Reti di Calcolatori

Appunti delle Lezioni di Reti di Calcolatori

Anno Accademico: 2024/25

Giacomo Sturm

Dipartimento di Ingegneria Civile, Informatica e delle Tecnologie Aeronautiche
Università degli Studi "Roma Tre"

Sorgente del file LaTeX disponibile al seguente link:
https://github.com/00Darxk/Reti-di-Calcolatori/

Indice

1 Introduzione

Una qualsiasi interconnessione di calcolatori può rappresentare una rete di calcolatori, ma in base alla distanza reciproca tra questi componenti si tratta di reti differenti. Convenzionalmente si considerano reti di calcolatori, sistemi di calcolatori interconnessi ad una distanza superiore ai 50 cm. Una distanza minore, fino ai 5 cm, generalmente interessa componenti dello stesso computer, sulla stessa scheda madre, connesse tra di loro; mentre una distanza inferiore ai 5 cm rappresenta componenti sullo stesso chip. Inoltre le reti considerate possono essere ulteriormente divise in base alla distanza dei loro elementi:

- Se hanno una distanza minore di 5 km, si tratta di risorse connesse sulla stessa rete o edificio, o su edifici vicini. Questo tipo di rete si chiama Local Area Network (LAN);
- Se hanno una distanza superiore ai 5 km, si tratta di risorse connesse su una vasta area geografica. Questo tipo di rete si chiama Wide Area Network (WAN).

Tra questi due livelli possono essere presenti anche tecnologie molto diverse tra di loro, queste tecnologie vengono identificate da acronimi da cui è possibile ricavare lo scopo della tecnologia, senza tuttavia conoscere il suo funzionamento.

Una connessione tra componenti di una rete coinvolge sempre uno scambio di informazioni, tramite uno scambio di messaggi in serie. Gli elementi della rete effettuano degli accessi ad essa apparentemente in parallelo e simultanei, per poter comunicare tra di loro. Mentre su componenti sulla stessa macchina o sullo stesso chip avvengono tramite accessi ad una memoria condivisa.

Le connessioni tra componenti di una rete avvengono su uno strato fisico, quindi attraverso diversi mezzi trasmissivi, i quali non verranno analizzati approfonditamente a questo livello di astrazione. Tra i più comuni mezzi trasmissivi abbiamo cavi in fibra ottica, o in rame, ed onde radio.

1.1 Commutazione

Una rete di calcolatori può essere rappresentata come un grafo composto da vari nodi, per realizzare tutte le possibili coppie di calcolatori che potrebbero comunicare tra di loro attraverso la rete. Ma se venissero collegati individualmente tutte le possibili coppie di calcolatori necessiterebbe di infrastrutture massicce, poiché il numero dei possibili percorsi aumenta quadraticamente rispetto all'aumento dei calcolatori della rete. Infatti avendo n, tutte le possibili combinazioni tra questi calcolatori sono $n(n-2)/2$, nel caso ognuna di queste coppie corrisponda ad una connessione differente, il costo di costruzione e gestione della rete sarebbe eccessivo.

Per risolvere questo problema e diminuire il numero totale di connessioni nella rete si utilizza il meccanismo della commutazione. Questo termine risale alla telefonia, dove ai sorse lo stesso problema, risolto introducendo centralini intermedi dove si potevano collegare diverse area telefoniche contenenti i telefoni che tentavano di comunicare. In questo modo si può drasticamente diminuire il numero di connessioni individuali nella rete, e non bisogna integrare un numero elevato di connessioni all'aggiunta di un singolo elemento. Si indica quindi con commutazione di circuito questo meccanismo di creare una connessione fisica tra due calcolatori, connettendo diverse zone della rete attraverso nodi intermedi. L'interazione tra computer consiste intrinsecamente da grandi

quantità di dati trasmessi velocemente a grandi distanze, per cui hanno bisogno di infrastrutture dedicate massicce, si preferisce quindi questo sistema di nodi intermedi, nonostante non consenta di soddisfare contemporaneamente tutte le coppie di calcolatori.

