Metodologie Informatiche Applicate al Turismo

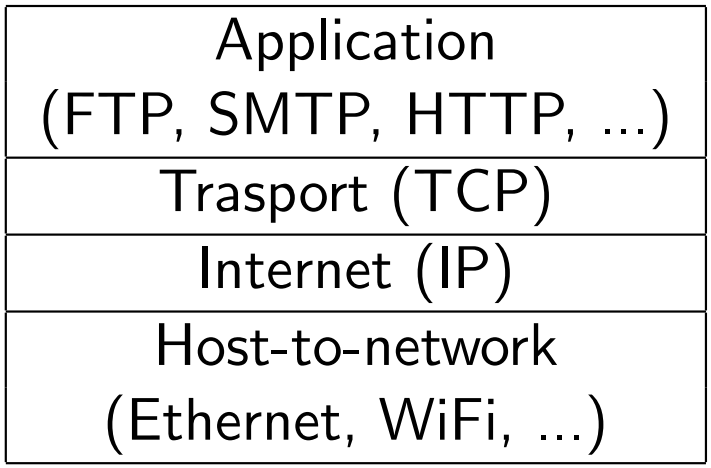
6. - Protocolli a livello applicazione

Paolo Milazzo

Dipartimento di Informatica, Università di Pisa
http://pages.di.unipi.it/milazzo
milazzo@di.unipi.it

Corso di Laurea in Scienze del Turismo
A.A. 2016/2017

I protocolli a livello applicazione



I protocolli di livello applicazione si dividono in due categorie:

- Protocolli di applicazione vera e propria (e.g. SMTP, HTTP, FTP, etc...) che forniscono il servizio agli utenti finali
- Protocolli di servizio (e.g. DNS) forniscono servizi alle applicazioni usate dagli utenti

Questi protocolli utilizzano TCP/IP come strumento di comunicazione affidabile tra nodi della rete.

I protocolli a livello applicazione

Vediamo un esempio di protocolli a livello applicazione:

- I protocolli per la posta elettronica (SMTP, POP3 e IMAP)

Tutti questi protocolli si scambiano messaggi testuali e definiscono:

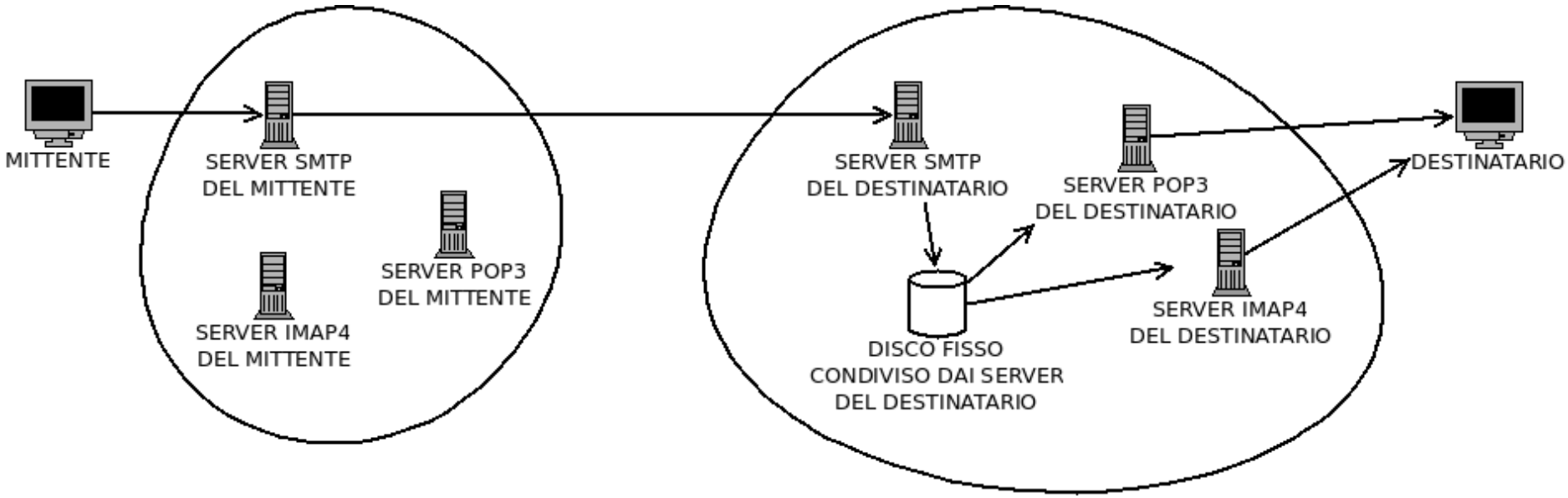
- Formato e significato dei messaggi di testo
- Regole per l'invio e la ricezione di tali messaggi

