

Università di Roma Tor Vergata
Corso di Laurea triennale in Informatica
Sistemi operativi e reti
A.A. 2017-18

Pietro Frasca

Parte II: Reti di calcolatori
Lezione 8 (32)

Giovedì 29-03-2018

Connessione non persistente e persistente

- Il TCP offre alle applicazioni un servizio di trasferimento di dati affidabile. Il client richiede la connessione al server, dopo la fase di handshake il TCP crea una connessione punto-punto tra le due estremità. Le due parti possono scambiarsi i dati contemporaneamente nelle due direzioni, fino a quando una delle due estremità chiude la connessione.
- L'HTTP usa la connessione TCP in due modalità dette connessione **persistente** e **non persistente**.
- La versione HTTP/1.1 e HTTP//2 usano le **connessioni persistenti** nella modalità di default. Le connessioni persistenti, consentono di trasferire più file con un'unica connessione TCP.
- La vecchia versione HTTP/1.0 invece usa la connessione non persistente che consente di trasferire un singolo file per ogni connessione TCP. Tuttavia, è possibile configurare sia i client che i server HTTP/1.1 per utilizzare le connessioni non persistenti.

Connessione non persistente

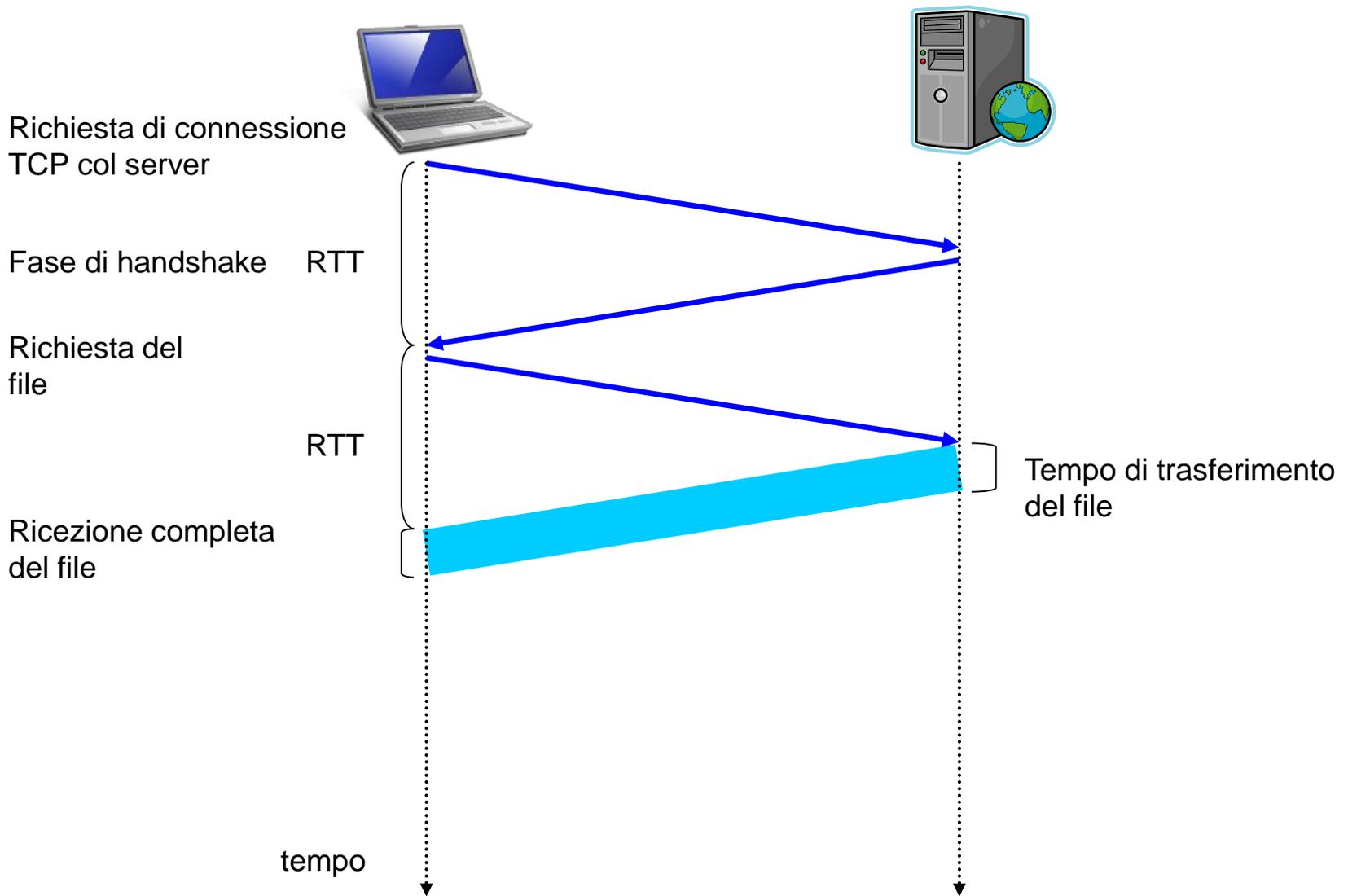
Descriviamo con un esempio come avviene il trasferimento di file con la connessione **non persistente**. Supponiamo che un client richieda una pagina formata da un file base HTML e da 10 immagini JPEG, per un totale di 11 file. La transazione HTTP avviene nel seguente modo:

1. Il browser instaura una connessione TCP col server (ad esempio www.pf.uniroma2.it) alla porta **80**.
2. Il browser invia un messaggio di richiesta HTTP.
3. Il server HTTP riceve il messaggio di richiesta, cerca il file richiesto e se lo trova, lo inserisce in un messaggio di risposta HTTP che invia al client.
4. Il server web **chiude la connessione TCP** instaurata con il client.
5. Il client riceve il messaggio di risposta e riconosce che il file ricevuto è un file HTML, quindi lo interpreta e trova i riferimenti ai 10 oggetti JPEG. La connessione TCP si chiude.
6. La sequenza relativa ai passi 1-4 viene ripetuta per ciascuno dei file JPEG.

Listato del file html dell'esempio:

```
<h1> Esempio di connessione <b>non persistente</b></h1>  
   
   
   
   
 
```

- Notiamo che ciascuna connessione TCP trasporta un messaggio di richiesta e un messaggio di risposta. Quindi, in questo esempio, quando un utente richiede una pagina, si generano **11 connessioni TCP**.
- Il browser (client) può ottenere le 10 immagini JPEG su 10 connessioni TCP in serie, o su più connessioni TCP in parallelo (fino ad un numero massimo prestabilito). Generalmente i browser possono essere configurati con un numero massimo di connessioni parallele, tipicamente 5-10.
- L'uso di connessioni in parallelo diminuisce i tempi di risposta.



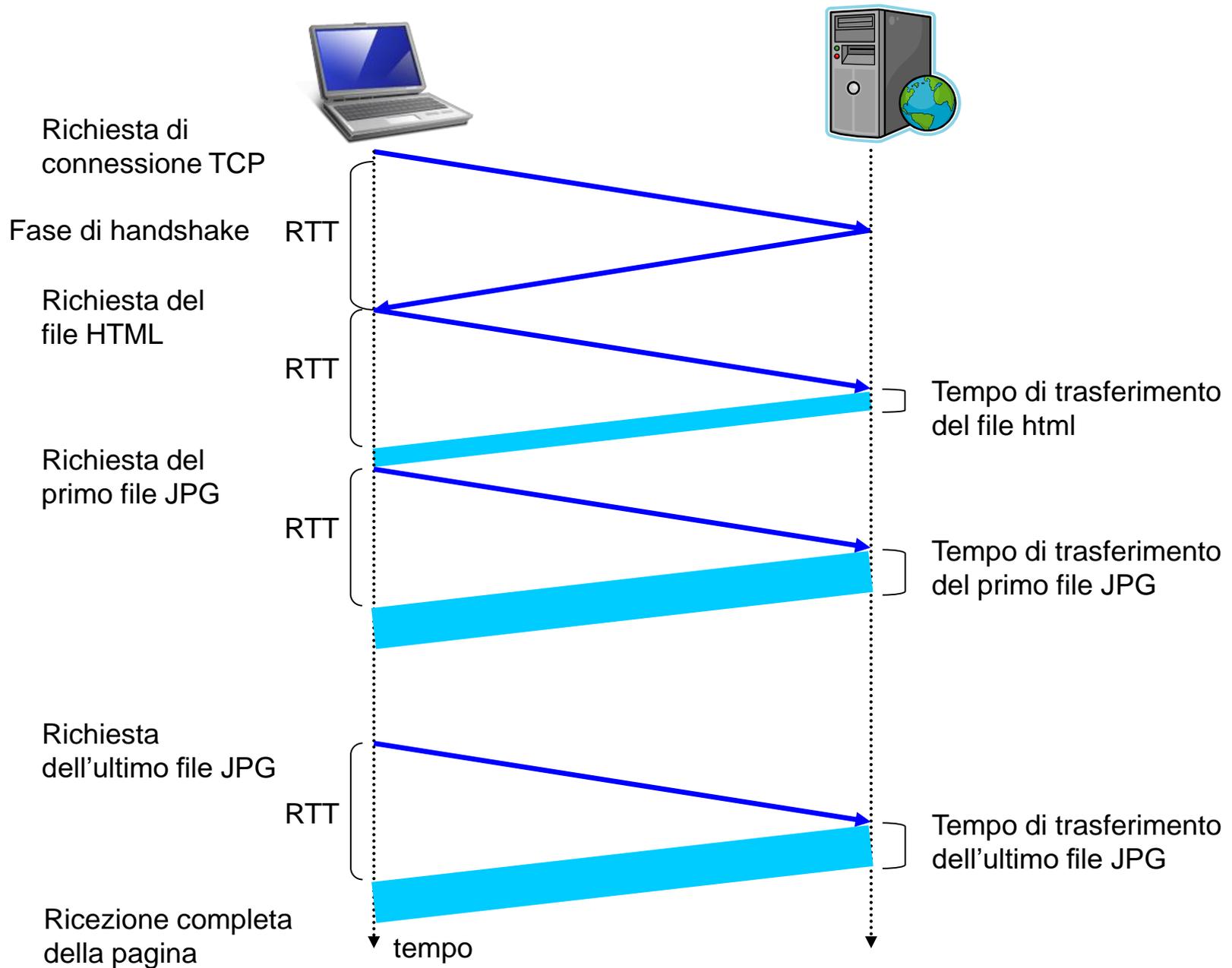
- Il server, quando riceve il messaggio di richiesta, invia il file HTML mediante il TCP. Il tempo che passa dall'inizio della richiesta all'ottenimento della risposta HTTP è di circa due **RTT** più il tempo di trasferimento del file HTML.
- Con **RTT (Round Trip Time, Tempo di andata e ritorno)** si indica l'intervallo di tempo che passa da quando un piccolo messaggio (come ad esempio il messaggio di richiesta di connessione), parte dal client, arriva al server e ritorna al client.
- Le connessioni non persistenti hanno alcuni svantaggi.
 - per ogni file richiesto è necessario instaurare una nuova connessione TCP che richiede buffer e variabili sia sul client che sul server. Se il server web deve soddisfare richieste di centinaia di client è evidente che risulterà fortemente sovraccarico;
 - prima che inizi il trasferimento di ogni file devono trascorrere due RTT.

