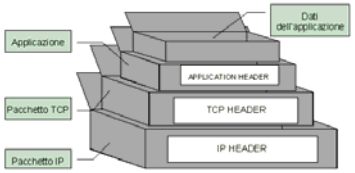
Il livello applicativo dell'architettura TCP/IP

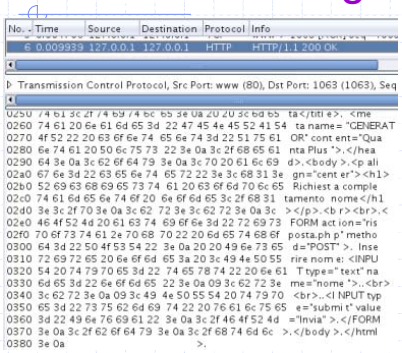
Dott. Libanore Luca

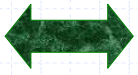
Livello 5: finalmente arriva l'Utente!



Vedremo ***come***, appoggiandosi sul livello di trasporto, e poi su tutti i livelli sottostanti della pila di protocolli, le ***applicazioni*** *interagiscono* per fornire i servizi all'**utente**

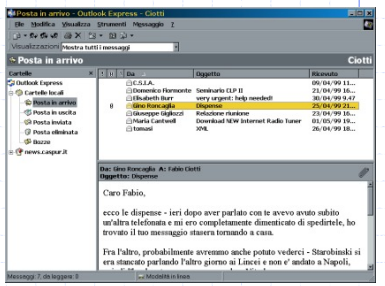
I protocolli e i servizi associati: uso il WWW grazie a HTTP





Vedremo come funzionano i **protocolli** (Es. HTTP) associati ai principali **servizi** (Es. WWW) utilizzati dagli utenti

Cos'è un servizio?



Un ***servizio*** è:

"qualsiasi prestazione fornita da ..... (ente pubblico, impresa privata, etc.) che serva a **soddisfare** un'esigenza della collettività (o di un gruppo di utenti)"

Cos'è un protocollo?

I protocolli

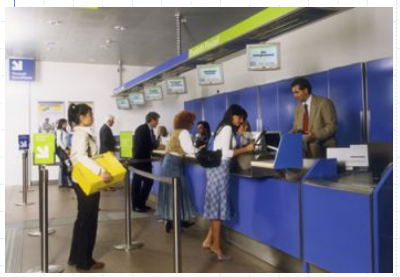
forniscono le **regole** per la comunicazione

1. Alzare il braccio per chiedere di parlare
2. Non si parla mai contemporaneamente
3. .....

contengono i dettagli dei **formati** dei **messaggi**

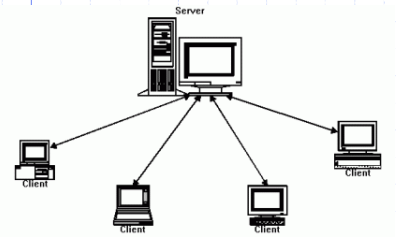
La grammatica della lingua (italiano, inglese, arabo, etc.) utilizzata per comunicare

Alcune regole dei protocolli di posta ordinaria



1. Le lettere vanno consegnate all'ufficio postale o in un'apposita buca delle lettere
2. Se il destinatario della lettera è un utente dell'ufficio postale allora la consegna avviene direttamente
3. Se il destinatario NON è un utente dell'ufficio postale allora la lettera viene inoltrata all'ufficio postale competente
4. .....

Alcune regole dei protocolli di posta elettronica

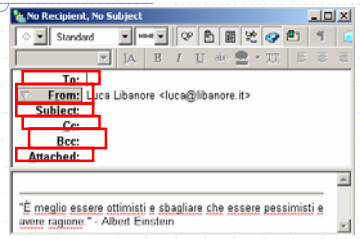


1. Le e-mail vanno spedite ad un server SMTP
2. Se il destinatario dell'e-mail possiede uno spazio su disco (mailbox) presso il medesimo server SMTP allora la consegna è immediata
3. Se il destinatario NON possiede una mailbox sul server SMTP allora l'e-mail viene inoltrata al server SMTP competente
4. .....

Alcuni aspetti del formato dei messaggi del protocollo di posta ordinaria

DIMENSIONI (*)	PICCOLO	MEDIO
Altezza (mm)	120	250
Lunghezza (mm)	235	353
Spessore (mm)	5	25
Peso (g)	50	2000

Alcuni aspetti del formato dei messaggi del protocollo di posta elettronica



"Lines in a message MUST be a maximum of 998 characters excluding the CRLF, but it is RECOMMENDED that lines be limited to 78 characters excluding the CRLF."

Tratto dal RFC 2822

"La dimensione massima di una e-mail che può essere inviata utilizzando una casella di posta registrata su un contratto Tin.it Free non può superare i 3 MB"

Tratto da http://help.virgilio.it/assistenza/index.jsp?id=6080

Quindi il protocollo come...

Insieme del

come comportarsi (regole)

e del

come devono essere i messaggi che ci si scambia (formato)

I servizi non sono realizzati con tecniche particolarmente difficili!

I protocolli che andremo a vedere sono prevalentemente dei protocolli ASCII → gli applicativi, appoggiandosi su una connesione gestita dal livello di trasporto, si scambiano dei **messaggi/comandi testuali**

Che protocolli/servizi andremo a vedere?

- Protocollo DNS → Servizio: risoluzione di nomi di host in indirizzi IP
- Protocolli SMTP, POP3, IMAP → Servizio: Posta elettronica

Il Domain Name System (DNS)

Dott. Libanore Luca

Quale esigenza soddisfa il DNS?

È più semplice ricordare:

A. 100011110110101.....1001

Indirizzo di rete su 32 BIT

B. 216.239.59.104 (rappresentazione in quadruple dell'indirizzo di rete)

C. google.it

D. Nessuna delle precedenti?!

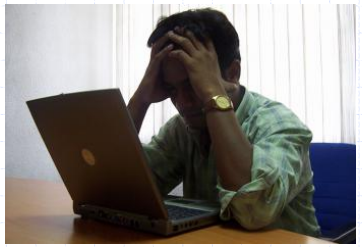
DNS soddisfa l'esigenza di RENDERE AGEVOLE l'utilizzo della rete e soprattutto L'IDENTIFICAZIONE dei nodi della rete

Cosa succederebbe senza DNS?

Nel caso volessi vedere le ultime notizie provenienti dal sito della Gazzetta dovrei....

Per l'utente finale....

lavorare con indirizzi espressi come 194.20.158.102, magari seguiti dal numero della porta (80 nell'esempio precedente), non è qualcosa di agevole, meno che mai mnemonico



L'elaboratore dovrebbe agevole il compito di un utente, non complicarlo!

Ed, infatti, è quello che succede con il DNS!

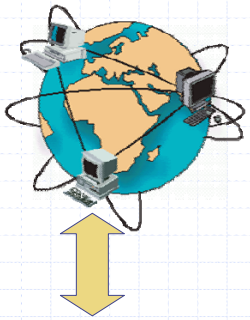
Il DNS è un SISTEMA
che permette di utilizzare
dei NOMI al posto degli INDIRIZZI IP,
realizzando una TRADUZIONE AUTOMATICA
dal nome all'indirizzo

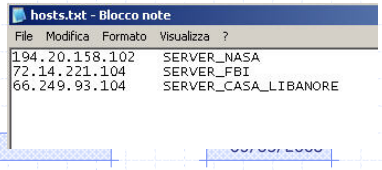
Il DNS permette di...

utilizzare delle ***stringhe di caratteri ASCII*** di tipo mnemonico, più facili da ricordare e più utili **PER L'UTENTE**

Primi passi: Vediamo che sistema si usava prima di inventare il DNS

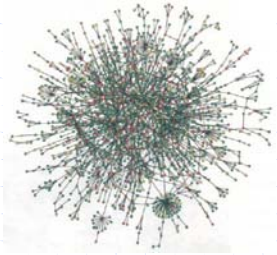
- L'esigenza di associare nomi e indirizzi di rete è nata quando Internet era ancora molto piccola
- Essendoci **pochi computer** era sufficiente utilizzare un *file che mettesse in corrispondenza gli indirizzi di rete e i nomi* che gli utenti volevano associare ai nodi della rete
- Il passo successivo era **distribuire** questo **file a tutti i nodi partecipanti alla rete stessa**





Problemi (come sempre!)... e se Internet diventa gigante?

Internet, con il passare del tempo, si è evoluta fino a diventare enorme (=tantissimi nodi)



Questo è un tipico esempio di problema di ***scalabilità***

In che senso scalabilità?

VS.

Scalabilità: capacità di un sistema di "crescere" (o "decrescere") in funzione delle necessità e delle disponibilità

Esempio dal "Mondo che ci circonda" numero 1

- La bicicletta è un mezzo di trasporto scalabile, non scalabile o poco scalabile?
- L'automobile?
- Quale dei due mezzi è più scalabile?
- Provate a fare un altro esempio tratto dal mondo reale che spieghi bene il concetto di scalabilità. (possibile domanda d'esame!)

Questa prima soluzione di associare nomi e indirizzi in un file è scalabile?

Un sistema che funziona bene sia con piccoli che grandi numeri è detto scalabile

La soluzione del file non è per niente scalabile!

È facile capire perchè non era scalabile.....