Tutti gli aspetti più tecnici (conversione in binario, instradamento sulla rete, controllo degli errori di trasmissione, ecc...) sono risolti a livello più basso da TCP/IP

Protocolli per la posta elettronica

La posta elettronica è basata sull'applicazione di 3 protocolli di VII livello:

- Simple Mail Transport Protocol (SMTP): trasporta un messaggio di posta elettronica dal computer del mittente fino alla casella di posta del destinatario (nel disco fisso del server del destinatario);
- Post Office Protocol ver. 3 (POP3) e Internet Message Access Protocol ver. 4 (IMAP4): consentono al destinatario di accedere alla propria casella di posta (sul proprio server) per scaricare la posta ricevuta



Il protocollo SMTP (1)

- SMTP è un protocollo text-based per lo scambio di messaggi di posta elettronica.
- Una connessione SMTP è composta da una apertura, una sequenza di comandi, e una chiusura.
- Ad ogni comando corrisponde una risposta composta da un codice numerico (per le applicazioni) e una stringa leggibile (per gli umani).
  - MAIL FROM:⟨Smith@alpha.com⟩
  - 250 OK
  - RCPT TO:⟨Green@alpha.com⟩
  - 550 No such user here

- Esempio di scambio di messaggi tra un client (programma di posta elettronica, in blu) e il server SMTP del mittente (in rosso)
  - La spedizione di un messaggio SMTP avviene attraverso l'identificazione del mittente (MAIL FROM), del/dei destinatari (RCPT TO), e del messaggio da trasmettere (DATA):
  - 220 alpha.com Simple Mail Transfer Service Ready
  - HELO beta.com
  - 250 alpha.com says: Nice to meet you beta.com
  - MAIL FROM:⟨Smith@alpha.com⟩
  - 250 OK
  - RCPT TO:⟨Green@alpha.com⟩
  - 550 No such user here
  - RCPT TO:⟨Brown@alpha.com⟩
  - 250 OK
  - DATA
  - 354 Start mail input; end with ⟨CRLF⟩.⟨CRLF⟩
  - Blah blah blah....
  - etc. etc. etc.
  - .
  - 250 OK
  - QUIT
  - 250 alpha.com Service closing transmission channel

I protocolli POP3 e IMAP4

- SMTP si occupa di recapitare un messaggio di posta elettronica nel disco fisso del server del destinatario
  - Allo stesso modo in cui un postino recapita la posta nella cassetta della posta
- Per consultare la posta ricevuta (cioè, per andare a ritirare la posta da dento la cassetta) esistno due protocolli altenativi:
  - POP3 permette ad un programma di posta eletronica di accedere alla mailbox posta su un server, consentendo soltanto la possibilità di scaricare e cancellare mail (e poco altro)
  - IMAP4 consente a un programma di posta elettronica di effettuare operazioni più "sofisticate" sulla casella di posta elettronica. Ad esempio: leggere i messaggi senza scaricarli (favorisce l'accesso da diverse postazioni) e creare cartelle sul server in cui suddividere i messaggi ricevuti
- Il protocollo POP3 è sempre meno usato...

Nomi e indirizzi: il DNS (1)

- I computer su Internet sono identificati dal loro indirizzo IP;
- Nel contesto del WWW è molto più comune utilizzare URL (Uniform Resource Locator, ossia indirizzi nella forma `http://www.di.unipi.it`) anzichè indirizzi IP;
- Il Domain Name System (DNS) è un servizio di naming globale che traduce URL in indirizzi IP.

