Esercitazione

Introduzione
[Capitolo 1]

– 1 –

- Quale vantaggio presenta una rete a commutazione di circuito rispetto a una a commutazione di pacchetto?
- Una rete a commutazione di circuito può garantire una certa quantità di banda end-to-end per la durata della connessione. La maggiorparte delle reti a commutazione di pacchetto (incluso Internet) non posso fornire alcun tipo di garanzie simili

– 2 –

- Supponete di avere un solo commutatore di pacchetto tra un host di invio e uno di ricezione
- La frequenza di trasmissione tra l'host d'invio e il commutatore e tra il commutatore e l'host di ricezione sono ripsettivamente R1 e R2
- Nell'ipotesi che il commutatore adotti la commutazione di pacchetto ***store-and-forward***, qual è il ritardo totale da un capo all'altro (end-to-end) per inviare un pacchetto di lunghezza L?
  - Si ignorino i ritardi di accodamento, propagazione e elaborazione

– 2 –

- Il mittente completa la trasmissione al tempo....
  - t1=L/R1
- Quando il commutatore riceve il pacchetto (nessun ritardo di propagazione)?
  - Al tempo t1
- Il commutatore completa la trasmissione al tempo....
  - t2=t1+L/R2
- Quindi, il ritardo totale end-to-end è
  - L/R1+L/R2

– 3 –

- Quali sono le differenze principali tra ISP di livello 1 e 2?
- Un ISP a livello 1 è connesso a tutti gli altri ISP di livello 1
- Un ISP a livello 2 è connesso solo ad alcuni ISP di livello 1. Inoltre, un ISP di livello 2 è un cliente di uno o più ISP di livello 1
- Gli ISP di livello 1 sono chiamati anche....
  - Dorsali

– 4.1 –

Throughput: frequenza (bit/unità di tempo) alla quale i bit sono ricevuti dal destinatario

- Supponete che l'host A voglia inviare un file voluminoso all'host B
- Il cammino tra A e B ha tre collegamenti con frequenze
  - R1=500kbps, R2=2Mbps e R3=1Mbps
- Assumete che non vi sia altro traffico nella rete e altri tipi di ritardo: qual è il throughput per il trasferimento del file?
  - 500kbps

– 4.2 –

- Supponete che l'host A voglia inviare un file voluminoso all'host B
- Il cammino tra A e B ha tre collegamenti con frequenze
  - R1=500kbps, R2=2Mbps e R3=1Mbps
- Supponete ora che il file sia di 4 milioni di byte. Approssimativamente, quanto tempo occorrerà per trasferire il file all'host B?
  - 64 secondi

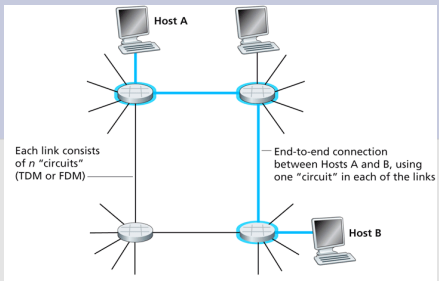
– 5 –

- Considerate un'applicazione che trasmette dati a frequenza costante. Inoltre, una volta avviata, l'applicazione continuerà a funzionare per un periodo di tempo relativamente lungo
- Per questa applicazione sarebbe più appropriata una rete a commutazione di pacchetto o a commutazione di circuito?
- Perché?

– 5 –

- Rete a commutazione di circuito
- Infatti, l'applicazione prevede lunghe sessioni con richieste di banda costante
- Inoltre, dato che la frequenza di trasmissione è nota e non ha "picchi", la banda può essere riservata per ogni sessione senza sprechi significativi
- Infine, i costi per stabilire il circuito sono ammortizzati dalla durata delle sessioni

– 6 –



- Considerate la rete a commutazione di circuito in figura: ci sono n circuiti su ogni collegamento
- Qual è il massimo numero di connessioni contemporaneamente attive in un dato istante in questa rete?
  - 4n