Poiché questa grande quantità di dati deve attraversare la rete velocemente, si utilizza una diversa tecnica di comunicazione a livello dei singoli messaggi, dividendoli in pacchetti da spedire separatamene. Nella commutazione a datagramma questi pacchetti vengono spediti su linee anche diverse e si mescolano a tutti i pacchetti che attraversano quel percorso. Le linee non sono quindi ad uso esclusivo di una singola connessione. Ma per ricomporre il messaggio originale bisogna combinare questi pacchetti nello stesso ordine in cui sono stati separati, sono necessari dati aggiuntivi per poter riconoscere il loro ordine, perso durante la trasmissione. La distanza attraversata da ciascun pacchetto infatti non è garantito sia uguale.

Esiste inoltre un altro tipo di commutazione a circuito virtuale, dove i pacchetti vengono inviati sullo stesso percorso sequenzialmente, ed ogni linea può essere condivisa da un altro circuito virtuale, quindi non sono ad uso esclusivo. In questo caso invece è necessario un meccanismo per poter distinguere tra di loro questi circuiti virtuali sulla stessa linea fisica.

Si è risolto tramite la commutazione di pacchetto l'esclusività delle linee della rete, introdotta dal modello a commutazione di circuito.

La rete internet moderna utilizza la commutazione a datagramma, per motivi economici e gestionali. Altrimenti sarebbe necessario un gestore della rete che deve trovare un percorso ed attribuirlo ad una coppia ad ogni tentativo di connessione. Data la complessità della rete moderna, lasciare che i pacchetti trovino il percorso autonomamente è la scelta più efficiente. Per realizzare una commutazione a datagramma, piccole aree geografiche diverse vengono coperte da "Internet Service Provider" (ISP) differenti che possono comunicare solamente con altri ISP adiacenti. Quindi all'invio di un pacchetto, se il destinatario non è nella stessa zona dell'ISP corrente, questo lo invia ad un ISP adiacente che crede possa contenere il destinatario, così anche per la ricezione da un altro ISP. In caso il destinatario sia nella zona dell'ISP corrente, questo lo trasmette a lui. Accordi possono essere stipulati da chiunque a chiunque, un ISP ha sempre la necessità di trasmettere i pacchetti attraverso la rete.

In certi casi l'ISP può gestire la rete a circuito virtuale, se sia il destinatario che il mittente siano coperti dal singolo ISP.

1.2 Velocità

Nelle reti LAN e WAN si possono trasmettere dati a velocità diverse:

- LAN: velocità tra 10 ai 100 Mb/s;
- WAN: velocità tra 64 Kb/s ai 200-400 Mb/s.

Le reti WAN presentano hanno molti livelli di retrocompatibilità mantenuti, per cui si possono trasmettere dati a velocità minori di una rete LAN. In generale sono sempre richieste reti a velocità di trasmissione elevata, e connessioni ad alta velocità.

Data una rete si può definire la velocità in due modi differenti. Se si considera il tempo in cui il primo bit del messaggio arriva a destinazione. Le connessioni ad alta velocità vengono realizzate in

linee a fibra ottica, per cui i bit vengono inviati come impulsi di luce, e viaggiano ad una velocità costante, quindi per ogni rete la velocità di trasmissione di un singolo bit è la stessa. Si definisce quindi il tempo di ritardo o delay, il tempo per trasmettere un singolo bit, alla velocità della luce, sulla rete e dipende interamente dalla distanza.

Un pacchetto non viene rappresentato da un singolo bit, per cui non possono essere trasmessi alla stessa velocità, si definisce banda la quantità di bit trasmessi contemporaneamente sulla linea. Questa si chiama banda, e generalmente è sempre possibile comprare più banda in modo relativamente facile, ma è molto difficile comprare meno delay.

1.3 Gestione delle Risorse

La rete è essenzialmente un insieme di risorse interconnesse tra di loro e dalla teoria dei sistemi operativi, il loro controllo può essere descritto da varie attività:

- Verifica dei diritti d'accesso;
- Sequenziamento degli accessi alla risorsa;
- Esecuzione delle operazioni disponibili.

