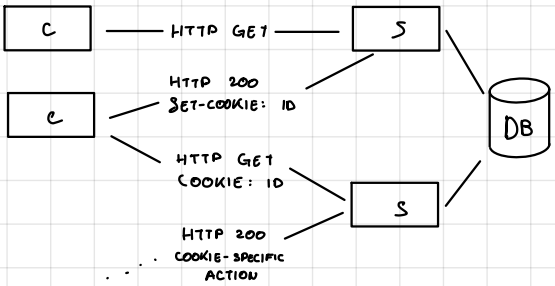
. . .

## 3.1 HTTP (RFC 1945, 2616)

. . .

### 3.1.1 i COOKIES

I cookies sono uno strumento per mantenere uno stato. I cookie sono header che vengono salvate sull'host.
Un database di cookie viene mantenuto dal server.
Quando il client visita per la prima volta il sito esso viene identificato da un ID unico.




### 3.1.2 PROXY HTTP

I proxy sono dei middle-man che hanno compito di rispondere alle richieste senza coinvolgere il server. Il client, quindi, parla con il proxy. Il server vede solo richieste da parte del proxy.
Il proxy può anche cachare delle risorse in modo da renderle più accessibili ad altri utenti del proxy
I proxy sono degli application gateway, ovvero instradatori a livello applicativo.

### 3.1.2 HTTP 2.0

L'obiettivo è ridurre i loading time dei siti e alcuni problemi del HTTP 1.1:
- lavora in binario (trasporta frame)
- usa il pipelining (comunicazioni in parallelo)
- gli header vengono compressi:
  - codifica di Huffman
  - indexing
  - codifica differenziale
- si usa il server push: il server manda risposte senza richiesta
- usa TLS
- offre flow control (tramite stream)

### 3.1.3 HTTPS

HTTP non offre garanzie su integrità, confidenzialità e autenticazione tra server e client.
Le parti di sicurezza possono essere aggiunte a livello di trasporto da SSL e TLS.
Le connessioni SSL/TLS si dividono in:
- HANDSHAKE: le due parti si identificano e si mettono d'accordo su cifratura e si scambiano le chiavi
- TRANSFER: ogni PDU è cifrata con la cifratura decisa
- CLOSE: viene chiusa la connessione e le chiavi vengono dimenticate.

La parte di handshake è la più critica:
- viene scambiato un certificato generato da una CA. Esistono diverse CA riconosciute salvate nei browser dette root certificates. Un certificato contiene:
  - chiave pubblica dell'entità certificata;
  - informazioni sul server;
  - firma digitale della CA;
- viene generata una chiave simmetrica per la cifratura delle PDU; (PUB → ENCRYPT; PRIV → DECRYPT)
- le chiave simmetriche vengono inviate su una connessione con cifratura asimmetrica (viene utilizzata la chiave pubblica nel cert)

## 3.2 POSTA ELETTRONICA (SMTP, POP3, IMAP)

La posta elettronica è composta da client (User agent) e dai mail server. Ogni mail server contiene le varie caselle di posta degli utenti e una coda di messaggi da inviare.

I server di posta comunicano in SMTP e si comportano sia da server, quando ricevono posta da un User Agent, sia da client, quando deve inviare un messaggio ad altri server.

Per leggere la posta vengono usati i protocolli POP3 e IMAP

### 3.2.1 SMTP

È client-server codificato ASCII. Usa il TCP sulla porta 25. Sia header che corpo sono codificati in ASCII, quindi anche i messaggi devono essere codicati in ASCII.

Il colloquio avviene con comandi tipo HELO, MAIL FROM, RCPT TO, DATA e QUIT Non vi è nessun tipo di autenticazione nel protocollo. I meccanismi di autenticazione sono stati aggiunti tramite una connessione a un server d'entrata (verifica mittente) e uno d'uscita (verifica ricevitore). La ricorrenza è molto leggera.

I dati della mail sono strutturati in HEADER e BODY (ASCII) Alcuni header sono 'Subject:', 'To:' e 'From:'. Per aggirare la restrizione ASCII viene usata l'estensione MIME che permette di trasferire mail multiparte, con ogni parte identificata da un tipo. I dati binari vengono convertiti in ASCII grazie alla codifica in base64.

### 3.2.2 ACCESSO MAILBOX

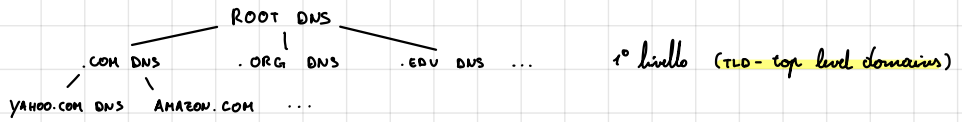
Vengono usati principalmente POP3 e IMAP. Il POP3 è più vecchio e più semplice.

La comunicazione POP3 avviene di solito sulla porta 110 è divisa in:
- autenticazione: il client si identifica coi comandi 'USER' e 'PASS'. Il server risponde +OK o -ERR.
- transazione: il client lavora sulla sua casella con comandi tipo 'list', 'retr', 'dele' e 'quit'.

## 3.4 DNS

Gli indirizzi IP sono numeri. I numeri sono difficili da ricordare. Il DNS è un'applicazione che traduce nomi simbolici in indirizzi IP. Il DNS ha un database distribuito che contiene le varie associazioni e usa una comunicazione UDP per scambiare info. Offre anche altri servizi come ad esempio load distribution.

Il DNS è strutturato in gerarchia:




I vari name servers (NS) sono di due tipi:
- LOCAL: forniti dall'ISP collegati direttamente con l'host.
- AUTHORATIVE: NS responsabile di un particolare hostname

Ogni host ha configurato l'indirizzo del LNS.

La risoluzione può avvenire in due modi: ITERATIVO e RICORSIVO.

Esempio iterativo:
1. il client DNS dell'host comunica con LNS
2. LNS contatta il root NS.
3. il Root NS segnala al LNS il TLD responsabile
4. il LNS contatta il TLD e gli chiede il sito
5. il TLD segnala l'authoritative NS del sito
6. il NS autoritativo comunica l'IP del sito.

Esempio ricorsivo:
1. il client DNS dell'host comunica con LNS
2. LNS contatta il root NS.
3. il Root NS contatta il TLD responsabile
4. il TLD contatta il NS autoritativo.
5. il NS autoritativo rifonisce l'IP
6. 7. 8. l'informazione torna su

Un NS, una volta ripreso l'IP di un dominio su cui non ha autorità può memorizzarla per rendere successivi lookup più veloci. Ogni cache ha un TTL (time to live) che decide per quanto tempo la cache è valida.
Le informazioni usate dai DNS hanno questo formato:

NAME, VALUE, TYPE, TTL

Il tipo può essere:
- 'A': NAME è il nome di un host e VALUE è l'IP.
- 'NS': NAME è un domain e VALUE è il nome di un NS che può ripetere le info di NS.
- 'CNAME': NAME è un alias per un host il cui vero nome è VALUE
- 'MX': NAME è dominio di mail o alias e VALUE è il nome del mail server.