– 7.1 –

- Consideriamo due host, A e B, collegati da una singola connessione con velocità di Rbps. Supponiamo che i due host siano separati da *m* metri, e che la velocità di propagazione lungo il collegamento sia di *s* m/sec. A sta per inviare un pacchetto di L bit a B
- Esprimete il ritardo di propagazione in funzione di *m* e *s*
  - $d_{prop} = m/s$ secondi

– 7.2 –

- Consideriamo due host, A e B, collegati da una singola connessione con velocità di Rbps. Supponiamo che i due host siano separati da *m* metri, e che la velocità di propagazione lungo il collegamento sia di *s* m/sec. A sta per inviare un pacchetto di L bit a B
- Determinate il tempo di trasmissione del pacchetto
  - $d_{trasm} = L/R$ secondi

– 7.3 –

- Consideriamo due host, A e B, collegati da una singola connessione con velocità di Rbps. Supponiamo che i due host siano separati da *m* metri, e che la velocità di propagazione lungo il collegamento sia di *s* m/sec. A sta per inviare un pacchetto di L bit a B
- Tralasciando i ritardi di elaborazione e di accodamento, ricavate un'espressione del ritardo end-to-end
  - $d_{end\text{-}to\text{-}end} = (m/s + L/R)$ secondi

– 7.4 –

- Consideriamo due host, A e B, collegati da una singola connessione con velocità di Rbps. Supponiamo che i due host siano separati da *m* metri, e che la velocità di propagazione lungo il collegamento sia di *s* m/sec. A sta per inviare un pacchetto di L bit a B
- Supponiamo che l'host A cominci a trasmettere il pacchetto all'istante t=0. All'istante $t=d_{trasm}$ dove si trova l'ultimo bit del pacchetto?
  - L'ultimo bit sta appena lasciando A

– 7.5 –

- Consideriamo due host, A e B, collegati da una singola connessione con velocità di Rbps. Supponiamo che i due host siano separati da *m* metri, e che la velocità di propagazione lungo il collegamento sia di *s* m/sec. A sta per inviare un pacchetto di L bit a B
- Supponiamo che $d_{prop} > d_{trasm}$. All'istante $t=d_{trasm}$ dove si trova il primo bit del pacchetto?
  - Il primo bit è sul collegamento e non ha ancora raggiunto B

– 7.6 –

- Consideriamo due host, A e B, collegati da una singola connessione con velocità di Rbps. Supponiamo che i due host siano separati da *m* metri, e che la velocità di propagazione lungo il collegamento sia di *s* m/sec. A sta per inviare un pacchetto di L bit a B
- Supponiamo che $d_{prop} < d_{trasm}$. All'istante $t=d_{trasm}$ dove si trova il primo bit del pacchetto?
  - Il primo bit ha raggiunto B

– 8 –

- Considerate un pacchetto di lunghezza L che viene creato su A, viaggia su un collegamento verso un commutatore di pacchetto, e da questi si propaga su un secondo collegamento verso B
- Siano d_i, s_i e R_i la lunghezza, la velocità di propagazione e la frequenza di trasmissione del collegamento i (i=1, 2). Il commutatore, inolre, ritarda ciascun pacchetto di d_{proc}
- Assumendo che non ci siano altri ritardi, qual è il ritardo totale end-to-end per il pacchetto?
  - $L/R1+d1/s1+d_{proc}+L/R2+d2/s2$

– 9 –

- Stesse ipotesi del problema precedente, dove
  - R1=R2=R e $d_{proc}=0$
  - Il commutatore di pacchetto non utilizza store-and-forward (a proposito, cos'è?)
    - Trasmette immediatamente i bit che riceve
- Qual è ora il ritardo totale end-to-end per il pacchetto?
  - $L/R+d1/s1+d2/s2$

– 10 –

- Supponiamo che un pacchetto arrivi a un commutatore di pacchetto al tempo t. In questo istante
  - È in corso la trasmissione di un altro pacchetto, che è stato già trasmesso per metà
  - Altri 3 pacchetti sono in coda per la trasmissione (in ordine d'arrivo)
- Supponete che tutti i pacchetti siano di 1000 byte e la frequenza del collegamento sia di 1 Mbps
- Qual è il ritardo di accodamento per il pacchetto (cos'è)?
  - Il pacchetto deve attendere la trasmissione di 3500 byte = 28000 bits, quindi 28 msec