Ad ogni risorsa vengono assegnato almeno un gestore, di numero variabile in base al tipo di gestione. Le modalità di gestione delle risorse sono varie, si dividono in gestione autocratica e multilaterale. Nella gestione autocratica ogni risorsa ha un unico gestore associato ed univoco. Nella gestione multilaterale per ogni risorsa può esserci più di un gestore, si possono identificare quindi tre sottotipi di questa gestione:

- Gestione partizionata, dove attività di gestione viene effettuata da un singolo processo;
- Gestione successiva, dove tutte le attività di gestione vengono effettuate a turni da più processi;
- Gestione replicata, dove tutti i gestori partecipano a ciascuna attività, se ogni gestore ha peso decisionale uguale allora si tratta di gestione democratica.

La gestione replicata fornisce una forte resistenza ai guasti per un numero elevato di gestori che partecipano a ciascuna istanza di una attività, con alto grado di uguaglianza nella responsabilità di gestione. Sono abbastanza diffusi meccanismi di elezione per la scelta dei gestori.

2 Kathará

Le reti di calcolatori sono complicate, comprendono vari dispositivi, tra cui computer e router. Tante interfacce e protocolli diversi, collegati attraverso interconnessioni fisiche. Realizzano delle strutture topologiche molto vaste e complesse.

Si vuole sperimentare con reti anche molto complesse, senza utilizzare dispositivi fisici. Anche se si ha a disposizione una rete reale, sarebbe difficile convincere un provider a fornire a costi non eccessivi la loro rete per effettuare esperimenti su di essa. Realizzare una rete esclusivamente per effettuare questi esperimenti allo stesso modo si rivelerebbe estremamente costoso.

2.1 Introduzione

Kathará è un framework basato su container per effettuare esperimenti su reti di calcolatori, è un progetto open-source su github, per cui ognuno è in grado di contribuire allo sviluppo. Permette di effettuare emulazione di rete, differente da simulazione di rete. Negli strumenti di simulazione di un sistema si vogliono riprodurre le prestazioni di un sistema reale, la sua latenza, il ritardo, gli errori, la perdita di pacchetti, etc. Non si considerano le sue funzionalità, vengono analizzati solamente le prestazioni ed i parametri specifici della rete. Invece l'emulazione mira a riprodurre accuratamente le funzionalità offerte dal sistema, senza limitazioni di prestazioni. In questo caso le prestazioni rappresentano aspetti marginali della rete.

Per emulare una rete si utilizza una macchina in funzione da host, all'interno della quale vengono eseguiti container autonomi. Essenzialmente macchine isolate, queste vengono poi collegate da fili virtuali, in modo da simulare la presenza di una rete fisica, nonostante sia effettuata su una singola macchina. In Kathará i container sono collegati tra di loro in domini di collisione virtuali, non fili virtuali, per motivi di praticità. Questo consente di collegare l'host ad ogni dominio di collisione, per poter analizzare il comportamento su tutta la rete. Utilizzando fili virtuali invece non sarebbe stato così semplice da implementare.

Ciascuno dei container viene configurato come un dispositivo, in principio sono tutti uguali, ma possono rappresentare computer, router, switch, etc. Per rappresentare tutti gli elementi di una rete realisticamente.

I container rappresentano una virtualizzazione leggera, non sono macchine virtuali, quindi viene emulato un altro sistema operativo al loro interno, ma solamente un'applicazione o una piccola parte di un sistema operativo. Sono quindi molto leggeri, con un tempo di avvio ridotto rispetto alle macchine virtuali, utilizzate generalmente per realizzare micro-servizi che godono di vita autonoma.

Per Kathará viene utilizzato il sistema Docker, il più ampiamente utilizzato a livello globale per realizzare virtualizzazioni leggere. Per realizzare un container è necessaria un'immagine, ovvero un'insieme di software e le loro librerie e binari, necessari alla loro esecuzione, statici. Sono disponibili diverse immagini per realizzare dispositivi diversi, ma in questo corso verrà utilizzata solamente l'immagine di base presente in Kathará. Tramite quest'immagine è possibile realizzare un computer, un router, un bridge ed è possibile inserire al suo interno applicazioni di tipo server, servizi web, etc. Tutti gli elementi necessari per effettuare esperimenti di rete in questo corso. Ogni container poiché rappresenta un'esecuzione isolata di un'immagine, può essere costruiro su

un'immagine diverse, oppure di versione diversa dagli altri container in esecuzione, senza influire sul funzionamento di Kathará.