L'utilizzo di URL anzichè indirizzi IP offre due vantaggi:

- Gli URL sono più facilmente leggibili e utilizzabili dagli utenti degli indirizzi IP;
- Gli URL offrono un livello di astrazione: se si riorganizza una rete cambiando alcuni indirizzi IP è sufficiente aggiornare il DNS per rendere il cambiamento invisibile agli utenti.

Nomi e indirizzi: il DNS (2)

Il DNS può essere visto come un elenco telefonico globale in cui i nomi anziché essere in ordine alfabetico hanno una struttura gerarchica

- I domini di primo livello (parte finale dell'URL) sono sigle di nazioni (e.g. `it`,`uk`,`fr`,`...`) o domini generici (e.g. `com`,`edu`,`org`,`...`);
- I domini di livello inferiore sono indicati da destra a sinistra separati da punti: e.g. in `www.di.unipi.it`
  - `it` è il dominio di primo livello;
  - `unipi` è il dominio di secondo livello attribuito all'Università di Pisa
  - `di` è il dominio di terzo livello attribuito al Dipartimento di Informatica;
  - `www` è il nome convenzionale del computer che ospita il web server in un dominio.

Nomi e indirizzi: il DNS (3)

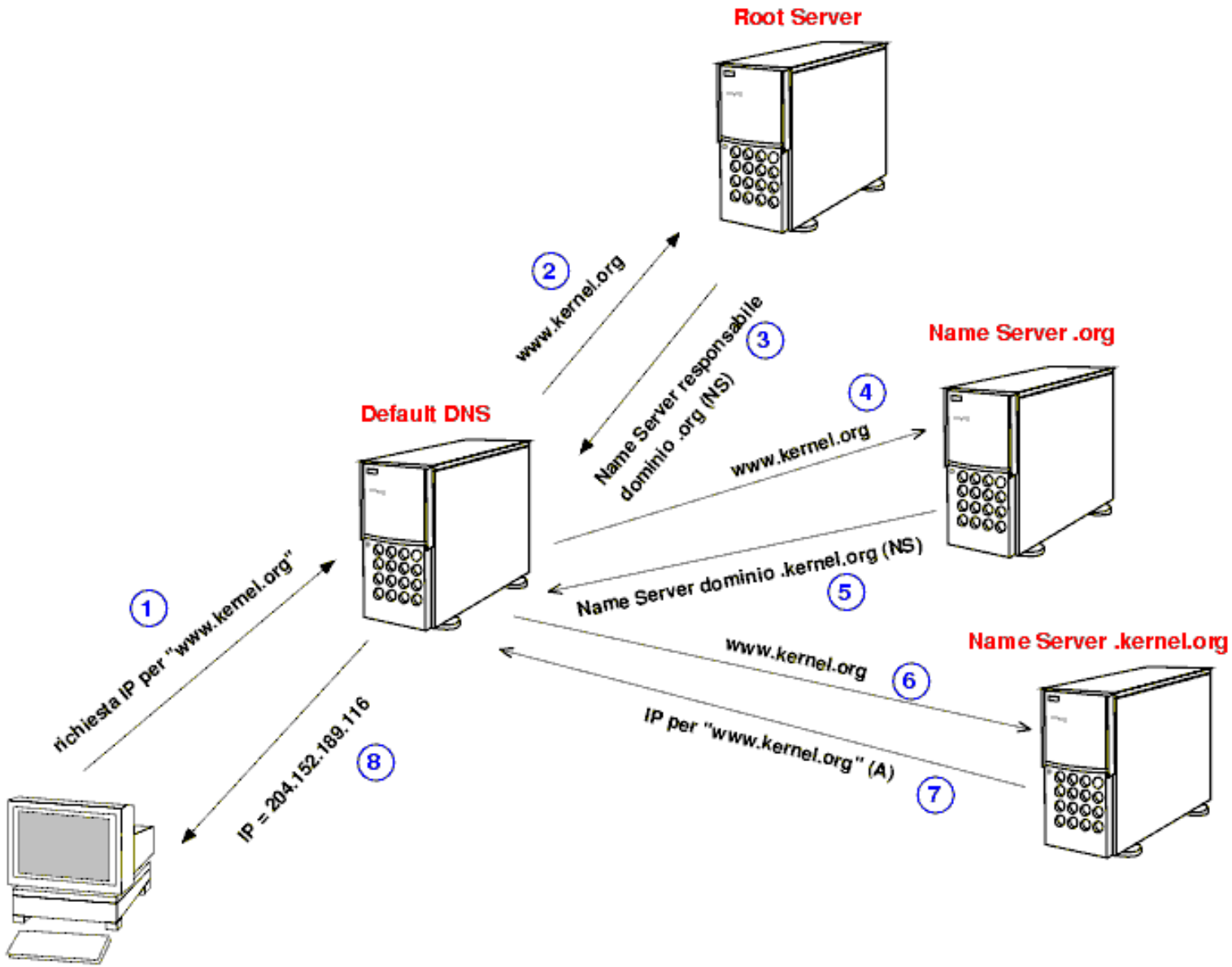
La risoluzione di un nome di dominio tramite DNS funziona come segue:

- Ogni computer collegato a Internet ha un server DNS di riferimento (default DNS, fornito dal fornitore di collegamento a Internet);
- Quando un computer deve risolvere un nome lo inoltra al proprio server DNS di riferimento;
- Se il server DNS non conosce l'indirizzo IP corrispondente al nome richiesto inoltra la richiesta a un *root server* (un server DNS che conosce gli indirizzi dei server DNS responsabili dei domini di primo livello);
- Il server DNS invia ora la richiesta al server responsabile del dominio di primo livello dell'indirizzo richiesto; Tale server fornisce la risposta richiesta, oppure fornisce l'indirizzo del server DNS del dominio di secondo livello di interesse;
- Il procedimento continua finchè la richiesta non riceve risposta.

I vari server DNS possono fare uso di tecniche di caching (memorizzazione) delle informazioni scambiate per evitare successivamente di ripetere tutti questi passaggi

Nomi e indirizzi: il DNS (4)

Un esempio (ricerca dell'indirizzo IP di `www.kernel.org`):



Il protocollo HTTP (1)

L'HyperText Transfer Protocol (HTTP 1.1) è un protocollo di comunicazione per la richiesta e la trasmissione di pagine web (e non solo)

Prevede messaggi testuali che consentono:

- A un client (il browser) di richiedere a un server (il sito) di inviare una certa pagina web
- A un server (il sito) di rispondere a una richiesta inviando un documento HTML che descrive la pagina (e/o altro, se serve)

Il protocollo HTTP (2)

Il protocollo HTTP, oltre alle figure del client e del server, può prevedere la presenza di intermediari, tra cui i proxy.

Un proxy:

- è un programma che riceve richieste HTTP dai client (opportunamente configurati: l'indirizzo di un eventuale proxy va indicato nel browser) e le inoltra ai server web;
- può applicare tecniche di caching dei contenuti ricevuti dai server per soddisfare autonomamente successive richieste dei client (senza contattare nuovamente i server);
- può applicare filtri ai contenuti ricevuti (e.g. parental control).

Il protocollo HTTP (3)

Il protocollo HTTP:

- E' stateless (senza stato): il risultato di una richiesta non può dipendere da richieste precedenti;
- Costringe lo sviluppatore di siti web a trovare metodi alternativi per memorizzare una forma di stato (e.g. cookies).

Un messaggio di richiesta HTTP è costituito da:

- Una riga di richiesta che contiene un "metodo" (descrizione del tipo di richiesta), un URL e l'indicazione della versione di HTTP utilizzata;
- Una sezione di header (contenente informazioni varie)
- Il corpo del messaggio

Il protocollo HTTP (3)

Il metodo può essere uno dei seguenti:

- GET: per richiedere al server l'invio della risorsa indicata nell'URL;
- POST: per inviare informazioni al server;
- HEAD: simile a GET, richiede al server l'invio di informazioni sulla risorsa indicata nell'URL (non la risorsa stessa);
- PUT: per richiedere al server di sostituire una risorsa specificata con un'altra;
- DELETE: per richiedere al server di cancellare una risorsa;
- TRACE: per ricostruire la sequenza di intermediari tra il client e il server;
- OPTIONS: per ottenere informazioni sulle opzioni di comunicazione.