– 11 –

- Supponete che N pacchetti arrivino contemporaneamente a un collegamento nel quale non vi sono pacchetti in corso di trasmissione o in coda
- Ciascun pacchetto è di lunghezza L e la frequenza di trasmissione è R
- Qual è il ritardo medio di accodamento per gli N pacchetti?
  - $[0+L/R+2(L/R)+3(L/R)....+(N-1)(L/R)]/N=$ $(L/RN)(1+2+....+N-1)=[L/RN][N(N-1)/2]=$ $L(N-1)/2R$

Esercitazione

Livello di Applicazione
[Capitolo 2]

– 12 –

- Supponete di voler fare una transazione da un client remoto a un server il più velocemente possibile
- Userete TCP o UDP? Perché? Motivare la risposta in termini di RTT necessari
  - UDP: con UDP la transazione può essere completata in un RTT (richiesta+risposta)
  - TCP: con TCP un minimo di 2 RTT sono necessari (uno per stabilire la connessione e un altro per la richiesta effettiva)

– 13 –

- Cosa significa “un protocollo usa handshaking”?
  - Un protocollo usa handshaking se le due parti si scambiano prima dei pacchetti di controllo prima di inviare i dati
  - SMTP usa handshaking a livello applicazione?
    - SI
  - HTTP (a livello applicazione)?
    - NO

– 14 –

- Supponete che Alice, con un account di posta elettronica basato sul WEB (e.g., gmail), invii un messaggio a Roberto che accede alla propria casella usando POP3
- Descrivete come il messaggio giugne dall'host di Alice a quello di Roberto, elencando i protocolli a livello di applicazione usati per trasferire il messaggio tra i due host
  - Il messaggio è inviato dall'host di Alice al suo mail server (HTTP)
  - Il mail server di Alice invia il messaggio al mail server di Roberto (SMTP)
  - Roberto trasferisce il messaggio dal suo mail server al suo host (POP3)

– 15 –

- Dal punto di vista dell'utente, qual è la differenza tra la modalità scarica-e-cancella e scarica-e-mantieni in POP3?
  - Scarica-e-cancella: dopo che l'utente ha scaricato il messaggio dal server POP, il messaggio è cancellato
  - Scarica-e-mantieni: il messaggio non viene cancellato dopo averlo scaricato
    - Quindi, ogni volta che l'utente scarica la sua posta elettronica da una macchina nuova, ***TUTTI*** i messaggi sul server vengono scaricati (anche quelli già letti precedentemente da qualche altra macchina)

– 16 –

- Considerate un nuovo peer, Alice, che entra a far parte di un torrent (BitTorrent) senza aver nessuna parte del file
- Senza parti, non può diventare uno dei quattro uploader principali di qualcuno degli altri peer, in quanto non ha nulla da inviare
- Come fa Alice a ottenere la sua prima parte del file?
  - Alice ottiene il suo primo pezzo del file quando viene selezionata casualmente da uno dei suoi vicini
  - Ogni quanto tempo un peer effettua questa selezione casuale di un vicino?
    - Ogni 30 secondi

– 17.1 –

- In cosa consiste una rete di copertura (overlay network) in un sistema di condivisione file P2P?
  - Una rete di copertura in un systema di file sharing P2P consiste dei nodi che partecipano al sistema e dei collegamenti logici tra i nodi
- Come si formano i collegamenti logici della rete di copertura?
  - C'è un collegamento logico fra A e B se esiste una connessione TCP semi-permanente tra A e B
- La rete di copertura include i router?
  - NO

– 17.2 –

- Come viene creata e mantenuta una rete di copertura con “query flooding”?
  - Un peer che vuole unirsi alla rete, contatta peer già presenti (la lista la recupara da siti noti)
  - Tenta di stabilire connessioni TCP con i peer che rispondono
  - I vicini contattati, inoltrano l'informazione sulla nuova connessione utilizzando il flooding
  - I peer contattati dal flooding fanno pervenire al nuovo peer i loro IP (mediante “percorso inverso”)
  - Il nuovo peer tenta di stabilire nuove connessioni con i peer che hanno risposto al flooding