Un dispositivo si presenta con un terminale, su cui possono essere eseguiti comandi specifici a quel dispositivo, una memoria dedicata, un filesystem e zero o più interfacce di rete. Ogni interfaccia di rete è connessa ad un unico dominio di collisione. Tutte le interfacce trattate rispetteranno il protocollo 4.3 per comunicare tra di loro.

Kathará dispone di tre tipi di comandi:

- "v-commands": utilizzano il carattere `v` come prefisso, e sono comandi di basso livello per configurare ed avviare un singolo dispositivo;
- "l-commands": utilizzano il carattere `l` come prefisso, e permettono di gestire un intero "lab", configurandolo ed avviandolo;
- "Global commands": sono comandi principalmente di gestione.

I v-commands sono:

- `vstart`: comando di avvio di un dispositivo;
- `vconfig`: aggiunge un file di configurazione ad un dispositivo attualmente in esecuzione;
- `vclean`: termina l'esecuzione di un dispositivo.

Gli l-commands sono:

- `lstart`: comando di avvio di un lab;
- `lconfig`: permette di effettuare operazioni di configurazione su un dispositivo di un lab già attivo;
- `lclean`: termina l'esecuzione del lab;
- `lrestart`: termina e riavvia tutti i dispositivi del lab;
- `linfo`: fornisce informazioni sul lab.

I global commands sono:

- `check`: controlla l'ambiente del sistema, per controllare che l'installazione è andata a buon fine;
- `connect`: permette di collegarsi ad una macchina di Kathará già attiva;
- `list`: mostra tutti le macchine di Kathará in esecuzione per l'utente corrente;
- `settings`: mostra e configura le opzioni di Kathará;
- `wipe`: elimina tutte le macchine di Kathará, le loro connessioni ed eventuali opzioni.

Per testare il funzionamento di Kathará dopo l'installazione è consigliabile utilizzare i seguenti comandi:

```
> kathara check
```

Per controllare il suo corretto funzionamento si può testare la creazione di una singola macchina:

```
> kathara vstart -n pc1 --eth 0:A
```

Si crea una macchina di nome `pc1`, tramite l'opzione `-n` e si connette al dominio di collisione `A` con l'opzione `--eth 0:A`, che specifica anche il tipo di connessione virtuale, in questo caso ethernet. Se non vengono sollevati errori dopo questi comandi, si può interrompere la sua esecuzione con:

```
> kathara vclean -n pc1
```

Un lab di Kathará è un insieme di dispositivi che possono essere avviati e terminati contemporaneamente. La loro struttura consiste in una directory principale del lab, sempre contenente il file `lab.conf`, dove viene descritta la topologia della rete. Se sono è necessario, sono presenti subdirectories dove vengono specificate le configurazioni per i singoli dispositivi. Inoltre per ciascun dispositivo è ipotizzabile la presenza di un file chiamato con il nome del dispositivo di tipo `.startup` dove vengono descritte le operazioni da effettuare dal dispositivo all'avvio.

Negli esercizi ed in sede d'esame non sarà richiesto di realizzare un lab, ma si dovrà analizzare il comportamento di lab preesistenti, agendo sulle reti ed in caso correggendo eventuali errori o bug.

Il file `lab.conf` descrive la topologia della rete ed i dispositivi che devono essere avviati all'avvio del lab. Contiene istruzioni di sintassi:

```
<machine>[<arg>]=<value>
```

Dove `<machine>` è il nome della macchina su cui si vuole effettuare una certa operazione, `<arg>` è il tipo di operazione da effettuare su quella macchina. Se questo argomento è un numero indica a quale interfaccia della macchina si riferisce l'assegnazione, ovvero connessione ad un certo dominio di collisione, di valore specificato dal termine `value`.

Per avere un dispositivo nel lab, questo deve essere citato nel file di configurazione. Non è possibile riferirsi ad un'interfaccia successiva, se non si è già assegnata la sua precedente. Gli unici caratteri consentiti per definire il nome di una macchina o il nome di un dominio di collisione sono caratteri alfanumerici, dove lettere maiuscole e minuscole vengono considerate uguali.