Connessione persistente

- La connessione persistente, consente di trasferire più file con un'unica connessione TCP. Ad esempio, la pagina Web dell'esempio precedente, costituita da un file html e da 10 immagini jpeg, può essere inviata su un'unica connessione TCP persistente.
- La chiusura della connessione avviene quando il server HTTP rileva che la connessione stessa non è usata da un certo tempo, detto **intervallo di timeout**.
- Esistono due modalità di funzionamento della connessione persistente: **senza parallelismo** e **con parallelismo**.

Connessione persistente senza parallelismo

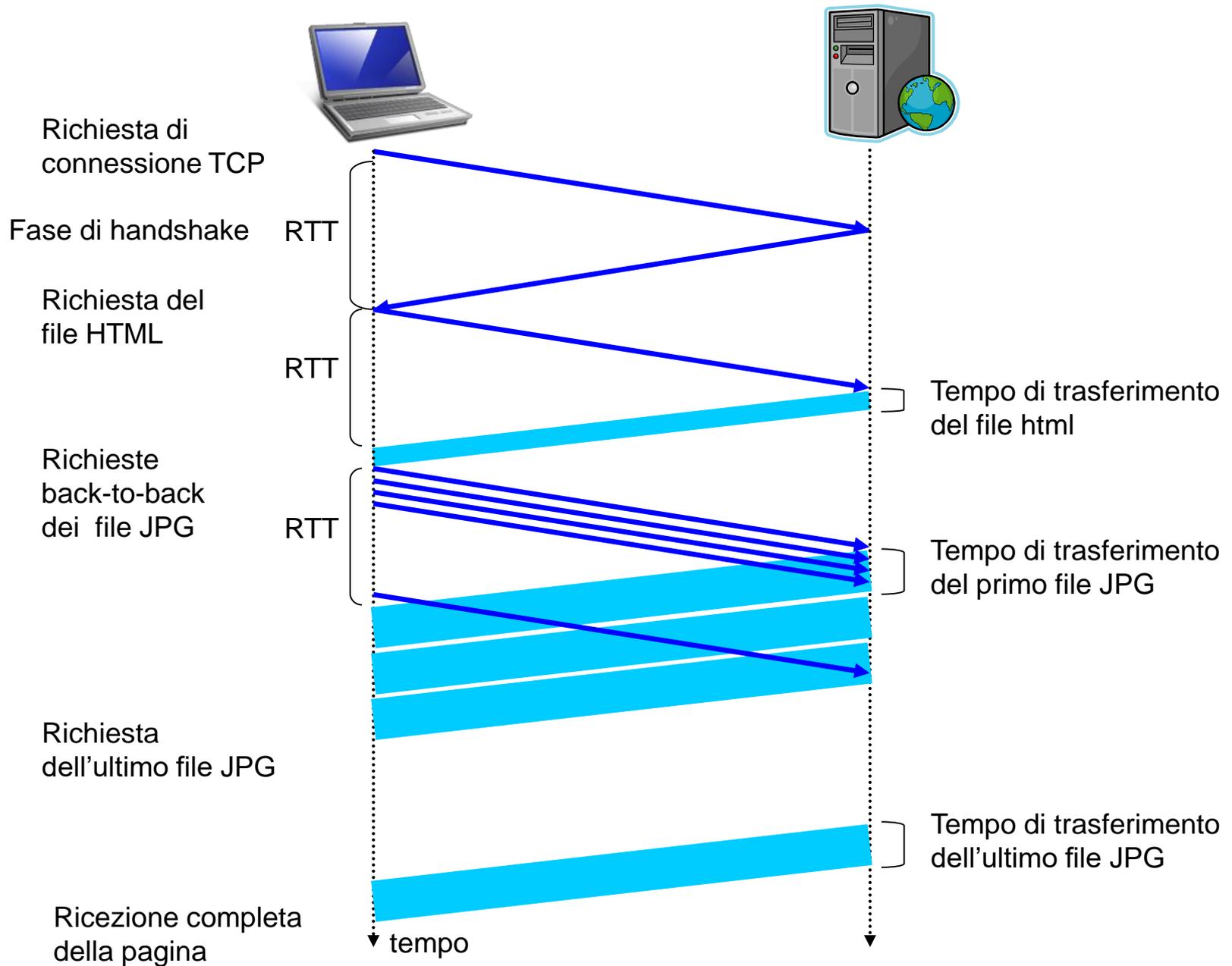
- Nella modalità persistente senza parallelismo, il client può inviare una nuova richiesta solo quando la risposta precedente è stata ricevuta.



- In questo caso prima che inizi il trasferimento di ciascuna delle 10 immagini, dell'esempio precedente, deve passare **un RTT**.

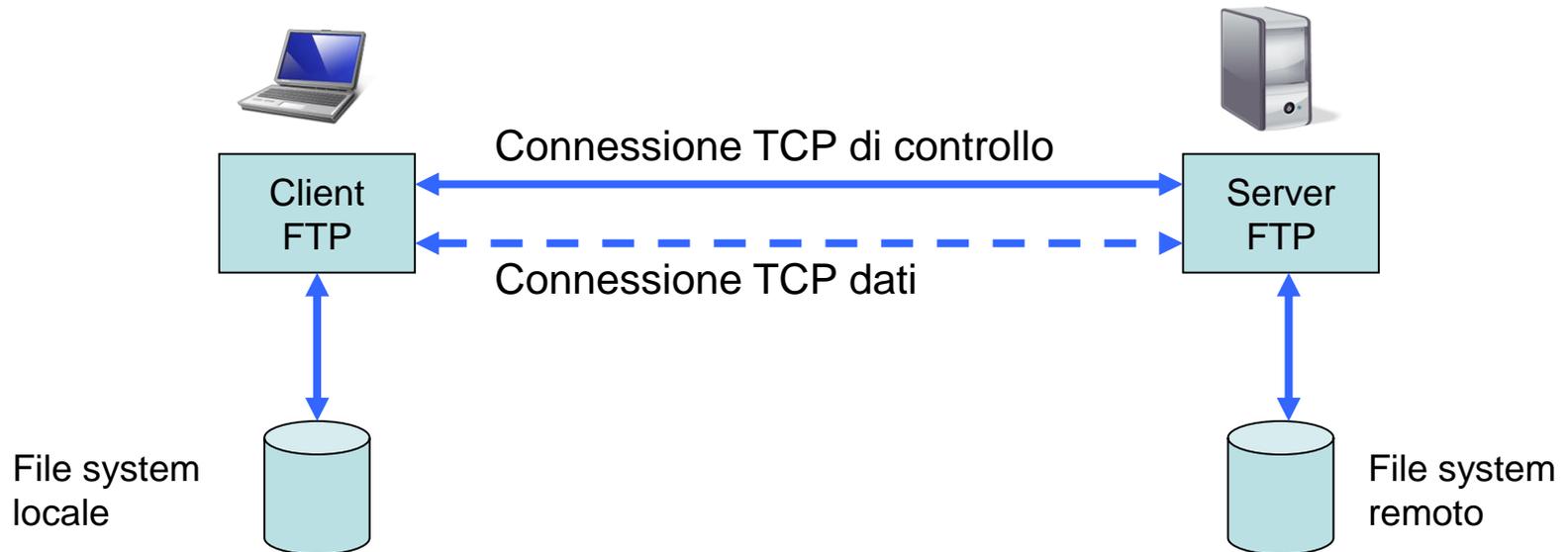
Connessione persistente con parallelismo

