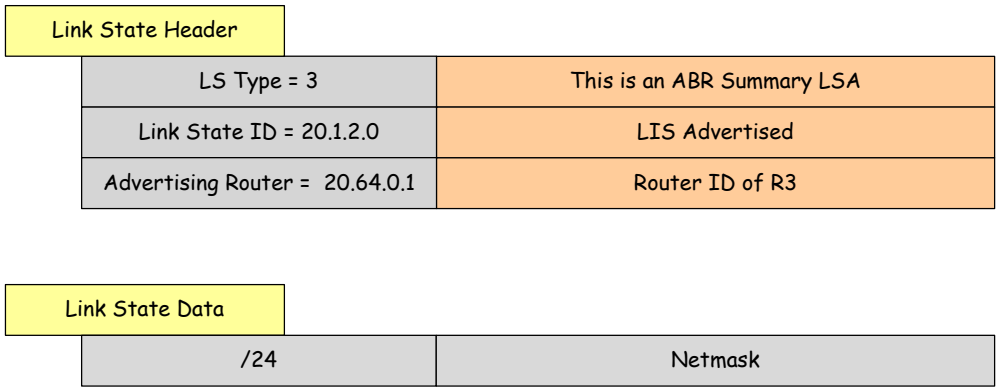


Figura 1.7: Router LSA generato da R3 nell'Area 1 per annunciare il link R1/R3

Network Prefix	Next Hop IP Address
20.1.1.0/24	via 20.1.3.1
20.1.1.0/24	via 20.1.2.1
20.1.2.0/24	directly connected
20.1.3.0/24	directly connected
20.64.0.0/24	directly connected
20.2.1.0/24	via 20.64.0.2
20.2.2.0/24	via 20.64.0.2
20.2.3.0/24	via 20.64.0.2

Figura 1.8: Routing Table di R3

LS Type	Link ID	Advertising Router
Router	20.64.0.1	20.64.0.1
Router	20.64.0.2	20.64.0.2
Network	20.64.0.2	20.64.0.2
ABR Summary	20.1.0.0	20.64.0.1
ABR Summary	20.2.0.0	20.64.0.2

Figura 1.9: Link state database di R3 relativamente all'Area 0 quando vi è aggregazione

6. I link state database di R3 relativamente all'*Area* 0 e all'*Area* 1 non cambiano rispetto a quelli visti nel punto precedente.

LS Type	Link ID	Advertising Router
Router	20.1.2.1	20.1.2.1
Router	20.1.3.1	20.1.3.1
Router	20.64.0.1	20.64.0.1
ABR Summay	20.2.0.0	20.64.0.1
ABR Summay	20.64.0.0	20.64.0.1

Figura 1.10: Link state database di R3 relativamente all'Area 1 quando vi è aggregazione

ESERCIZIO 2: Si considerino due applicazioni in comunicazione tramite una rete di calcolatori. Sia il *Round Trip Time* (RTT) tra i due *host* delle applicazioni costante nel tempo e pari a 1 time slot. Si supponga che le applicazioni utilizzino il protocollo di trasporto TCP, nella versione *Reno*, la quale impiega i meccanismi di *Slow start*, *Fast retransmit* e *Fast recovery* per il controllo della congestione. Il timeout sia pari a 2 time slots. Si considerino il tempo di trasmissione dei segmenti, il tempo di trasmissione dei messaggi di riscontro e il tempo di elaborazione del *TCP receiver* e del *TCP sender* trascurabili rispetto al RTT. Per semplicita` si suppongano i segmenti numerati a partire da 1. Siano, inizialmente, la *congestion window* (CW) pari ad 1 e la *congestion threshold* (CT) pari a 4. La *Advertised Window* del ricevitore abbia dimensione infinita.