– 18 –

- L'intestazione `Date` nel messaggio di risposta HTTP indica quando è stato modificato per l'ultima volta l'oggetto nella risposta. Vero o Falso?
  - Falso, indica la data di creazione e invio della risposta da parte del server
- Quale intestazione nel messaggio di risposta indica l'ultima modifica all'oggetto?
  - `Last-modified`

– 19.1 –

- Elencate tutti i comandi del cliente che riuscite a trovare nella RFC 959 (FTP)
  - Comandi di accesso
    - `USER, PASS, ACCT, CWD, CDUP, SMNT, REIN, QUIT`
  - Comandi per l'invio di parametri
    - `PORT, PASV, TYPE, STRU, MODE`
  - Comandi di servizio
    - `RETR, STOR, STOU, APPE, ALLO, REST, RNFR, RNTO, ABOR, DELE, RMD, MRD, PWD, LIST, NLST, SITE, SYST, STAT, HELP, NOOP`

– 19.2 –

- Su quale connessione sono trasferiti i file?
  - Connessione dati
- Su quale connessione sono trasferiti i comandi?
  - Connessione di controllo
- C'è qualche comando che viene trasferito sulla connessione dati?
  - NO
- Qual è la dimensione del byte per il trasferimento dei dati?
  - 8 bits

– 19.3 –

- A cosa serve il tipo dati `IMAGE`?
  - Per trasferire dati binari
- A cosa serve il comando di controllo `CWD`?
  - [Change Working Directory] Permette all'utente di cambiare directory
- A cosa serve il comando `TYPE`?
  - Specifica la rappresentazione dei dati da trasferire
- Fate un esempio di `TYPE`?
  - `IMAGE`

– 19.4 –

- Quali sono i comandi per inviare o richiedere il trasferimento di un file?
  - `STOR` e `RETR`
- Con quale comando si richiede l'elenco dei file presenti nella directory passata come argomento?
  - `LIST`
- A cosa serve il comando `NOOP`?
  - A nulla. Causa solo una risposta OK da parte del server

– 19.5 –

- Quale codice di risposta significa “Connessione dati aperta; nessun trasferimento in corso”?
  - `225`
- Quale codice di risposta significa “Connessione dati non possibile”
  - `425`

– 20.1 –

```
Get /cs453/index.html HTTP/1.1<cr><lf> Host:
gaia.cs.umass.edu<cr><lf>User-Agent: Mozilla/5.0
[....]<cr><lf>Accept:ext/xml,application/xml,application/xhtml
+xml,text/html;q=0.9,text/plain;[....]<cr><lf>Accept-Encoding:
zip,deflate<cr><lf>Accept-Charset: ISO-8859-1,utf-
8;q=0.7,*;q=0.7<cr><lf>Keep-Alive: 300<cr><lf>Connection:keep-
alive<cr><lf><cr><lf>
```

- Dato questo frammento di richiesta HTTP, qual è la URL completa del documento richiesto?

  `http://gaia.cs.umass.edu/cs453/index.html`

– 20.2 –

```
Get /cs453/index.html HTTP/1.1<cr><lf> Host:
gaia.cs.umass.edu<cr><lf>User-Agent: Mozilla/5.0
[....]<cr><lf>Accept:ext/xml,application/xml,application/xhtml
+xml,text/html;q=0.9,text/plain;[....]<cr><lf>Accept-Encoding:
zip,deflate<cr><lf>Accept-Charset: ISO-8859-1,utf-
8;q=0.7,*;q=0.7<cr><lf>Keep-Alive: 300<cr><lf>Connection:keep-
alive<cr><lf><cr><lf>
```

- Qual è la versione HTTP utilizzata dal browser?
  - 1.1

– 20.3 –

```
Get /cs453/index.html HTTP/1.1<cr><lf> Host:
gaia.cs.umass.edu<cr><lf>User-Agent: Mozilla/5.0
[....]<cr><lf>Accept:ext/xml,application/xml,application/xhtml
+xml,text/html;q=0.9,text/plain;[....]<cr><lf>Accept-Encoding:
zip,deflate<cr><lf>Accept-Charset: ISO-8859-1,utf-
8;q=0.7,*;q=0.7<cr><lf>Keep-Alive: 300<cr><lf>Connection:keep-
alive<cr><lf><cr><lf>
```