1. Quando i nodi sono diventati tantissimi, il file a sua volta ha assunto dimensioni gigantesche

2. Questo sistema prevedeva che, il file contenente l'associazione NOME-INDIRIZZO DI RETE, venisse distribuito a TUTTI i partecipanti della rete → inoltre c'erano problemi di aggiornamento del file!

Allora questo sistema lo buttiamo via?!

NO, anzi!

Questo file, che prende il nome di **hosts**, esiste ancora in molti sistemi operativi, sia in ambito Windows che in ambito Unix

Ma **a cosa serve** dato che il sistema non è scalabile? **DEFINIZIONE LOCALE, CACHING, …**

Esempio dal “Mondo che ci circonda” numero 2: hosts in Windows XP SP2

EditPlus - [C:\WINDOWS\system32\drivers\etc\hosts]

File Edit View Search Document Project Tools Window Help

```
1  # Copyright (c) 1993-1999 Microsoft Corp.
2  #
3  # Questo è un esempio di file HOSTS usato da Microsoft TCP/IP per Windows.
4  #
5  # Questo file contiene la mappatura degli indirizzi IP ai nomi host.
6  # Ogni voce dovrebbe occupare una singola riga. L'indirizzo IP dovrebbe
7  # trovarsi nella prima colonna seguito dal nome host corrispondente.
8  # L'indirizzo e il nome host dovrebbero essere separati da almeno uno spazio
9  # o punto di tabulazione.
10 #
11 # È inoltre possibile inserire commenti (come questi) nelle singole righe
12 # o dopo il nome del computer caratterizzato da un simbolo '#'.
13 #
14 # Per esempio:
15 #
16 #      102.54.94.97     rhino.acme.com          # server origine
17 #       38.25.63.10     x.acme.com              # client host x
18
19 127.0.0.1       localhost
20 192.168.0.4     server
21 216.239.59.104  google.it
```

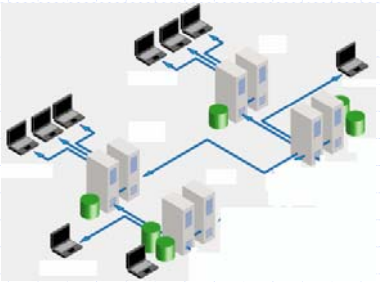
Date un'occhiata al sito **hpHosts**

- http://www.hosts-file.net/
- E' interessante perché propone dei file ***hosts*** di diverso genere
- Sono file ***hosts*** utili per ambiente Windows/MacOS/Linux
- Proviamo a vedere insieme il contenuto di uno di questi file ***hosts***! Secondo voi, in questo caso, qual è l'utilità di questo file?

Ora però pensiamo ad un sistema SCALABILE..

.. Ovvero utilizzabile in tutti casi (sia grandi che piccole dimensioni!)

In generale, abbiamo bisogno di un sistema che memorizzi, in modo intelligente, l'insieme dei nomi della rete Internet! Questo viene fatto con una sorta di **Database distribuito**



Il sistema utilizzato al giorno d'oggi: Domain Name System (DNS)

- Come detto, si basa su un Database distribuito dei nomi.
- È in grado di gestire una grandissima quantità di nomi (risolve il problema della scalabilità).
- Ma cosa utilizza il DNS per funzionare? Una RETE di SERVER che sono in grado di rispondere alle nostre richieste.

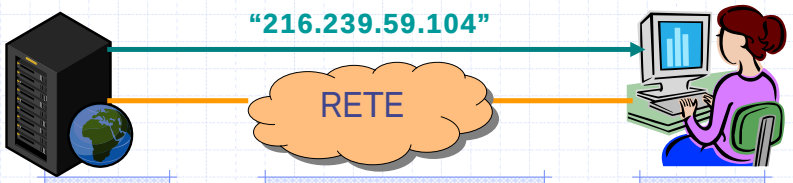
Come funziona il DNS?

Atto 1, Scena 1: Il programma applicativo alla ricerca dell'indirizzo perduto!

- Attori protagonisti: Programma applicativo (APPS1), Server DNS (S_DNS)
- Attore non protagonista: Il nodo www.google.it
- Antefatto: APPS1 vuole collegarsi a www.google.it ma prima DEVE conoscere il suo indirizzo IP
- Dialogo: APPS1 rivolto a S_DNS (che attende richieste sulla Porta 53):

Atto 1, Scena 2: S_DNS, ovvero il processo in attesa sulla porta 53, risponde

- Attori protagonisti: Programma applicativo (APPS1), Server DNS (S_DNS)
- Antefatto: S_DNS, dopo aver ricevuto la richiesta da APPS1, consulta il suo database interno, elabora la risposta e spedisce il risultato a APPS1
- Dialogo: S_DNS rivolto a APPS1:



Atto 1, Scena 3: Finalmente APPS1 può contattare www.google.it!

- Attori protagonisti: Programma applicativo (APPS1), Il nodo www.google.it
- Svolgimento del racconto: APPS1 è venuto finalmente a conoscenza di dove si trova www.google.it; ora APPS1 non conoscerà solo più un nome, ma avrà a disposizione anche l'***INDIRIZZO*** preciso per contattare www.google.it e ***INOLTRARE*** le sue ***RICHIESTE APPLICATIVE*** (Es: "Dammi la tua pagina iniziale!")
- E tutti vissero felici e contenti (o quasi!)

Approfondimento per chi ancora non ha capito come funziona il DNS?!

- Scena: in treno incontrate un vostro amico che vi presenta una ragazza bellissima (o un ragazzo bellissimo). Ammaliati da questa creatura divina non vi osate chiederle dove abita per farle recapitare delle rose (o delle lettere). Dopo esser scesi dal treno e aver sbattuto 30 volte la testa contro il muro, come pensate di fare per non farvela sfuggire? Vi recate nel palazzo dove abita il vostro amico e bussate alla sua porta (casualmente la numero 53). Lui cercherà nella rubrica il nome della ragazza e vi comunicherà l'indirizzo di dove abita! Per stavolta siete salvi! Buona fortuna!

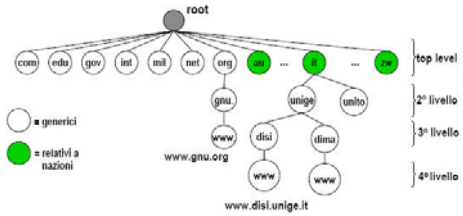
In realtà, il DNS non è solo una sorta rubrica, ma è un contenitore di tante informazioni utili

- ***Il DNS***, proprio come il vostro amico del racconto di prima, NON fornisce solo indirizzi ma ***contiene anche*** un certo numero di ***altre informazioni utili*** per la rete (l'amico per la ragazza bellissima!)
- Questo poiché ci sono ***altre informazioni di gestione relative ad Internet*** che a questo punto possono stare dentro il DNS, beneficiando di questo sistema di database distribuito

Prima di parlare di queste ulteriori informazioni, guardiamo come sono organizzati i nomi in Internet

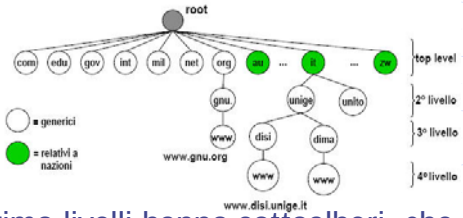
- Parliamo ora di **Gerarchia dei nomi (o dei domini)**
- L'organizzazione dei nomi in Internet è di tipo gerarchico, ovvero strutturata su differenti livelli ognuno con un'importanza diversa
- *Qual è la struttura dati, in informatica, che meglio esprime il concetto di gerarchia a livelli?*

I “Sovrani della gerarchia”: Domini Top Level



Nel primo livello di nomi, che deriva direttamente dalla radice di questa struttura ad albero, abbiamo una suddivisione dei vari nomi, fra i domini generici (.com per i domini a scopo commerciale, .edu per i domini a scopo didattico, etc.) oppure i domini nazionali che sono identificati dalla sigla del paese

E negli altri livelli cosa troviamo?



I nomi di primo livelli hanno sottoalberi, che rappresentano la suddivisione in domini dei singoli domini o generici o nazionali
Ad esempio, nel dominio it, abbiamo il dominio relativo all’università di Genova e poi all’interno dell’Università di Genova i vari dipartimenti

Che vantaggi abbiamo ad avere un'organizzazione dei nomi di questo tipo?

La struttura che abbiamo appena visto ci permette di avere *una costruzione dei nomi con una certa regolarità*

Il primo termine che compare nel nome, in questo caso, è quello di un calcolatore

Questo modo di organizzare i nomi aiuta la loro facilità ad essere ricordati (che è uno degli scopi del DNS!)