- Con la **connessione persistente con parallelismo**, il client può inviare una richiesta appena trova nella pagine HTML base un riferimento di file, senza attendere la risposta alla precedente richiesta.
- Quando il server riceve le richieste, invia i file uno dopo l'altro.
- Il parallelismo, quindi, utilizza **un solo RTT per tutti i file richiesti** invece di un RTT per ciascun file come avviene quando non si usa il parallelismo.
- Inoltre la connessione TCP con parallelismo resta attiva per tempo più piccolo rispetto a quella senza parallelismo.
- Per default, **l'HTTP/1.1** usa la connessione persistente con parallelismo.



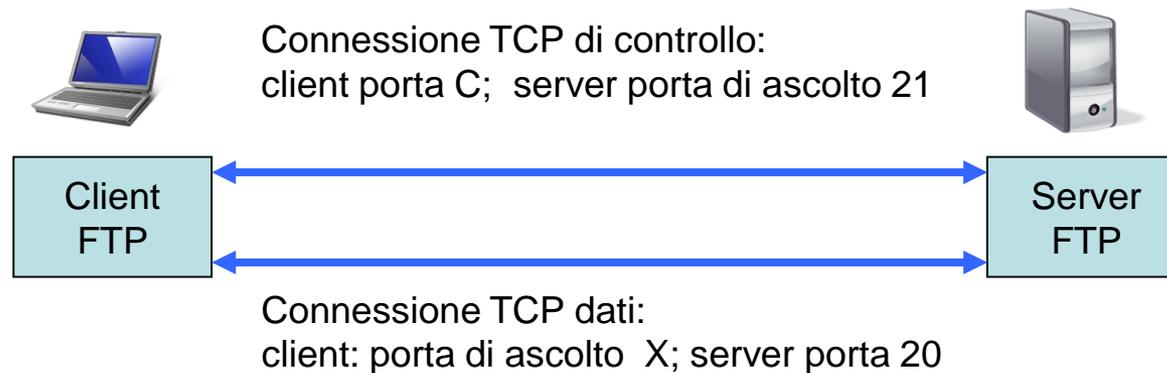
Il protocollo FTP

- Il protocollo FTP (**F**ile **T**ransfer **P**rotocol, *protocollo di trasferimento file*), consente di trasferire file da un host (locale) ad un altro host (remoto) e viceversa.
- FTP si è evoluto negli anni. La prima specifica (RFC 114), sviluppata presso il **MIT** (Massachusetts Institute of Technology), è del 1971. L'attuale specifica (RFC 959) è del 1985.