- Il broswer richiede una connessione persistente o non persistente?
  - Persistente

– 20.4 –

```
Get /cs453/index.html HTTP/1.1<cr><lf> Host:
gaia.cs.umass.edu<cr><lf>User-Agent: Mozilla/5.0
[....]<cr><lf>Accept:ext/xml,application/xml,application/xhtml
+xml,text/html;q=0.9,text/plain;[....]<cr><lf>Accept-Encoding:
zip,deflate<cr><lf>Accept-Charset: ISO-8859-1,utf-
8;q=0.7,*;q=0.7<cr><lf>Keep-Alive: 300<cr><lf>Connection:keep-
alive<cr><lf><cr><lf>
```

- Qual è l'indirizzo IP dell'host sul quale è in esecuzione il client (browser)?
  - Non si può determinare!! Come si determina?
    - Bisogna accedere al datagramma IP, che trasporta il segmento TCP, che include la richiesta GET

– 21.1 –

```
HTTP/1.1 200 OK<cr><lf>Date: Tue, 11 Nov 2008
12:39:45GMT<cr><lf>Server: Apache/2.0.52 (Debian)<cr><lf>Last-
Modified: Sat, 10 Dec 2005 18:27:46GMT<cr><lf>ETag: "526c3-
f22-a88a4c80"<cr><lf>Accept-Ranges: bytes<cr><lf>Content-
length: 3874<cr><lf>Keep-Alive:
timeout=max=100<cr><lf>Connection: Keep-Alive<cr><lf>Content-
Type: text/html; charset=ISO-8859-1<cr><lf><cr><lf><!doctype
html public "-//w3c//dtd html 4.0
transitional//en"><lf><html><lf><head><meta http-
equiv="Content-Type" content="text/html"; charset=iso-8859-
1"><lf><meta name="GENERATOR" [....]>
```

- Dato questo frammento di risposta HTTP, è stato capace il server di trovare il documento?
  - SI

– 21.2 –

```
HTTP/1.1 200 OK<cr><lf>Date: Tue, 11 Nov 2008
12:39:45GMT<cr><lf>Server: Apache/2.0.52 (Debian)<cr><lf>Last-
Modified: Sat, 10 Dec 2005 18:27:46GMT<cr><lf>ETag: "526c3-
f22-a88a4c80"<cr><lf>Accept-Ranges: bytes<cr><lf>Content-
length: 3874<cr><lf>Keep-Alive:
timeout=max=100<cr><lf>Connection: Keep-Alive<cr><lf>Content-
Type: text/html; charset=ISO-8859-1<cr><lf><cr><lf><!doctype
html public "-//w3c//dtd html 4.0
transitional//en"><lf><html><lf><head><meta http-
equiv="Content-Type" content="text/html"; charset=iso-8859-
1"><lf><meta name="GENERATOR" [....]>
```

- In quale istante il documento di risposta è stato fornito?
  - 11 Nov 2008 alle 12:39:45 GMT

– 21.3 –

```
HTTP/1.1 200 OK<cr><lf>Date: Tue, 11 Nov 2008
12:39:45GMT<cr><lf>Server: Apache/2.0.52 (Debian)<cr><lf>Last-
Modified: Sat, 10 Dec 2005 18:27:46GMT<cr><lf>ETag: "526c3-
f22-a88a4c80"<cr><lf>Accept-Ranges: bytes<cr><lf>Content-
length: 3874<cr><lf>Keep-Alive:
timeout=max=100<cr><lf>Connection: Keep-Alive<cr><lf>Content-
Type: text/html; charset=ISO-8859-1<cr><lf><cr><lf><!doctype
html public "-//w3c//dtd html 4.0
transitional//en"><lf><html><lf><head><meta http-
equiv="Content-Type" content="text/html"; charset=iso-8859-
1"><lf><meta name="GENERATOR" [....]>
```

- Quando è stato modificato l'ultima volta il documento?
  - 10 Dec 2005 alle 18:27:46 GMT