Dopo aver visto come sono organizzati i nomi, vediamo come sono organizzati i SERVER DNS

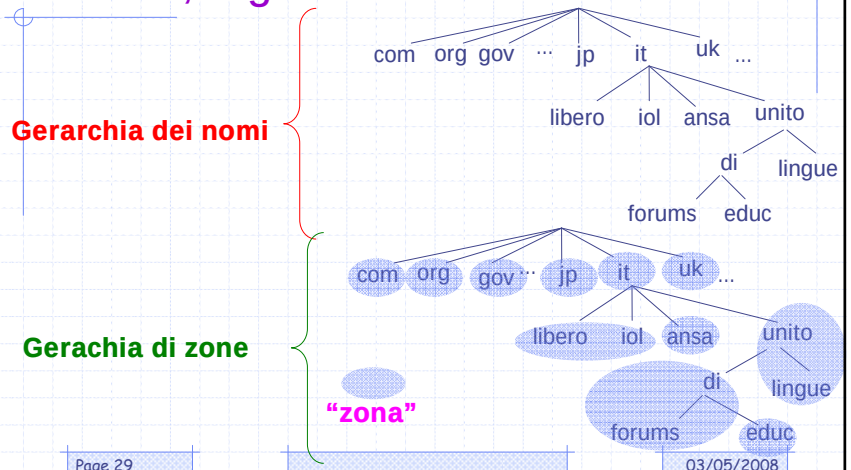
Nelle prossime slides vedremo com'è organizzata la rete dei server DNS

La rete dei server DNS è organizzata a ZONE

Ma cosa vuol dire questo? Vuol dire che, ***all'interno della gerarchia dei nomi abbiamo una struttura gerarchica di zone***

Difficile?! La prossima slide ci aiuterà a capire meglio il senso di struttura gerarchia di zone

Ecco cosa vuol dire struttura gerarchica di zone, o gerarchia dei name server



Esistono le zone DNS per una questione di responsabilità (altrimenti ritorniamo al problema del file gigante!)

- ***Una zona DNS è una parte dello spazio dei nomi,*** che è sotto una ***stessa gestione amministrativa*** e quindi è gestita da uno o più server
- ***Una zona DNS è costituita da un dominio e dai suoi sottodomini che non sono a loro volta delegati*** (potrebbero esserci sottodomini esclusi dalla zona perché delegati/affidati ad un'altra zona, vedi l'esempio di educ!)

Vediamo da vicino come i server DNS sfruttano la gerarchia a zone

Al secondo livello, troviamo, ad esempio un *server DNS* che si occupa di tutti i *resource record* riguardanti sia *libero* che *iol*; il server DNS è lo stesso per entrambi i domini; se al server DNS viene richiesta un'informazione riguardante libero o iol risponderà direttamente lui, se invece la richiesta riguarda, ad esempio, unito allora inoltrerà la query al server di competenza, dato che unito non fa parte della sua zona

E per unito cosa succede?

Ora guardiamo un caso un po' particolare! Il dominio di secondo livello **unito** ha due sottodomini (in realtà ne ha molti di più!): ***lingue***, che fa parte della stessa zona di unito, e ***di*** che fa parte di un'altra zona; ci sarà quindi un server DNS che contiene i resource record di ***unito*** e del suo sottodominio ***lingue*** MA NON del sottodominio ***di***, che sarà di competenza di un altro server DNS

Ma i server che sono competenti per una zona come si chiamano? AUTORITATIVI

- Si dice che il ***server DNS è AUTORITATIVO per una zona, quando contiene i resource record ad essa relativi***
- Quando ad un server DNS viene fatta una richiesta per un nome di cui NON è AUTORITATIVO, si connetterà con altri DNS di altre zone per reperire informazioni relative a nodi remoti

I server AUTORITATIVI: Ovvero "stasera ho già un altro impegno e non posso uscire".... Ma...

Vi è mai capitato che un vostro amico vi telefoni per chiedervi di uscire e voi dobbiate rispondere "Mi dispiace ma ho allenamento". Ovviamente il passaparola farà si che tutti i vostri amici vengano a conoscenza che voi, quella serata, non ci sarete. E se per caso l'allenamento salta perché l'allenatore si è ammalato? Ecco l'informazione che hanno i vostri amici non è più valida/affidabile! Solo voi siete AUTORITATIVI rispetto agli accadimenti della vostra vita e di quello che vi circonda! Gli altri si basano su informazioni provenienti da voi ma per cui non possono garantire siano sempre valide! L'informazione ritornerà valida quando voi la comunicherete, ma nonostante questo potrebbe nuovamente cambiare in breve tempo (Es. se chiama una bella ragazza). *Come potete capire è affidabile SOLO una notizia che proviene DIRETTAMENTE da voi* (ovvero dal server AUTORITATIVO!)

Ma nel caso del DNS quando diventa nuovamente valida l'informazione?

Se con i vostri *amici* l'*INFORMAZIONE* ritornerà *VALIDA quando VOI LO COMUNICHERETE*, *nel* sistema *DNS* l*'INFORMAZIONE* torna *VALIDA quando i server NON AUTORITATIVI CHIEDONO* (tramite query) *a quello AUTORITATIVO* nuovamente l'associazione indirizzo di rete/nome

Questo accade quando scade il TTL legato ad un particolare record! (non preoccupatevi capiremo nelle prossime slides questo cosa significa!)

Normalmente sono presenti più server autoritativi per una zona

I diversi server che sono delegati per una zona dovrebbero contenere le stesse informazioni, in modo che uno qualsiasi di questi possa rispondere ad una query per un record della zona (LOAD BALANCING e FAULT TOLLERANCE)

Ma il DNS, alla fine della fiera, cosa e come memorizza?

Il DNS mette in associazione questi nomi con degli indirizzi, utilizzando una struttura di registrazione delle informazioni (chiamato *descrittore di risorsa* o RESOURCE RECORD) che ha questo aspetto:

www.ansa.it. 86400 IN A 194.244.5.206

- www.ansa.it. — NOME DI DOMINIO
- 86400 — "TIME TO LIVE" (86400 sono i secondi in un giorno)
- IN — CLASSE (IN = Internet)
- A — TIPO
- 194.244.5.206 — VALORE

Vediamo i vari campi singolarmente, partendo dal TTL

- Il Time To Live rappresenta un'informazione relativa alla "stabilità" del record
- Quando memorizziamo delle informazioni in un sistema distribuito, c'è il solito problema delle informazioni soggette a cambiamenti
- Il valore, in secondi, indicato il tempo entro cui si suppone che l'informazione resti stabile, quindi non sia necessario poi andarla ad aggiornare prelevandola dall'origine presso cui viene generata questa informazione

Importante, ricordiamoci che stiamo parlando di un sistema distribuito!

Quindi, per evitare del traffico inutile sulla rete, per le continue richieste della stessa informazione, queste ***informazioni vengono memorizzate temporaneamente su i DNS*** sparsi in giro per il mondo

Pertanto sarà necessario, periodicamente, ***aggiornare queste informazioni***

Il TTL dice ogni quanto è opportuno aggiornare tale COPIA LOCALE DELL'INFORMAZIONE

Vediamo gli altri 3 campi

- Il campo CLASSE del record, tipicamente è IN (oppure 1) per quanto riguarda i record INternet (esistono anche altre classi ma a noi non interessano, ad esempio Hesiod, ovvero sono indirizzi utilizzati solo al MIT)
- Il campo TIPO, nel caso di un indirizzo IP è A (ma esistono anche altri tipi, come vedremo nella prossima slide)
- Infine, il campo VALORE, che effettivamente rappresenta l'indirizzo IP del nodo cercato (o le altre informazioni nel caso degli altri TIPI)

I descrittori, però, possono essere di diversi tipi!

SOA	Start of Authority	Sono più campi che rappresentano i parametri per quella zona (Server dei nomi primario, indirizzo di posta del responsabile della zona)
NS	Nome del server	Quando si vuol sapere solo il Nome del server di dominio e non tutti gli altri parametri di zona si usa questo campo al posto di SOA
A	Address	Indirizzo IP dell'host
MX	Mail exchange	La priorità e il nome con cui il dominio desidera accettare la posta elettronica
CNAME	Canonical Name	Utilizzato per creare alias di nomi di dominio
PTR	Pointer	Alias per un indirizzo IP

Proviamo a capire meglio quali informazioni contengono questi tipi

- Il tipo SOA contiene un insieme di informazioni relative ai *parametri di funzionamento per la zona*
- Il tipo **NS** contiene solo alcune delle informazioni che restituisce il tipo SOA (in particolare *nome del server di dominio ed eventualmente il suo indirizzo di rete*)
- Il tipo MX contiene informazioni sul dominio di posta elettronica, quindi *come deve essere instradata la posta elettronica* (in particolare con la *possibilità* di avere *più alternative* nel caso in cui un server di posta elettronica non si accessibile)

Il tipo CNAME, ovvero un modo per creare degli ALIAS per un servizio

- Il tipo CNAME, contiene degli alias, ovvero permette di avere nomi diversi che si riferiscono allo stesso servizio/server (Es. Il CNAME di www.fbi.gov è fbi.edgesuite.net e sono entrambi lo stesso server WEB); non tutti i nomi di domini hanno associato un CNAME;
- Il tipo **PTR**, dato un indirizzo di rete restituisce il corrispettivo Nome di Dominio (o alias);

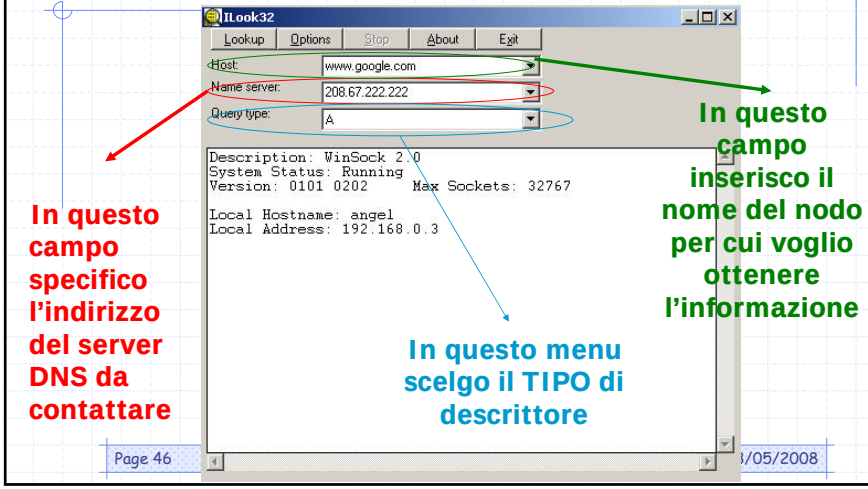
Come potete intuire il DNS è un sistema molto utile e complesso

Infatti, abbiamo la possibilità di descrivere una grande quantità di informazioni tramite questi ***server DNS***, che sono ***collegati l'uno con l'altro*** e accessibili dai client nel momento in cui hanno bisogno dell'informazione

Vediamo ora qualche esempio di come ci risponde il DNS quando noi lo interroghiamo!

