

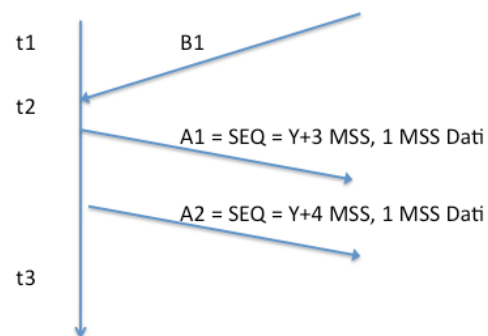
TELEMATICA
PROVA SCRITTA
4 Febbraio 2013
Tempo a disposizione 120 minuti

Es1. (9 pt) (a) Indicare quale è il significato dei campi di un record DNS del tipo (*name,value,NS,...*). (b) Consideriamo uno scenario in cui il server dns locale D, dns.dip.univ.edu, debba risolvere il nome host21.cs.yale.edu. Supponiamo che D contenga il record DNS (host21.cs.yale.edu, dns.yale.edu, NS...). Descrivere la sequenza di messaggi scambiati tra i server DNS coinvolti nella risoluzione dell'interrogazione "iterativa" del nome host21.cs.yale.edu da parte del dns D.

Es.2 (9 pt)

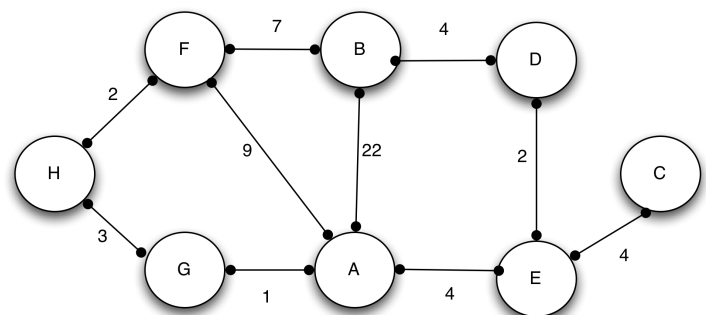
Consideriamo un'applicazione A che ha già stabilito una connessione TCP con un suo pari. Supponiamo che al momento t_1 il valore della finestra di congestione **CW** dell'host di A sia pari a $3MSS$ e gli indici **base** e **next** valgano rispettivamente Y e $Y+3MSS$. Considerare lo scenario in figura. Specificare i possibili valori dei campi **ACK** e **RcvWin** contenuti nel segmento B1 ricevuto dall'host di A, e i valori delle variabili **CW**, **base** e **next** al tempo t_3 . Supponiamo che quando riceve il segmento dal suo pari, l'host di A debba ancora spedire altri 3 MSS di dati per conto di A. Supponiamo infine che al tempo t_1 TCP si trovi nello stato di

congestion avoidance e che il TCP di A non spedisca ulteriori segmenti (fino al verificarsi di nuovi eventi)



Es3. (4) Indicare, giustificando la risposta, se un server Web S è in grado con il meccanismo dei cookies di riconoscere sempre un utente tutte le volte che si collega al server.

Es4 (10 pt) Supponiamo che il sistema autonomo descritto dal grafo a lato utilizzi un algoritmo di tipo link-state per la gestione delle tabelle di routing. (a) Determinare la tabella di routing del nodo A. (b) Utilizzando le informazioni ottenute durante l'applicazione dell'algoritmo link state, determinare il valore di centralità (closeness) del nodo A.



Traccia Soluzione

Es1

- a) *value* è il nome di un server di competenza per *name*
b) Consideriamo i seguenti server DNS con i relativi indirizzi IP

Nome DNS	Indirizzo IP
D _{EDU}	I _{EDU}
D _{CS} = dns.cs.yale.edu	I _{CS}
D _{YALE} = dns.yale.edu	I _{YALE}

Supponiamo per semplicità che l'indirizzo IP del TLD di edu sia noto.