– 21.4 –

```
HTTP/1.1 200 OK<cr><lf>Date: Tue, 11 Nov 2008
12:39:45GMT<cr><lf>Server: Apache/2.0.52 (Debian)<cr><lf>Last-
Modified: Sat, 10 Dec 2005 18:27:46GMT<cr><lf>ETag: "526c3-
f22-a88a4c80"<cr><lf>Accept-Ranges: bytes<cr><lf>Content-
length: 3874<cr><lf>Keep-Alive:
timeout=max=100<cr><lf>Connection: Keep-Alive<cr><lf>Content-
Type: text/html; charset=ISO-8859-1<cr><lf><cr><lf><!doctype
html public "-//w3c//dtd html 4.0
transitional//en"><lf><html><lf><head><meta http-
equiv="Content-Type" content="text/html"; charset=iso-8859-
1"><lf><meta name="GENERATOR" [....]>
```

- Quanti byte ci sono nel documento inviato al client?
  - 3874

– 21.5 –

```
HTTP/1.1 200 OK<cr><lf>Date: Tue, 11 Nov 2008
12:39:45GMT<cr><lf>Server: Apache/2.0.52 (Debian)<cr><lf>Last-
Modified: Sat, 10 Dec 2005 18:27:46GMT<cr><lf>ETag: "526c3-
f22-a88a4c80"<cr><lf>Accept-Ranges: bytes<cr><lf>Content-
length: 3874<cr><lf>Keep-Alive:
timeout=max=100<cr><lf>Connection: Keep-Alive<cr><lf>Content-
Type: text/html; charset=ISO-8859-1<cr><lf><cr><lf><!doctype
html public "-//w3c//dtd html 4.0
transitional//en"><lf><html><lf><head><meta http-
equiv="Content-Type" content="text/html"; charset=iso-8859-
1"><lf><meta name="GENERATOR" [....]>
```

- Cosa sono i primi 5 byte del documento inviato al client?
  - <!doc

– 21.6 –

```
HTTP/1.1 200 OK<cr><lf>Date: Tue, 11 Nov 2008
12:39:45GMT<cr><lf>Server: Apache/2.0.52 (Debian)<cr><lf>Last-
Modified: Sat, 10 Dec 2005 18:27:46GMT<cr><lf>ETag: "526c3-
f22-a88a4c80"<cr><lf>Accept-Ranges: bytes<cr><lf>Content-
length: 3874<cr><lf>Keep-Alive:
timeout=max=100<cr><lf>Connection: Keep-Alive<cr><lf>Content-
Type: text/html; charset=ISO-8859-1<cr><lf><cr><lf><!doctype
html public "-//w3c//dtd html 4.0
transitional//en"><lf><html><lf><head><meta http-
equiv="Content-Type" content="text/html"; charset=iso-8859-
1"><lf><meta name="GENERATOR" [....]>
```

- Il server ha accettato la connessione persistente?
  - SI

– 22.1 –

- Considerate l'accesso alla vostra posta elettronica tramite POP3, e supponiamo che il client di posta elettronica sia configurato per operare in modalità ***scarica-e-cancella***. Completare la seguente transazione

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: [....]
S: .
?
?
```

– 22.1 –

- Considerate l'accesso alla vostra posta elettronica tramite POP3, e supponiamo che il client di posta elettronica sia configurato per operare in modalità ***scarica-e-cancella***. Completare la seguente transazione

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: [....]
S: .
```

```
C: dele 1
C: retr 2
S: [....]
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```

– 22.2 –

- Considerate l'accesso alla vostra posta elettronica tramite POP3, e supponiamo che il client di posta elettronica sia configurato per operare in modalità ***scarica-e-mantieni***. Completare la seguente transazione

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: [....]
S: .
?
?
```

– 22.2 –

- Considerate l'accesso alla vostra posta elettronica tramite POP3, e supponiamo che il client di posta elettronica sia configurato per operare in modalità ***scarica-e-mantieni***. Completare la seguente transazione