- Utilizzeremo ilook32, un client DNS in ambiente Windows
- Questo applicativo ci permette, specificando il nome per cui cerchiamo l'informazione, di chiedere uno dei TIPI di descrittore che abbiamo visto nelle slides precedenti

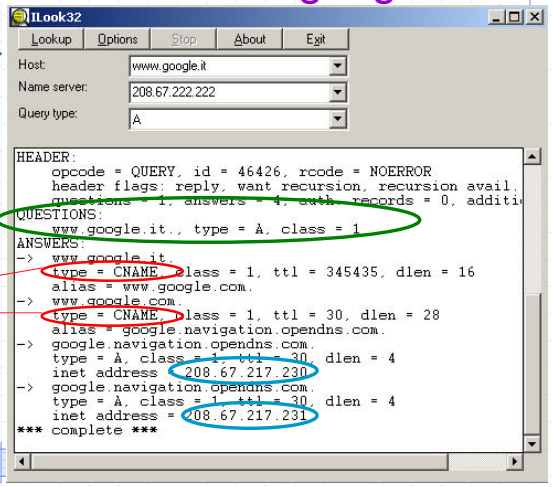
Vediamo com è fatto ilook32



Esempio 1: Proviamo a chiedergli l'indirizzo associato a www.google.it

La richiesta è stata www.google.it e type = A, quindi indirizzo, per una classe = 1 (ovvero Internet)

Oltre all'indirizzo troviamo anche una risposta sul tipo CNAME



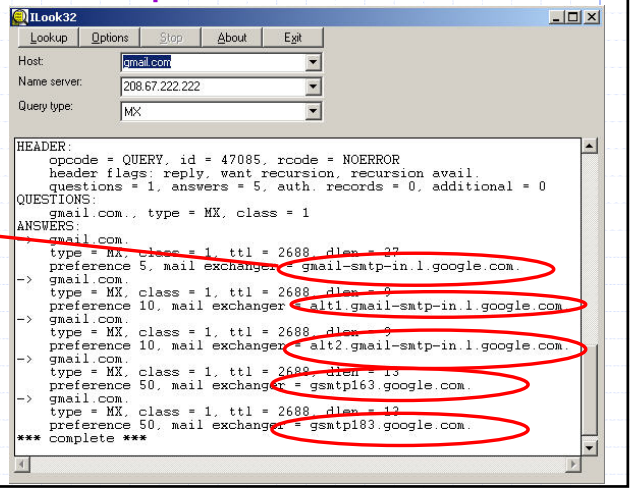
Perché un nome TANTI indirizzi di rete?

Perché www.google.it corrisponde a tanti indirizzi di rete?

Provate a pensarci! (Suggerimento: perché normalmente un italiano ha almeno 2 o più numeri di cellulare)?

Esempio 2: informazioni relativi a dei domini di posta elettronica

Con questa query abbiamo scoperto il NOME dei mail exchanger del dominio di posta gmail.com, ovvero dei server a cui ci si deve connettere per mandare email agli utenti di quel dominio

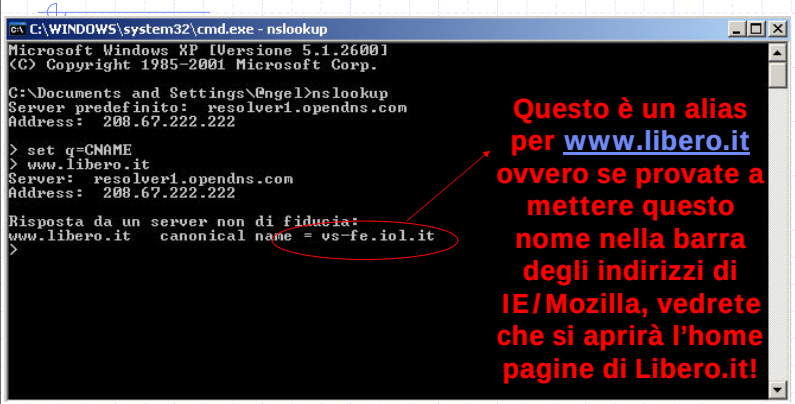


Nslookup: il client DNS di Windows!

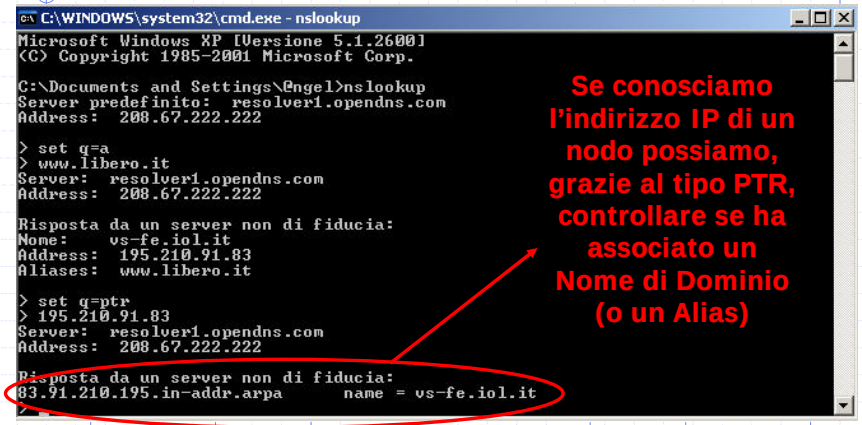
- Proviamo ad utilizzare un altro client DNS, quello messo a disposione da Windows
- E' un client testuale
- Per usarlo:
- Digitare nslookup da interfaccia MS-DOS
- Comandi da utilizzare:
  - Set q=TIPO
  - Server x.x.x.x (indirizzo di rete del server DNS che si vuol interrogare, altrimenti usa il server che è indicato nella impostazioni di rete di Windows)
- Lo stesso client c'è anche in Linux e si usa quasi nello stesso modo (provare per credere!)

Vediamo un'interfaccia d'esempio di nslookup

Esempio 3:vediamo un esempio di CNAME

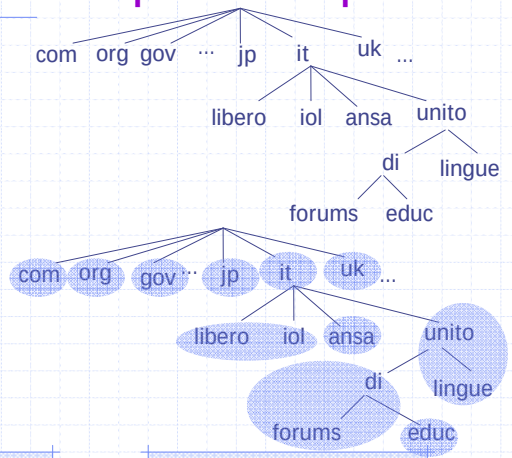


Esempio 4: il tipo PTR un modo per fare una query al contrario!

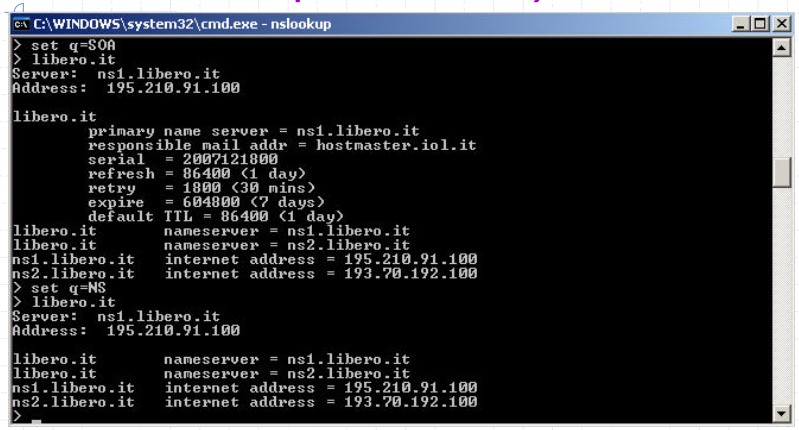


Se conosciamo l'indirizzo IP di un nodo possiamo, grazie al tipo PTR, controllare se ha associato un Nome di Dominio (o un Alias)

Esempio 5: il tipo SOA



Esempio 6: il tipo NS (una bagianata rispetto a SOA)



E se io voglio tutti in TIPI in una query sola?

Semplicemente si usa il tipo ALL e otterrò come risposta tutti i tipi che un server DNS può visualizzare rispetto al nome DNS fornito in INPUT (ovviamente quelli per cui non è stato definito un valore non vengono visualizzati!)

Es: set q=ALL

Ci manca ancora un componente per comprendere bene come funziona il DNS!