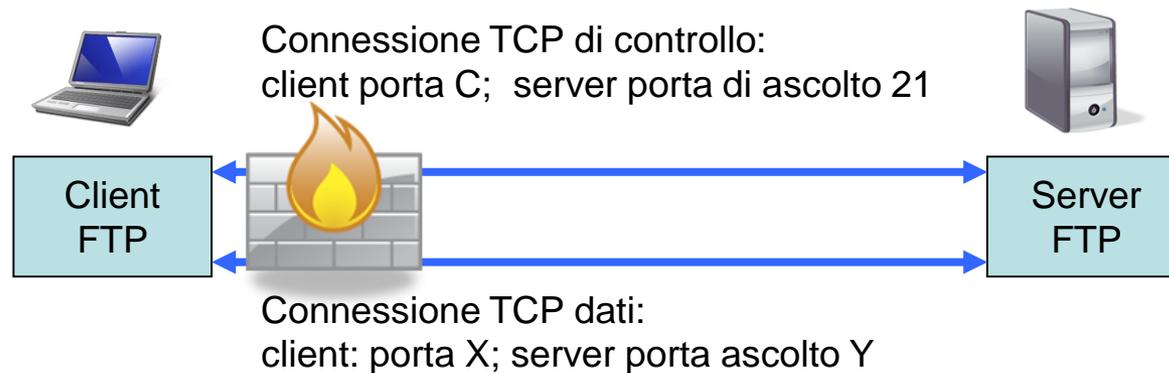


- Anche l'FTP, come l'HTTP, utilizza il TCP. L'FTP usa due connessioni TCP in parallelo: una **connessione di controllo** *per inviare i comandi* e una **connessione dati** per il trasferimento dei file.
- Il client prima instaura una connessione TCP di controllo con il server sulla **porta 21**. Il numero di porta di connessione del client è scelto automaticamente dal TCP che assegna il primo numero non usato superiore a 1024.
- Il client invia l'identificazione dell'utente e la password.
- Quando il client invia un comando per un trasferimento di file al server, viene creata una connessione dati TCP che sarà usata per il trasferimento di file. L'FTP invia un file sulla connessione dati e poi la chiude. Se, durante la stessa sessione, l'utente vuole trasferire un altro file, l'FTP apre un'altra connessione dati. Quindi, con FTP, la connessione di controllo rimane aperta per tutta la durata della sessione, ma viene stabilita una nuova connessione dati per ciascun file trasferito all'interno della sessione cioè, **la connessione dati non è persistente**.

- FTP può essere eseguito in **modalità attiva o passiva**, che determina il modo in cui viene stabilita la connessione dati.
- In **modalità attiva**, il client attraverso la connessione di controllo, invia al server mediante il comando PORT il numero di porta, ad esempio **X** su cui esso ascolterà e il server avvia la connessione dati.



- In situazioni in cui il client è protetto da un firewall e non può accettare le connessioni TCP in ingresso, può essere utilizzata la **modalità passiva**. In questa modalità, il client invia il comando PASV al server e riceve da questo, mediante il comando PORT, un numero di **porta Y** di ascolto che il client utilizza per aprire la connessione dati al server.



- Il trasferimento dei dati può avvenire in quattro modalità, in base al formato dei dati:
 - **Modalità ASCII:** utilizzato per il testo. I dati se necessario vengono convertiti al formato ASCII a 8-bit prima della trasmissione e (ancora una volta, se necessario) nella rappresentazione di caratteri dell'host ricevente. Di conseguenza, questa modalità non è adatta per i file che contengono dati diversi da testo normale.
 - **Modalità immagine** (detta anche modalità **binaria**): l'host mittente invia il file byte per byte e il destinatario salva il flusso di byte appena lo riceve.
 - **Modalità EBCDIC:** utilizzata per il testo normale tra gli host che utilizzano il set di caratteri EBCDIC. Questa modalità è altrimenti come la modalità ASCII.
 - **Modalità locale:** permette a due computer con configurazioni identiche di inviare dati in un formato proprietario senza la necessità di convertirlo in ASCII

- Durante una sessione, l'FTP deve mantenere lo stato della sessione dell'utente. In particolare, deve tenere traccia delle directory, locale e remota, mentre l'utente naviga all'interno del file system locale e remoto. Dovendo tenere traccia di queste informazioni di stato e mantenere aperta sempre la connessione di controllo, i SO limitano il numero totale di sessioni che l'FTP può aprire simultaneamente.

Comandi FTP

- Il client e il server si scambiano messaggi di richiesta e risposta, attraverso la connessione di controllo (porta 21), in formato ASCII a sette bit. Quindi, come per i messaggi HTTP, i comandi FTP sono leggibili.
- Ogni messaggio di richiesta (comando) ha un formato costituito da un massimo di quattro caratteri maiuscoli ASCII.
- Alcuni dei comandi più usati sono mostrati nella tabella seguente:

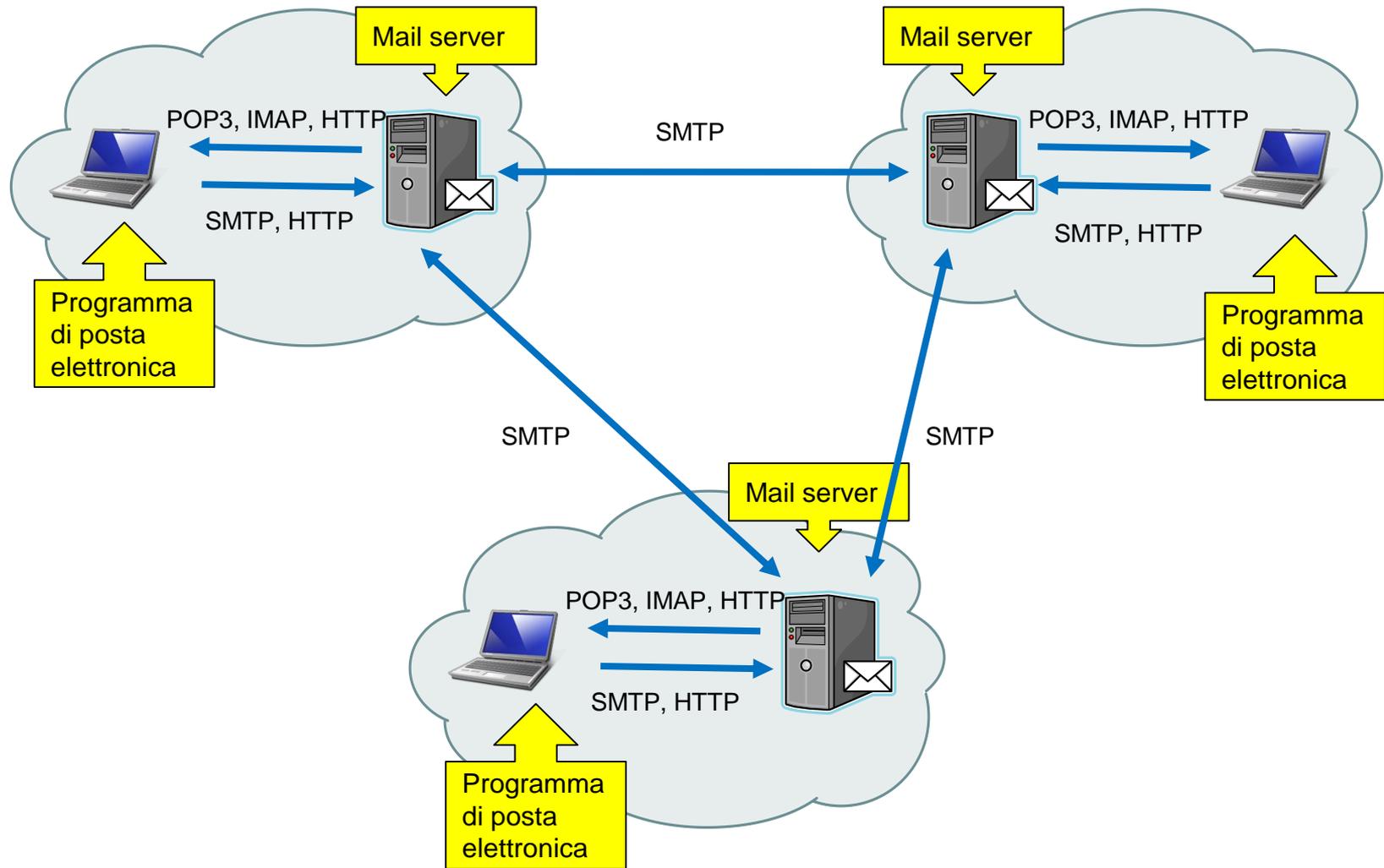
Comando	descrizione
CWD path	Cambia directory corrente
DELE path	Cancella il file specificato in path.
HELP	Ritorna la lista dei comandi accettati dal server. Con argomento fornisce spiegazioni riguardo il comando specificato.
LIST [path]	Visualizza un elenco di tutti i file contenuti nella directory remota. L'elenco dei file viene spedito su una nuova e non persistente connessione dati TCP.
PASS password	Invia la password al server
PASV	Inizializza una connessione passiva.
PWD	Visualizza la directory corrente remota.
QUIT	Chiude la sessione.
RETR path	trasferisce (<i>get</i>) un file dalla directory remota corrente alla directory corrente locale.
STOR path	trasferire (<i>put</i>) un file nella directory corrente dell'host remoto.
USER username	Invia l'identificativo dell'utente al server.

Posta elettronica

- La posta elettronica è stata una delle prime applicazioni di Internet. Col tempo è diventata sempre più sofisticata ed è una delle applicazioni di rete più utilizzate.
- La posta elettronica consente la comunicazione in modo asincrono: un'utente invia e riceve messaggi senza doversi sincronizzare con gli orari di altri utenti.
- Le attuali applicazioni di posta elettronica consentono di inviare messaggi di testo con allegati multimediali di qualsiasi tipo di file come musica, immagini, video, etc.
- Consente l'uso di **mailing list**, con le quali è possibile inviare un messaggio a più destinatari contemporaneamente.
- Attualmente sono stati realizzati numerosissimi programmi di posta elettronica. Ci sono anche applicazioni di pubblico dominio sia con interfaccia grafica che alfanumerica, come mail (unix) e pine.

- Il funzionamento della posta elettronica si basa su quattro componenti principali:
 - **i programmi di posta elettronica (*mail user agent*)**
 - **i server di posta elettronica (*mail server*)**
 - **l'SMTP (*Simple Mail Transfer Protocol*).**
 - **Protocolli per l'accesso alla posta (come *POP3, IMAP e HTTP*)**
- Quando un mittente termina di scrivere il suo messaggio, il programma di posta invia il messaggio al suo mail server, dove viene inserito nella coda dei messaggi in uscita. Quando il destinatario vuole leggere il messaggio, il suo programma recupera il messaggio dalla sua casella di posta nel suo server di posta.
- Un mail server è in grado di gestire molte utenze di posta dette mailbox (caselle di posta). Per ciascuna mailbox il mail server può memorizzare moltissimi messaggi.