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: [....]
S: .
C: retr 2
S: [....]
S: .
C: quit
S: +OK POP3 server signing off
```

– 22.3 –

- Supponiamo di essere nella modalità scarica-e-mantieni (esercizio precedente), e supponiamo di aver recuperato i messaggi 1 e 2 e di essere usciti dal POP
- 5 minuti più tardi accediamo nuovamente ai messaggi di posta elettronica, e supponiamo che in questo intervallo non sia arrivato alcun nuovo messaggio
- Produrre una trascrizione di questa seconda sessione POP

– 22.3 –

- Produrre una trascrizione di questa seconda sessione POP

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: [....]
S: .
C: retr 2
S: [....]
S: .
C: quit
S: +OK POP3 server signing off
```

– 23 –

- Sistema P2P con query flooding
- Quando un peer ha un file che corrisponde alla richiesta, invia un messaggio query utilizzando la tecnica del ***percorso inverso***
- Descrivete in dettaglio cosa succede....
- Quali sono i vantaggi e gli svantaggi di far stabilire al peer che detiene il file una connessione diretta TCP con il peer che lo ha richiesto?
  - ***Vantaggio***: il query hit è instradato direttamente da Internet e non viaggia attraverso peer intermedi, quindi il ritardo per l'invio del messaggio dovrebbe essere inferiore al ***percorso inverso***
  - ***Svantaggio***: ogni peer che ha il file richiesto cercherebbe di aprire una connessione TCP con il peer che lo ha richiesto (potrebbero essere decine o centinaia per una singola query)

– 24 –

- Supponiamo di cliccare, all'interno di un browser, su un collegamento per ottenere una pagina web
- L'indirizzo IP dell'URL non si trova nella cache dell'host locale → ricerca DNS
- Vengono visitati *n* server DNS, e la visita *i* costa RTTi
- Supponiamo che la pagina web acceduta contenga solo una piccola quantità di HTML. Sia RTT0 l'RTT tra l'host locale e il server che contiene l'oggetto
- Quanto tempo intercorre (in termini di RTT) tra il clic e la ***ricezione dell'oggetto*** presso il client?
  - RTT1+RTT2+....+RTTn+2RTT0

– 25.1 –

- Supponiamo di essere nelle ipotesi del problema precedente
- Inoltre, supponiamo che il file HTML referenzi tre oggetti molto piccoli sullo stesso server
- Ignorando il tempo di trasmissione degli oggetti, quanto tempo intercorre tra il clic e la ***ricezione completa di tutti gli oggetti*** con
- HTTP non persistente e con connessioni TCP in serie?
  - RTT1+RTT2+....+RTTn+2RTT0+3*2RTT0
- Quante connessioni TCP sono aperte con il server?
  - 1+3

– 25.2 –

- Supponiamo di essere nelle ipotesi del problema precedente
- Inoltre, supponiamo che il file HTML referenzi tre oggetti molto piccoli sullo stesso server
- Ignorando il tempo di trasmissione degli oggetti, quanto tempo intercorre tra il clic e la ***ricezione completa di tutti gli oggetti*** con
- HTTP non persistente e connessioni TCP parallele? (grado di parallelismo tipicamente controllabile dal browser)
  - RTT1+RTT2+....+RTTn+2RTT0+2RTT0
- Quante connessioni TCP sono aperte con il server?
  - 1+3

– 25.3 –

- Supponiamo di essere nelle ipotesi del problema precedente
- Inoltre, supponiamo che il file HTML referenzi tre oggetti molto piccoli sullo stesso server
- Ignorando il tempo di trasmissione degli oggetti, quanto tempo intercorre tra il clic e la ***ricezione completa di tutti gli oggetti*** con
- HTTP persistente e senza pipeline?
  - RTT1+RTT2+....+RTTn+2RTT0+3RTT0
- Quante connessioni TCP sono aperte con il server?
  - 1

– 25.4 –

- Supponiamo di essere nelle ipotesi del problema precedente
- Inoltre, supponiamo che il file HTML referenzi tre oggetti molto piccoli sullo stesso server
- Ignorando il tempo di trasmissione degli oggetti, quanto tempo intercorre tra il clic e la ***ricezione completa di tutti gli oggetti*** con
- HTTP persistente e con pipeline?
  - RTT1+RTT2+....+RTTn+2RTT0+RTT0
- Quante connessioni TCP sono aperte con il server?
  - 1