- La radice (root) della gerarchia dei nomi DNS è la zona . (punto), che è gestita da un insieme di server chiamati appunto ***root servers***
- I root servers **devono** conoscere il *nome e l'indirizzo dei server DNS di domini sottostanti* che sono sotto la loro autorità (Esempio: i root servers di .it devono conoscere tutti i server DNS che si occupano di gestire le varie zone dei sottodomini .it)
- ***Inoltre i root servers hanno il compito di reindirizzare le richieste relative a ciascun dominio di primo livello*** (per .it ad esempio le richieste per .com, .org, .de) ***ai server DNS propri di quel TLD (Top Domain Level).***

Ora abbiamo tutti gli strumenti per capire come funziona una risoluzione di nome

- In generale, per ottenere la risoluzione di un nome è necessario partire dalla radice, (1) interrogare uno dei root servers nel dominio di primo livello, ottenere il server che lo gestisce, (2) interrogarlo nel dominio di secondo livello, fino a (3) raggiungere il server autoritativo per il nome desiderato
- Questa tecnica è detta "***ricorsione***"
- Ecco a cosa serve il tipo SOA! Ovviamente è un tipo utilizzato dai server, e non dagli utenti normali!

Ma, io, utente normale, come utilizzo il DNS?

- Per utilizzare il servizio DNS, è necessario configurare su ciascun client uno o più server DNS di riferimento
- I server DNS sono predisposti a effettuare query ricorsive

1. Quando un ***nodo*** ha la necessità di comunicare con un altro nodo, ***chiede al server DNS di riferimento*** di effettuare il processo detto di ***risoluzione del nome*** in un indirizzo IP
2. Il ***server*** effettua una ***ricerca all'interno del suo database*** per ottenere l'indirizzo IP corrispondente al sistema ricercato
3. Se il ***server interrogato possiede l'informazione richiesta***, il processo di ricerca termina con l'***invio dell'indirizzo IP*** al richiedente
4. Se la ***ricerca*** ha esito ***negativo*** il server effettua una ***richiesta ricorsiva***

Piccoli aiuti per migliorare il sistema: il caching dei record DNS

- Alcuni ***server*** si prestano ad effettuare query ricorsive per conto di alcuni client
- Una volta che hanno ***ottenuto*** una ***risposta, memorizzano in una cache*** tutte le informazioni che hanno imparato, fino alla loro scadenza
- In questo modo diminuisce il traffico su rete e soprattutto la risposta è più rapida!

Ma il DNS usa TCP o UDP?

- Il servizio DNS può essere usato sulla porta 53 di UDP e di TCP
- Quando si usa UDP non si possono inviare datagrammi di dimensione superiore ad un valore specificato, molto minore della lunghezza massima di un datagramma UDP
- Perchè? Lo scopo di questa prescrizione è evitare che si formi eccessiva frammentazione che riduce l'affidabilità del trasporto

Se un server non riesce a fornire tutta la informazione necessaria nel datagramma UDP?

- Allora il server omette i dati che non può inviare e segnala che la risposta è stata troncata
- Se la risposta del server indica che c'è stato un troncamento, il client DNS riprova la domanda questa volta usando TCP

Alcuni semplici esempi!

- Ora vedremo dei file contenenti delle interazione fra un client (come nslookup) e vari server
- I file sono in formato tcpdump (nota utility di origine Unix), e possono essere esaminati usando qualunque software capace di "leggere" tale formato, quale ad esempio Wireshark (www.wireshark.com).

esempio1.eth: Ricerca di indirizzi per webmail.libero.it

In questo file si può notare che vengono ritornati più di un tipo A

Il DNS usato ritorna anche i nomi dei DNS server autorevoli, ma non i relativi indirizzi; questo comportamento non è ottimo, perché costringe il client a fare un'altra query se vuole contattare i server autorevoli

Se si esegue una seconda query, l'ordine con cui sono ritornati gli indirizzi è diverso; come mai?

esempio2.eth: e se contatto direttamente il DNS server di libero.it?

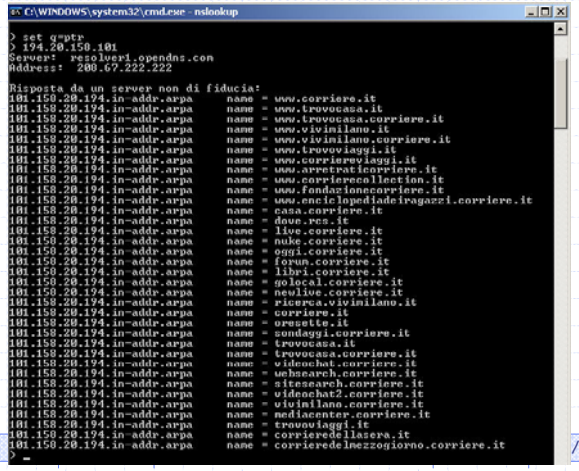
Se invece interroghiamo il DNS di libero.it possiamo avere l'indirizzo (Additional Records) di ambedue i DNS di libero.it

esempio3.eth: Ricerca del Name Server di libero.it effettuata sul name server di .it

Un DNS del dominio di .it risponde con gli indirizzi dei due DNS di libero.it, ma dice anche che la risposta non è autorevole, perché dal punto di vista di principio gli unici autorevoli per il dominio libero.it sono appunto i DNS del dominio

Però in questo modo chi cerca i DNS di un dominio li può scoprire e può contattarli per avere informazioni autorevoli

Esempio4.pcap: quando UDP non basta e bisogna usare TCP



Ma Internet, senza DNS, funziona lo stesso?

In teoria….SI

In realtà è tutto più complicato!

- Riscrittura programmi che usano al loro interno nomi invece di indirizzi (in questo modo il programma non deve essere cambiato se vario l'indirizzo di rete)
- E gli utenti chi li sente?!

Un po' di esercizi...

Utilizzando ilook32 o nslookup come client DNS e 151.99.125.2 come server iniziale:

1. Provare a cercare un nome non esistente; che risposta viene fornita dal server DNS? Analizzatela molto attentamente!
2. Ricerca del tipo SOA di libero.it. Che risposta viene fornita?
3. Ricerca del tipo SOA di libero.it utilizzando un server DNS di libero.it. Che risposta viene fornita?E' differente dalla precedente? Perché?

Catturare ed analizzare tutto il traffico generato utilizzando wireshark!

Altri esercizi ancora...

4. Ricerca del Name Server di libero.it effettuata sul name server di libero.it
5. Ricerca del Name Server di libero.it effettuata sul name server di .it. (Come si fa a scoprire chi è?)

- Che differenza c'è tra le due risposte?

Catturare ed analizzare tutto il traffico generato utilizzando wireshark!

Osservazioni sugli esercizi 4 e 5

- Come si vede un DNS del dominio di .it risponde con gli indirizzi dei due DNS di libero.it, ma dice anche che la risposta non è autorevole, perché dal punto di vista di principio gli unici autorevoli per il dominio libero.it sono appunto i DNS del dominio
- Però in questo modo chi cerca i DNS di un dominio li può scoprire e può contattarli per avere informazioni autorevoli
- ***I DNS del dominio "superiore" devono conoscere il nome e l'indirizzo dei DNS di domini sottostanti che sono sotto la loro autorità.***

Il sistema di posta elettronica

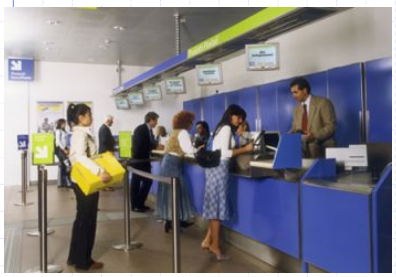
Dott. Libanore Luca

Da un servizio che l'utente usa incosapevolmente ad un servizio che vede da vicino!

La posta elettronica ha ***due aspetti particolarmente rilevanti*** per quanto riguarda il suo funzionamento e quello dei protocolli associati:

- Il formato dei messaggi, ovvero come devo essere strutturati i messaggi (FORMATO DEI MESSAGGI)
- Il trasferimento dei messaggi, ovvero come i messaggi devono transitare tra i vari server e poi arrivare al client dell'utente destinatario (REGOLE)

Alcune regole dei protocolli di posta ordinaria



1. Le lettere vanno consegnate all'ufficio postale o in un'apposita buca delle lettere
2. Se il destinatario della lettera è un utente dell'ufficio postale allora la consegna avviene direttamente
3. Se il destinatario NON è un utente dell'ufficio postale allora la lettera viene inoltrata all'ufficio postale competente
4. .....

Alcuni aspetti del formato dei messaggi del protocollo di posta ordinaria

DIMENSIONI (*)	PICCOLO	MEDIO
Altezza (mm)	120	250
Lunghezza (mm)	235	353
Spessore (mm)	5	25
Peso (g)	50	2000

Un'osservazione interessante: un servizio Internet e i suoi protocolli

Se per il protocollo DNS è facile fare confusione tra servizio e protocollo (servizio = risoluzione dei nomi, protocollo = DNS), il servizio di posta elettronica riesce a rendere netta la separazione tra servizio (la posta elettronica) e i protocolli che permettono di realizzarlo (SMTP, POP, IMAP)

Com'è il formato dei messaggi?