1. D $\rightarrow$ D_{EDU} (dns.yale.edu, A, ...) query per risolvere indirizzo del dns D_{YALE}
2. D_{EDU} $\rightarrow$ D (dns.yale.edu, A, I_{YALE}) risoluzione indirizzo del dns D_{YALE}
3. D_{EDU} $\rightarrow$ D_{YALE} (host21.cs.yale.edu, A, ...) query per risoluzione indirizzo host

A questo punto abbiamo due possibilità

- i) D_{YALE} $\rightarrow$ D (host21.cs.yale.edu, A, I) risoluzione indirizzo host
- ii) D_{YALE} $\rightarrow$ D (host21.cs.yale.edu, NS, dns.cs.yale.edu) risoluzione indirizzo host

Nel caso (i) abbiamo risolto l'indirizzo. Nel caso (ii) si procede come (1,2)

Es2

Dato che l'host di A spedisce un nuovo segmento di dati, il segmento B1 che ha ricevuto deve contenere un riscontro positivo e pertanto al tempo t2 **CW = CW + 1/3 MSS (=10/3 MSS)** dato che il TCP si trova nello stato di congestion avoidance. Dato che l'host aveva tre segmenti "in volo" dobbiamo considerare tre casi.

1a) Notiamo che l'host di A **non** può avere ricevuto il riscontro positivo del segmento più vecchio ancora in volo. Infatti, dopo avere ricevuto tale riscontro, avrebbe avuto ancora 2MSS di dati in volo, e non avrebbe quindi potuto spedire 2 nuovi MSS di dati.

b) Se l'host di A ha ricevuto il riscontro positivo del secondo segmento più vecchio ancora in volo, i valori contenuti nel segmento ricevuto saranno:

ACK = Y + 2 MSS

RcvWin $\geq$ 3 MSS

(infatti se RcvWin < 3 MSS TCP non potrebbe spedire 2 MSS di nuovi dati). Inoltre al tempo t1 avremo:

base = Y + 2 MSS

next = Y + 5 MSS

c) Se l'host di A ha ricevuto il riscontro positivo del segmento più giovane ancora in volo, i valori contenuti nel segmento ricevuto saranno:

ACK = Y + 3 MSS

RcvWin = 2 MSS

Inoltre al tempo t1 avremo:

base = Y + 3 MSS next = Y + 5 MSS

Es3. In generale non è possibile. I cookies (lato utente) sono memorizzati in un file del browser. L'utente potrebbe collegarsi con browser differenti. Inolte l'utente potrebbe cancellare il file dei cookies.

Es4.

Utilizziamo l'algoritmo di Dijkstra con una notazione compatta. Con la notazione A(c,B) indichiamo che c è il costo del collegamento (B,A) e che B è il predecessore di p(A). Inoltre non esplicitiamo i costi infiniti di inizializzazione.

Passo	Nodi Visitati	Nodi con distanza e predecessore, ordinati in base al costo
0	A	G(1,A), E(4, A), F(9,A), B(22,A)
1	A, G(1,A)	H(4,G), E(4, A), F(9,A), B(22,A)
2	A,G, H(4,G)	E(4, A), F(6,H), B(22,A)
3	A,G,H, E(4,A)	D(6,E), F(6,H), C(8,E), B(22,A)
4	A, G, H, E, D(6, E)	F(6,H), C(8,E), B(10,D)
5	A, G, H, E, D, F(6,H)	C(8,E), B(10,D)
6	A, G, H, E, D, F, C(8, E)	B(10,D)
7	A, G, H, E, D, F, C, B(10, D)	-

Per determinare la tabella di routing utilizziamo le informazioni costruite dall'algoritmo relativamente al predecessore. Ad esempio, il cammino minimo per andare da A a B ha costo 10 e come primo hop passa per E dato che $p(B) = D$, $p(D) = E$, $P(E) = A$.

Pertanto la tabella di routing risulta

B	E	10
C	E	8
D	E	6
E	E	4
F	G	6
G	G	1
H	G	4

Utilizzando le informazioni ottenute sui cammini minimi si determina facilmente la closeness (come distanza ottimale media).