I metodi più comuni (e i soli che approfondiremo) sono GET e POST.

Il protocollo HTTP (4)

Il metodo GET:

- è il metodo più comune, che viene utilizzato ogni volta, ad esempio, che si segue un link in una pagina web o che si inserisce un indirizzo in un browser;
- serve per richiedere al server l'invio di una pagina web indicata nell'URL
- nell'header del messaggio:
  - può porre condizioni sul formato (e.g. lingua della pagina) o le caratteristiche della pagina (dimensioni, data ultimo aggiornamento, ecc..)
  - può inviare al server informazioni addizionali (e.g. browser e sistema operativo utilizzato, pagina di provenienza)
- nel corpo del messaggio:
  - nulla...

Il protocollo HTTP (5)

Esempio di richiesta che usa il metodo GET:

```
GET www.di.unipi.it/index.html HTTP/1.1
Referer:  http://www.google.it/#q=informatica+unipi
User-Agent:  Mozilla/4.61 (Macintosh; I; PPC)
Accept-language:  en
```

Richiede al server l'invio della pagina `www.di.unipi.it/index.html`. La richiesta è stata generata seguendo un link ottenuto da una ricerca su google (pagina di provenienza), usando il browser Firefox versione 4.61 su un computer Apple. Si richiede inoltre di inviare la versione inglese della pagina.

Il protocollo HTTP (6)

Il metodo POST:

- Viene usato per trasmettere delle informazioni dal client al server;
- Esempio tipico di informazione trasmessa: i dati inseriti dall'utente in un modulo (form) in una pagina web;
- I dati trasmessi vengono inseriti nel corpo del messaggio;
- I dati vengono passati alla risorsa (e.g. di solito un programma capace di elaborare i dati ricevuti) specificata dall'URL.

Esempio di richiesta che usa il metodo POST (invio username e password):

```
POST www.di.unipi.it/login.php HTTP/1.1
User-Agent:  Mozilla/4.61 (Macintosh; I; PPC)

username=milazzo&password=12345678
```

Il protocollo HTTP (7)

Un messaggio di risposta a una richiesta di un contenuto è costituito da:

- Una "riga di stato" (status-line) che contiene un codice e una breve descrizione dell'esito della richiesta;
- Una sezione di header (che riporta informazioni varie)
- Il corpo del messaggio di risposta (che contiene di solito il documento HTML richiesto)

Le righe di stato più comuni includono:

200 Ok	metodo eseguito con successo
400 Bad request	errore sintattico nella richiesta
403 Forbidden	richiesta non autorizzabile
404 Not found	URL errato (molto comune)
500 Internal server error	Errore interno (comune con siti web dinamici)

L'header di una risposta a GET include anche il tipo MIME del dato trasmesso

Il protocollo HTTP (8)

Un esempio di richiesta (semplificata):

```
GET www.alpha.com/beta.html HTTP/1.1
```

E relativa risposta:

```
HTTP/1.1 200 OK
Date:  Mon, 28 Jun 2004 10:47:31 GMT
Server:  Apache/1.3.29 (Unix) PHP/4.3.4
Last-Modified:  Mon, 12 Jun 2004 11:32:12 GMT
Content-Language:  it
Content-Type:  text/html; charset=utf-8

<HTML> ....
```

Il protocollo HTTP (9)

- HTTP trasmette tutti i dati in chiaro;
- Esiste una versione di HTTP per comunicazioni che richiedono crittografia: HTTPS
- HTTPS:
  - utilizza TCP e SSL (crittografia) per trasmettere i soliti messaggi HTTP;
  - usa lo schema `https:` al posto di `http:` negli indirizzi.