Il formato dei ricorda quello di una normale lettera

Un'email riproduce la struttura di una lettera comune

Introducendo con una serie di linee codificate con codice ASCII, informazioni assolutamente analoghe.

```
Delivered-To: luca@libanore.it
Date: Sat, 29 Dec 2007 12:44:25 +0100 (CET)
From: "Idalpaos@fastwebnet.it" <Idalpaos@fastwebnet.it>
Reply-To: "Idalpaos@fastwebnet.it" <Idalpaos@fastwebnet.it>
To: <luca@libanore.it>
Subject: R: Domanda veloce
X-Originating-IP: 28.255.124.86
X-Spam-Rating: mxavas6.fe.aruba.it 1.6.2 0/1000/N

Ho solo iniziato con la teoria, e ti lascerei tutta la parte pratica (con una
prova con voto!). SarÃ  un successo.
Ciao
Luisa
```


Come dovremmo riscivire una normale lettera, seguendo il formato delle e-mail

Busta

Nome: Libanore Luca
Via: Via Giuseppe Verdi, 13
Cap: 10113
Città: Pinerolo

Intestazione

Da: Pres. Napolitano Giorgio
Indirizzo: Palazzo del Quirinale 00187 Roma
Oggetto: Nomina a Cavaliere

Corpo

Gentile Dott. Libanore,
La presente per informarLa della Sua avvenuta nomina a Cavaliere di Gran Croce della Repubblica Italiana.
Cordialità,

Giorgio Napolitano

Nel formato elettronico i nomi dei campi devono ESSERCI!

Il formato di un messaggio elettronico è stato definito tra gli standard di Internet

Gli standard Internet sono definiti negli RFC (http://www.ietf.org/rfc.html)

In particolare l'RFC che si occupa del formato di un messaggio elettronico è l'RFC 822

Cosa dice l'RFC 822

Un messaggio elettronico deve rispettare le seguenti indicazioni:

- I campi (A, Oggetto, etc.) sono rappresentati da ***singole linee*** di testo in ASCII (quindi ad ogni campo corrisponde una linea e campi diversi stanno su linee diverse)
- Il formato dei campi è per tutti:
  - Nome del campo (Es. A)
  - Carattere ':'
  - Valore (se previsto)

A: luca@libanore.it

Sempre dall'RFC 822

Nell'RFC 822 non sono proprio distinte bene la parte di Intestazione dalla parte di Busta, anche possiamo identificare le due funzionalità nei diversi nomi dei campi

Delivered-To: luca@libanore.it
Date: Sat, 29 Dec 2007 12:44:25 +0100 (CET)
From: "ldalpaos@fastwebnet.it" <ldalpaos@fastwebnet.it>
Reply-To: "ldalpaos@fastwebnet.it" <ldalpaos@fastwebnet.it>
To: <luca@libanore.it>
Subject: R: Domanda veloce

Ancora dall'RFC 822

Il corpo del messaggio segue questa sequenza di campi, DOPO UNA RIGA BIANCA

Delivered-To: luca@libanore.it
Date: Sat, 29 Dec 2007 12:44:25 +0100 (CET)
From: "ldalpaos@fastwebnet.it" <ldalpaos@fastwebnet.it>
Reply-To: "ldalpaos@fastwebnet.it" <ldalpaos@fastwebnet.it>
To: <luca@libanore.it>
Subject: R: Domanda veloce
X-Originating-IP: 28.255.124.86
X-Spam-Rating: mxavas6.fe.aruba.it 1.6.2 0/1000/N

Intestazione + Busta

← Una riga bianca

Ho solo iniziato con la teoria, e ti lascerei tutta la parte pratica (con una prova con voto!). SarÃ un successo.
Ciao
Luisa

Il corpo del messaggio

Ma quali sono i principali campi dell'RFC 822?

To:	destinatario/i primario/i
Cc:	destinatario/i secondario/i (Copia per Conoscenza)
Bcc:	Copia per Conoscenza NON NOTIFICATA ai destinatari primari e secondari
From:	Mittente (Es. From: Luca Libanore)
Sender:	Indirizzo di posta elettronica del mittente (Es. Sender: luca@libanore.it)

Ma ci sono anche altri campi importanti nell'RFC 822

Received:	Linea aggiunta da ogni agente di trasferimento lungo il percorso (ci permette di tracciare qual è stato il percorso che ha fatto il messaggio dal mittente al destinatario)
Date:	Data e ora di invio del messaggio
Reply-To:	Indirizzo di posta elettronica a cui inviare le risposte (se questo campo non è presente, la risposta viene spedita all'indirizzo presente nel campo From:)

Questi sono gli ultimi campi che ci interessano!

Message-Id:	Identificatore (unico) del messaggio per futuri riferimenti
In-Reply-To:	L'identificativo del messaggio a cui si sta rispondendo
Subject:	Argomento del messaggio (una riga)

L'RFC 822 ha alcuni limiti, principalmente legati al fatto che la codifica è esclusivamente ASCII!

Per far fronte ai limiti dell'RFC 822 è stato introdotto MIME

MIME (Multipurpose Internet Mail Extension) *aggiunge delle regole per i messaggi che non sono ASCII*.

In particolare, grazie a MIME, possiamo spedire in un messaggio di posta elettronica:

- *Caratteri accentati (Es. è ò ù)*
- *Caratteri non alfabetici* (si pensi alle lingue dell'estremo oriente che utilizzano caratteri grafici) (es. 英)
- *Grafica* (Es. *Ecco un pò di grafica!*)
- *Contenuti diversi dal testo scritto* (es. Informazioni in formato multimediale)



Senza MIME, ovvero utilizzando solo il formato dei messaggi contenuto nel RFC 822, messaggi come questo non si potrebbero spedire!

To: Luca Libanore <luca@libanore.it>
From: Luca Libanore <libanore@gmail.com>
Subject: Messaggio di prova
Cc:
Bcc:
Attached:

Ciao a tutti,
non è forse un bellissimo tramonto?!

:008

Ma come avviene il trasferimento dei messaggi dal mio computer a quello di un mio amico?

Il trasferimento dei messaggi di posta elettronica si realizza:

1. Utilizzando SMTP (Simple Mail Transfer Protocol) per il trasferimento dei messaggi tra i server di posta elettronica
2. Utilizzando POP3 (Post Office Protocol versione 3) o IMAP (Internet Message Access Protocol) per il trasferimento dei messaggi dal server di posta al client

Vediamo un pò di caratteristiche del protocollo SMTP

Il protocollo SMTP risponde sulla porta 25.

È un protocollo di tipo ASCII, ovvero il ***dialogo fra i vari server*** di posta elettronica avviene ***tramite comandi di tipo ASCII***

I comandi, e altre caratteristiche di questo protocollo, si possono trovare nel RFC 821.

Ecco alcuni dei comandi ASCII che si scambiano i server di posta

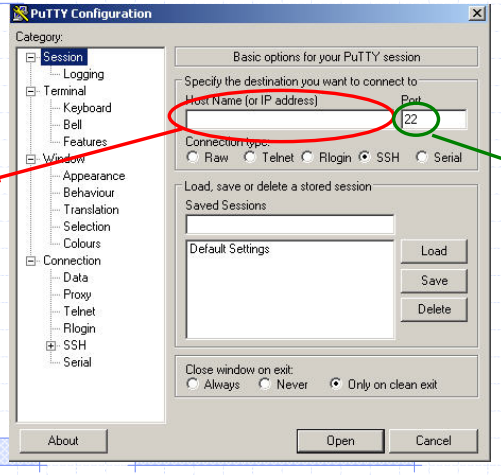
HELO	Dal client al server seguito dall'indirizzo DNS del client (è un abbreviazione dell'HELLO)
MAIL FROM:	Nella composizione di un nuovo messaggio indica il mittente (user@mailserver)
RCPT TO:	"recipient to": destinatario del messaggio (una linea RCPT TO per ogni destinatario)
DATA	Precede il messaggio vero e proprio, comprensivo di busta, intestazione e corpo
QUIT	Chiude la connessione

Vediamo ora come si comporta l'SMTP per la spedizione di un messaggio di posta elettronica

- Per provare il funzionamento di questo protocollo ASCII utilizzeremo l'applicativo TELNET, chiamando però il server non sulla normale porta telnet ma sulla porta 25
- A tal fine utilizzeremo PuTTY, un client telnet/ssh grafico gratuito, utilizzabile sia in piattaforma Windows che Unix (www.chiark.greenend.org.uk/~sgtatham/**putty**/)

Com'è fatto PuTTY?

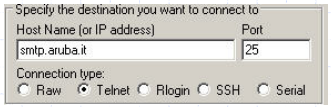
In questo campo si inserisce il nome, o indirizzo IP, del server da contattare



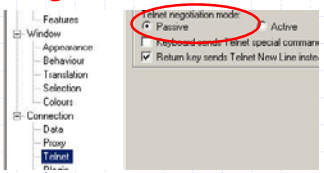
In questo campo si inserisce la porta del server; PuTTY può essere utilizzato con tutti i protocolli ASCII (SMTP, POP3, etc.)