La figura mostra uno schema generale della posta elettronica.



- Un messaggio inviato dal mittente, arriva al server di posta del mittente e quindi arriva al server di posta del destinatario, dove viene posto nella casella di posta.
- Quando un destinatario vuole accedere ai suoi messaggi, il server di posta esegue la **procedura di autenticazione** richiedendo username e password.
- Il server di posta gestiscono anche eventuali problemi che si verificano nel consegnare i messaggi al server del destinatario. Se il server mittente non può inviare la posta al server destinatario, allora mantiene il messaggio in una coda (**message queue**) e proverà a rinviarlo in seguito. L'intervallo di tempo tra tentativi successivi è configurabile e generalmente è di 30 minuti. Se il messaggio non viene spedito entro qualche giorno, il server rimuove il messaggio e notifica l'eccezione al mittente con un messaggio e-mail.

Indirizzo di posta elettronica

Un indirizzo di posta elettronica (email) è formato da due parti: la prima parte indica il nome dell'utente e la seconda il nome di dominio di appartenenza dell'utente. Le due parti sono separate dal carattere @. Ad esempio verdi@uniroma2.it è un indirizzo valido di posta.

SMTP

- L'SMTP, definito nella RFC 821, è il protocollo con architettura client/server usato per trasferire i messaggi di posta dal programma del mittente (client) al server del mittente e dal server mittente al server destinatario. Esso usa il TCP e la **porta 25**.
- In un mail server sono implementati sia il lato client che il lato server. Quando un server di posta invia messaggi, si comporta come client mentre quando riceve messaggi, si comporta come server.

- .

- Attualmente l'SMTP è da considerarsi una tecnologia superata per via di alcune sue caratteristiche obsolete. In particolare, i messaggi di posta hanno **formato ASCII a sette bit**. Questo formato andava bene nei primi anni '80 quando la velocità di trasmissione della rete era molto bassa e si inviavano prevalentemente messaggi di solo testo per e-mail senza allegati di grandi dimensioni.
- Con l'SMTP, per inviare dati binari multimediali (o anche dati ascii a 8 bit) è necessario che siano **codificati in ASCII a 7 bit prima della spedizione**. Quando i messaggi arrivano al destinatario è necessario decodificarli nel formato binario originale. L'HTTP invece non richiede che i dati multimediali siano codificati in ASCII prima del trasferimento

Esempio di dialogo tra il client SMTP e il server SMTP

- Con telnet è possibile attuare un dialogo diretto con un server SMTP:

```
telnet reti.uniroma2.it 25
```

```
220 reti.uniroma2.it ESMTP Sendmail 8.12.6/8.11.6;  
Mon, 26 Mar 2012 16:30:54 +0100 (CET)
```

```
MAIL FROM: frasca@mat.uniroma2.it
```

```
250 2.1.0 frasca@mat.uniroma2.it... Sender ok
```

```
RCPT TO: verdi@mat.uniroma2.it
```

```
250 2.1.5 frasca@mat.uniroma2.it... Recipient ok
```

```
DATA
```

```
354 Enter mail, end with "." on a line by itself
```

```
Ciao da Pietro
```

```
.
```

```
250 2.0.0 k0QGTswQ1109805 Message accepted for delivery
```

help

214-2.0.0 This is sendmail version 8.12.6

214-2.0.0 Topics:

214-2.0.0 HELO EHLO MAIL RCPT DATA

214-2.0.0 RSET NOOP QUIT HELP VERFY

214-2.0.0 EXPN VERB ETRN DSN AUTH

214-2.0.0 STARTTLS

214-2.0.0 For more info use "HELP <topic>".

214-2.0.0 To report bugs in the implementation send email to

214-2.0.0 sendmail-bugs@sendmail.org.

214-2.0.0 For local information send email to Postmaster at
your site.

214 2.0.0 End of HELP info

QUIT

221 2.0.0 reti.uniroma2.it closing connection

Connessione all'host perduta.

- Il server dà una risposta ad ogni comando. Ciascuna risposta è formata da un codice e da alcune descrizioni (facoltative).
- L'SMTP usa **connessioni TCP persistenti**: se il server di posta mittente ha vari messaggi da inviare allo stesso server di posta destinatario, esso può inviare tutti i messaggi sulla stessa connessione TCP.
- Ogni messaggio che un client vuole inviare inizia con il comando **MAIL FROM: indirizzo_mittente**, e termina con un punto su una riga. Il comando **QUIT** consente la spedizione di tutti i messaggi scritti.