Il candidato illustri tramite un diagramma temporale le fasi di invio dei segmenti e di ricezione dei messaggi di riscontro per i segmenti almeno fino a 14. In particolare, si evidenzino i valori della CW e della CT. Si supponga che il *TCP receiver* riceva correttamente tutti i segmenti tranne i seguenti: 8, 9, 13, e 14. Le ritrasmissioni di tali segmenti vengano ricevute correttamente. Il candidato consideri **separatamente** i seguenti due casi:

1. Il *TCP receiver* scarta tutti i segmenti ricevuti fuori ordine.
2. Il *TCP receiver* memorizza in un *reorder buffer* i segmenti ricevuti fuori ordine.

Altre ipotesi:

- l'applicazione mittente generi un numero di segmenti sufficiente a inviare ad ogni istante il massimo numero di segmenti ammessi dal livello trasporto TCP sottostante;
- Il *TCP receiver* invii un riscontro immediato (ACK) per ciascun segmento ricevuto correttamente e in ordine, e un riscontro duplicato (DUP-ACK) immediato per ciascun segmento ricevuto correttamente ma fuori ordine.
- in caso scatti il timeout di ritransmissione di un segmento nello stesso istante in cui il *TCP sender* incrementa la CW, le operazioni conseguenti al timeout vengono effettuate **dopo** avere incrementato opportunamente la CW; inoltre, l'approssimazione della CW avvenga sempre per difetto, a meno che tale operazione comporti l'impostazione di una CW nulla, nel quale caso si assuma CW = 1.

RISOLUZIONE

1. La Figura 2.1 illustra la dinamica del sistema se il *TCP receiver* scarta tutti i segmenti ricevuti fuori ordine. In particolare, all'istante t = 4 il *TCP sender* rileva congestione in seguito alla ricezione di tre riscontri duplicati consecutivi. In base al meccanismo di *Fast recovery*, il *TCP sender* imposta la CW e la CT al valore della CW precedente alla congestione, e entra in una fase di incremento lineare della CW. Invece, all'istante t = 7, il *TCP sender* non rileva congestione in quando il numero di riscontri duplicati, cioe` due, non e` sufficiente a far scattare il *Fast retransmit*. La congestione viene quindi rilevata in base al timeout del segmento 13 all'istante t = 8, successivamente al quale il *TCP sender* entra nella fase di *Slow start*.

2. La Figura 2.2 illustra la dinamica del sistema se il *TCP receiver* memorizza i segmenti ricevuti fuori ordine in un *reorder buffer*. In tale caso, la ricezione del segmento 9 da parte del *TCP receiver* determina l'invio del riscontro non duplicato A13, in quanto i segmenti 10, 11, e 12 sono stati correttamente ricevuti e memorizzati precedentemente. Un ragionamento analogo applica al segmento 14, ricevuto il quale il *TCP receiver* invia direttamente un riscrontro non duplicato A16. Nel caso venga inviato anche il segmento 15, esso e` interpretato dal *TCP receiver* come duplicato.

Figura 2.1:

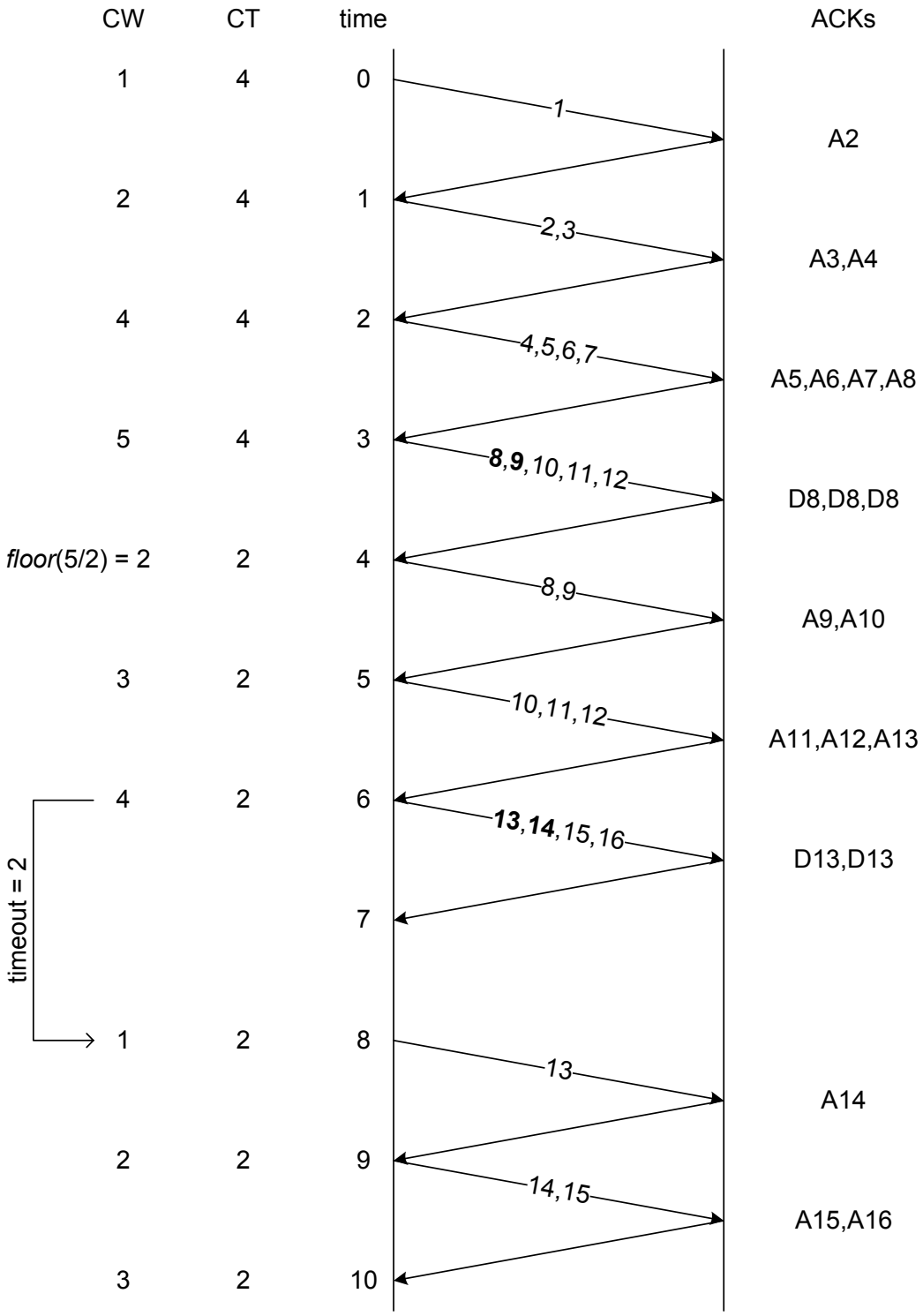
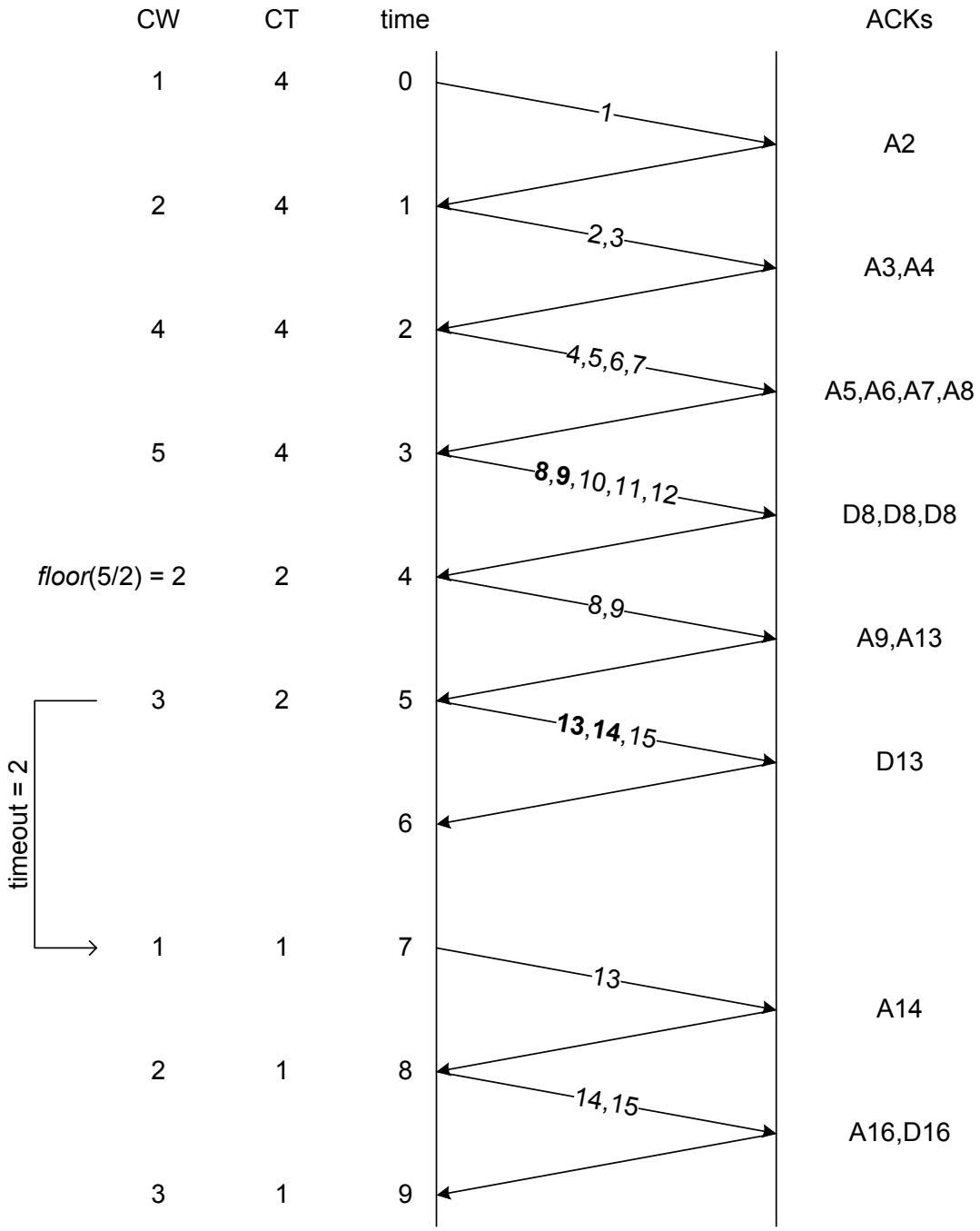

CW
CT
time
ACKs
1
4
0
1
A2
2
4
1
2,3
A3,A4
4
4
2
4,5,6,7
A5,A6,A7,A8
5
4
3
8,9,10,11,12
D8,D8,D8
floor(5/2) = 2
2
4
8,9
A9,A10
3
2
5
10,11,12
A11,A12,A13
4
2
6
13,14,15,16
D13,D13
7
timeout = 2
1
2
8
13
A14
2
2
9
14,15
A15,A16
3
2
10

Figura 2.2:


CW
CT
time
ACKs
1
4
0
1
A2
2
4
1
2,3
A3,A4
4
4
2
4,5,6,7
A5,A6,A7,A8
5
4
3
8,9,10,11,12
D8,D8,D8
floor(5/2) = 2
2
4
8,9
A9,A13
3
2
5
13,14,15
D13
timeout = 2
6
1
1
7
13
A14
2
1
8
14,15
A16,D16
3
1
9