Inseriamo le impostazioni in PuTTY

Server SMTP: smtp.aruba.it su porta TCP 25

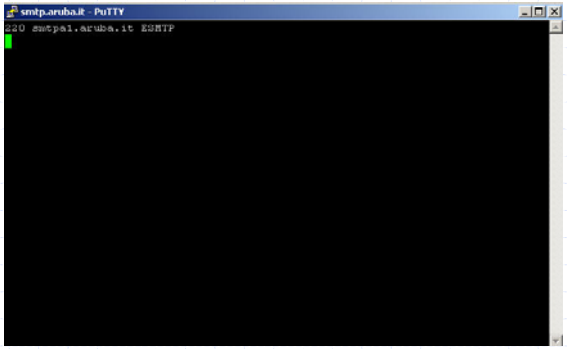


Piccola raccomandazione: configurare come passiva la *negoziazione* Telnet



E una volta stabilita la connesione…

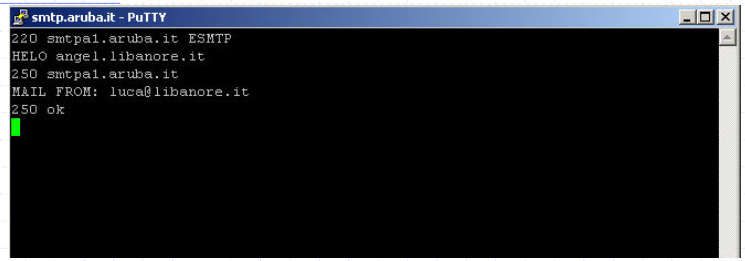
Il server SMTP si mette in attesa!



Noi lo salutiamo dicendogli il nome della nostra macchina (o quantomeno l'indirizzo di rete!)

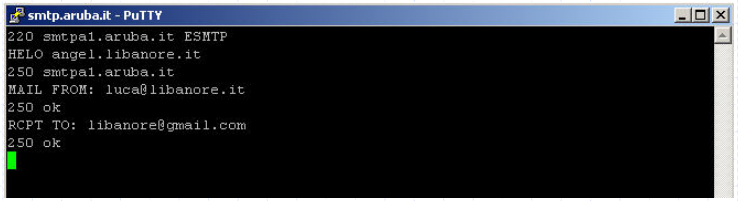
- Questo è il primo comando da fornire al server SMTP per iniziare una sessione di posta elettronica
- Una volta riconosciuti (alcuni server SMTP non permettono di continuare la sessione se non riconoscono la macchina o IP) possiamo creare un nuovo messaggio

Il primo comando per creare un nuovo messaggio è MAIL FROM:



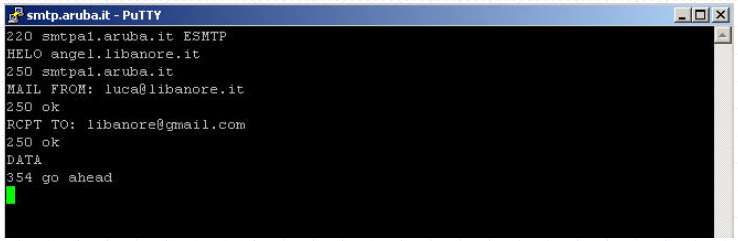
- In questo modo ho specificato il mittente del messaggio, ora andiamo a inserire il destinatario

Il comando RCPT TO:



Molti server SMTP non accettano la creazione di un nuovo messaggio di posta elettronica se il destinatario non è un loro cliente! Perchè? Lo vedremo dopo!

A questo punto possiamo utilizzare il comando DATA per creare il corpo del messaggio



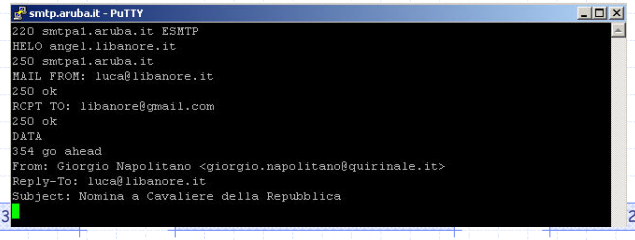
Inizieremo specificando alcuni campi, ovvero quelli presenti NEL RFC 822

Alcune osservazioni sul corpo del messaggio e sulla sua interpretazione

L'indirizzo a cui verrà spedito il messaggio è ormai già stato costruito nel comando RCPT TO: ma ***adesso avremo una serie di campi che verranno interpretati dal client*** che riceverà il messaggio, per spiegare all'utente da dove arriva il messaggio, qual è il soggetto e così via.

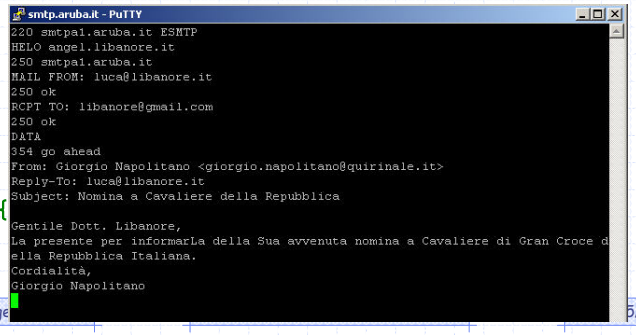
Utilizzando i campi specificati nel RFC 822, iniziamo a creare il messaggio, partendo dalle intestazioni!

- Per prima cosa l'indirizzo del mittente: per esempio scriveremo che il nostro indirizzo mittente sarà sergio.napolitano@quirinale.it bleffando per ingannare il destinatario!
- Poi possiamo dare un subject che ci ritorna utile (la nostra famosa nomina!)

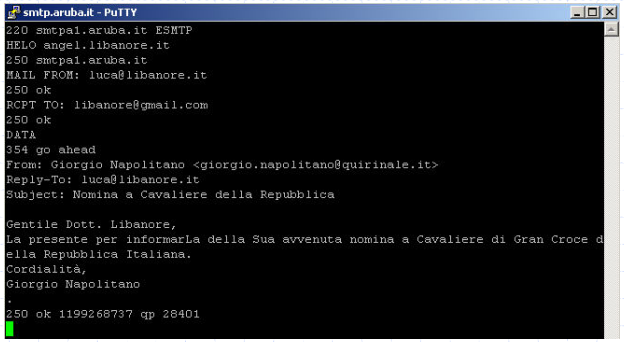


Ora concentriamoci sul corpo del messaggio vero e proprio!

Ora passiamo a scrivere alcune righe del messaggio, ricordandoci però che dobbiamo inserire ***una riga vuota*** tra le intestazioni e il corpo del messaggio!

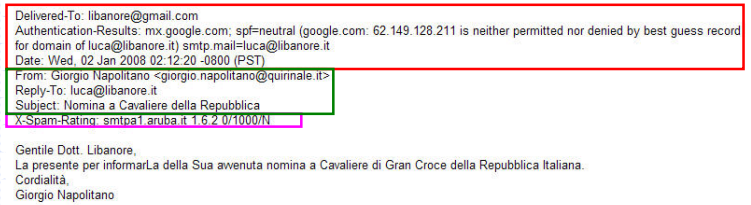


Concludiamo il messaggio con un . e via!



Il server ci dice che il messaggio è stato accettato e lo invierà!

Vediamo ora il nostro messaggio arrivato a destinazione!


Delivered-To: libanore@gmail.com
Authentication-Results: mx.google.com; spf=neutral (google.com: 62.149.128.211 is neither permitted nor denied by best guess record for domain of luca@libanore.it) smtp.mail=luca@libanore.it
Date: Wed, 02 Jan 2008 02:12:20 -0800 (PST)
From: Giorgio Napolitano <giorgio.napolitano@quirinale.it>
Reply-To: luca@libanore.it
Subject: Nomina a Cavaliere della Repubblica
X-Spam-Rating: smtpa1.aruba.it 1.6.2 0/1000/N

Gentile Dott. Libanore,
La presente per informarLa della Sua avvenuta nomina a Cavaliere di Gran Croce della Repubblica Italiana.
Cordialità,
Giorgio Napolitano

Come vediamo il mittente, per il client di posta, è proprio quello che abbiamo creato noi!

Vediamo una serie di intestazioni: *alcune sono quelle che abbiamo creato noi* (Reply-To, From e Subject) mentre *altre sono state aggiunte dai vari mailer attraversati* (*come lo stesso server di Aruba*)

Ma questa tecnica funziona con tutti i server SMTP? Purtroppo NO, anzi..

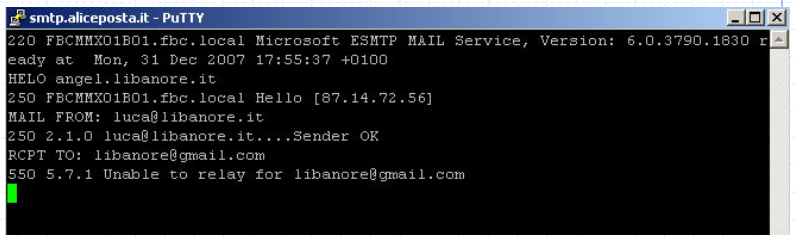
Le risposte dei server SMTP potrebbero essere le più svariate:

- Non posso perché il tuo indirizzo IP non è uno di quello autorizzati ad usare i miei servizi
- Non posso perché il ***Relaying*** (cos'è?) non è concesso!

Tutti questi problemi sono sorti da quando esiste il fenomeno dello *spamming*

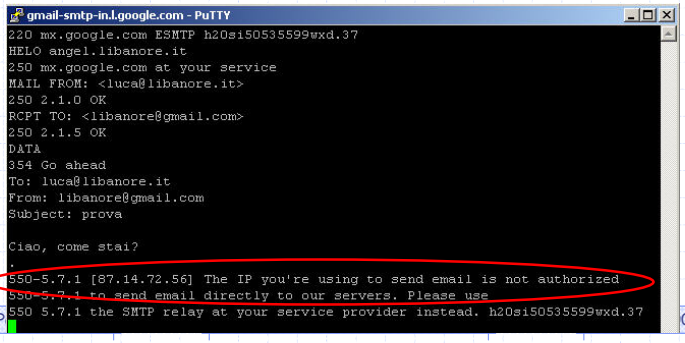
Al giorno d'oggi i server SMTP prima di accettare una e-mail da consegnare effettuano molti controlli (per evitare di finire in blacklist!)

Esempio di Relaying non permesso utilizzando il server SMTP smtp.aliceposta.it



Non posso offrirti il servizio perché il tuo indirizzo IP NON LO CONOSCO

E se proviamo a spedire un'email utilizzando un server SMTP di gmail? (ad esempio gmail-smtp-in.l.google.com)



I server SMTP di libero (MA NON SOLO) sono proprio maleducati!

Provate a fare un tentativo con un server SMTP di libero (ad esempio mxlibero1.libero.it): se non siete clienti di libero vi chiude direttamente la connessione in faccia!

Ora vediamo come avviene il trasferimento dei messaggi dal server al client!

Fino ad ora abbiamo visto:

- Il formato dei messaggi di posta elettronica (RFC 822 e MIME)
- Il trasferimento dei messaggi *dal client mittente verso il suo server di posta di riferimento* (protocollo SMTP)
- Il *trasferimento del messaggio tra i vari server di posta elettronica attraversati dal messaggio stesso* (protocollo SMTP)

Ora vedremo come avviene il trasferimento nell'ultimo passaggio, ovvero tra server e client destinatario

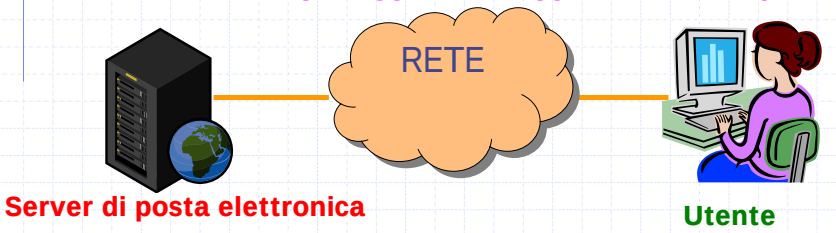
Un utente si collega tramite Internet ad un server di posta elettronica

Esiste un'esigenza degli utenti di utilizzare il sistema di posta elettronica in questo modo:

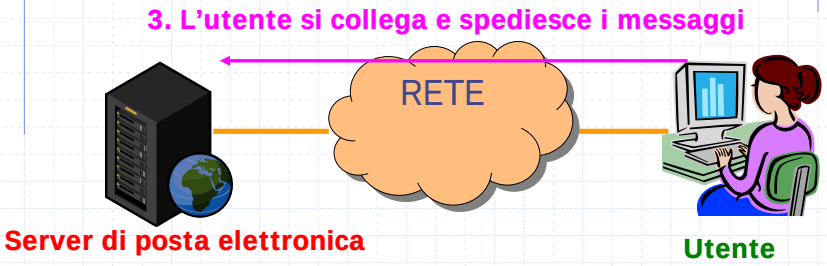
1. L'utente si collega e "scarica" i messaggi ricevuti

Dopo "aver scaricato" i messaggi

2. L'utente si scollega, legge i messaggi e scrive le risposte



Infine l'utente potrà ricollegarsi per spedire i nuovi messaggi



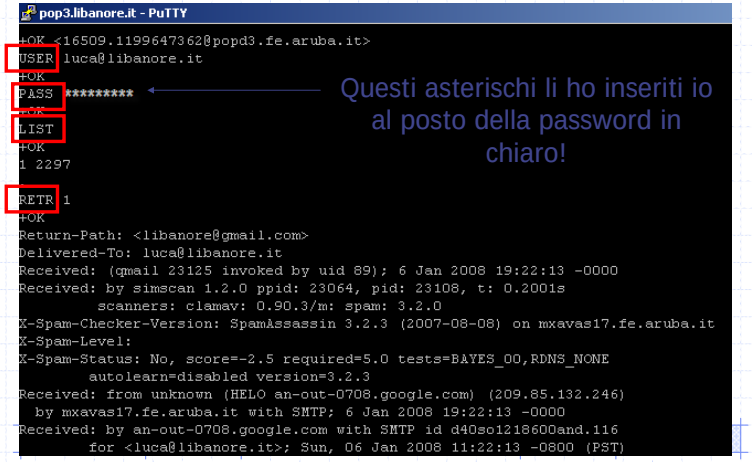
Esiste ancora l'esigenza di lavorare in locale?

- La necessità per gli utenti di scollegarsi e lavorare in locale, deriva dal ***1) non voler impegnare la linea telefonica*** (nel caso di collegamenti con linee tradizionali, quindi non ISDN, ADSL o Fibbra ottica) e ***2) di pagare di meno, visto che il costo del collegamento è a tempo***
- Queste esigenze sono ancora molto sentite dagli utenti standard, che non possiedono abbonamenti di tipo FLAT o non hanno effettuato il passaggio a linee a banda larga

Quali protocolli vengono utilizzati per trasferire i messaggi dal server al client?

- Il primo, e più conosciuto, è POP3 (Post Office Protocol)
- ***POP3*** permette di effettuare queste ***operazioni*** con una serie di ***comandi*** sempre di tipo ***ASCII***, abbastanza simili a quelli del SMTP (USER, PASS, LIST, RETR, DELE, QUIT)
- POP3 rimane in attesa sulla porta TCP 110
- Sebbene sia il protocollo più diffuso per il trasferimento dei messaggi dal server di posta al client non è l'unico: vediamo ora IMAP

Un esempio del funzionamento del protocollo POP3 (utilizzando telnet)



Questi asterischi li ho inseriti io al posto della password in chiaro!

IMAP (Interactive Mail Access Protocol)

- Anche questo protocollo serve per il trasferimento dei messaggi dal server di posta elettronica al client
- È un protocollo più sofisticato del POP3 ed ha una funzionalità un pò differenti
- *POP3: normalmente scarica i messaggi, copiandoli localmente sul personal computer dell'utente*, e li cancella dal server (per evitare problemi di spazio)
- *IMAP: mantiene i messaggi presso il server* (esiste anche la possibilità di scaricarli in locale)

Perché è utile mantenere i messaggi sul server?

- *Il protocollo IMAP consente all'utente di accedere ai messaggi*, anche quelli già letti in precedenza, *da qualsiasi macchina in rete*
- Mantenendo i messaggi sul server *consente di ricollegarsi tramite altri calcolatori sempre allo stesso server e vedere i vecchi messaggi*

Osservazione: ma i client permettono di mantenere i messaggi sul server anche con POP3!

- Verissimo! Ma normalmente, se ci fate caso, si utilizza un server POP3 differente da un server IMAP!
- Infatti, *un server POP3 sarà implementato* e configurato *partendo dalla considerazione che* la maggior parte degli *utenti scaricherà i messaggi in locale* liberando lo spazio sul server
- *Un server IMAP invece viene implementato* (anche a livello di risorse) avendo ben presente che la sua caratteristica risiede proprio *nel lasciare i messaggi sul server remoto*
- POP3 non è stato pensato per lasciare i messaggi sul server!Questa opzione è stata implementata a livello software, non è nativa nel protocollo, e come tutte le cose improvvisate è poco funzionale (e spesso poco funzionante)!

Inoltre IMAP...

Caratteristiche di IMAP ma non di POP3:

- **Supporto all'accesso a singole parti MIME di un messaggio**: IMAP permette di scaricare una singola parte MIME o addirittura sezioni delle parti, per avere un'anteprima del messaggio o per scaricare una mail senza i file allegati.
- **Accesso a molteplici caselle di posta sul server**: con il protocollo IMAP si possono creare, modificare o cancellare mailbox (di solito associate a cartelle) sul server.

Ancora su IMAP...

- **Possibilità di fare ricerche sul server**: IMAP permette al client di chiedere al server quali messaggi soddisfano un certo criterio, per fare, per esempio, delle ricerche sui messaggi senza doverli scaricare tutti.
- **Password criptate**: Con il protocollo POP le password vengono solitamente inviate in testo, rendendo facile, con una intercettazione, l'individuazione della password; con IMAP è possibile criptare la password, anche se server e client devono trovare un accordo sul metodo.
- Porta TCP utilizzata: 143

Un *"altro sistema"* per realizzare la posta elettronica: il webmail

Una Webmail è un'***applicazione web*** che permette di ***gestire*** un account di posta elettronica ***attraverso un browser***

Attraverso l'interfaccia grafica si ***stabilisce*** una normale ***connessione*** ***verso un server di posta*** SMTP, IMAP o POP3

Generalmente si utilizza IMAP per la sua struttura che consente l'accesso a molteplici caselle di posta sul server (creare, cancellare mailbox)

Vantaggi del Webmail

- Possibilità di *leggere la propria posta ovunque* vi sia una connessione ad internet
- I *messaggi non necessitano di essere scaricati*
- Le ***caselle di posta*** possono essere ***amministrate*** (create, modificate, cancellate) molto ***facilmente***
- ***Riconoscimento di spam e virus informatici*** (se offerto dal gestore!)

Svantaggi del Webmail:

- È ***richiesta una connessione sia per la visualizzazione che per la composizione*** dei messaggi
- Una ***connessione lenta influenza*** la ***funzionalit***à generale della webmail
- Le ***funzionalità di composizione*** di ***messaggi*** sono generalmente ***limitate per la formattazione*** di un messaggio