I filesystem di tutte le macchine sono indipendenti ed isolati dal filesystem della macchina host, ma sarebbe conveniente il alcuni casi poter accedere e scrivere su file nella macchina host, per salvare una serie di dati, da analizzare dopo la terminazione delle macchine. Esistono due diverse modalità per permette un'interfaccia tra i filesystem della macchina host e dei vari dispositivi. Si possono condividere file tra i due filesystem direttamente, in modo che ogni cambiamento su uno dei due sia riflesso anche nell'altro. Oppure è possibile condividere una copia del file, in modo da avere due file indipendenti contenenti le stesse informazioni, quest'ultima è certamente la più semplice, ma la meno funzionale. Su Kathará sono presenti entrambi questi approcci. Utilizza una cartella `/shared` all'interno del lab, contenuta nei filesystem di ogni dispositivo in esecuzione in quel lab per condividere direttamente un file. Questa condivisione è abilitata di default, ma è possibile

modificarlo nelle impostazioni. Invece per condividere una copia di un file si possono utilizzare le subdirectories di un dispositivo, direttamente collegate alle subdirectories del lab di quello specifico dispositivo. In queste stesse subdirectories sono contenuti i file di avvio `.startup` delle singole macchine. Contengono comandi shell da essere eseguiti all'avvio, per configurare le interfacce di rete o avviare certi servizi di rete.

Per avviare un lab di Kathará bisogna aprire una Powershell, su Windows, e navigare alla directory del lab. In questa directory vanno eseguiti i vari l-commands, per avviare o terminare l'esecuzione di un lab.

Per evitare eventuali complicazioni, in questi laboratori si disabilita l'opzione per il protocollo IPv6, poiché genera complicazioni di semantica di difficile interpretazione.

2.2 Laboratorio del 16 Ottobre: "Two Computers"

In questa esercitazione si vuole emulare la una connessione tra due calcolatori `PC1` e `PC2`. Vengono specificate le ultime due cifre del MAC address: `0:1` e `0:2`. Viene consigliato di rappresentare la topologia della rete, descritta nel file `lab.conf`, i file di estensione `.startup` sono i comandi eseguiti dalle ciascuna macchine all'accensione. In questo laboratorio, non ci sono comandi da eseguire all'avvio. Non sono presenti neanche cartelle, per cui la configurazione di queste macchine è estremamente basilare

Il file `lab.conf` contiene all'inizio i meta dati del file:

Seguono le dichiarazioni ed assegnazioni dei computer nel laboratorio. Dove si specifica il nome della macchina, e tra parentesi quadre l'opzione da assegnare. Si indica con 0 l'interfaccia `eth0`. La stringa assegnata consiste nel dominio di collisione specificato `A`.

Senza specificare l'indirizzo, Kathará utilizza un indirizzo casuale, per cui in molti di questi laboratori gli indirizzi MAC di queste macchine verranno assegnati per renderli facilmente leggibili:

```
pc1[0]="A/00:00:00:00:00:01"
```

Con `image` viene specificata l'immagine di docker utilizzata dal calcolatore, contiene tutti gli strumenti necessari per mandare, ricevere e analizzare i pacchetti:

```
pc1[image]="kathara/base"
```

La terza riga di assegnazione consiste nella configurazione del protocollo IPv6, protocollo molto invasivo, per cui si vederebbero dati non di interesse in questo esercizio in particolare:

```
pc1[ipv6]="false"
```

Quindi si disattiva questo protocollo. Analogamente si configura la macchina 2:

```
pc2[0]="A/00:00:00:00:00:02"
pc2[image]="kathara/base"
pc2[ipv6]="false"
```

Inoltre è possibile modificare l'indirizzo IP di un calcolatore nella rete tramite il comando `ip` si accedono a tutti i comandi sulle reti. Il termine `link` indica a quale livello appartiene il comando, in questo caso il livello due di link. Si specifica quale protocollo deve essere modificato con `set dev eth0`, e si specifica cosa viene modificato, in questo caso l'indirizzo `address`:

```
prompt> ip link set dev eth0 address 00:00:00:00:01
```

Per avviare Kathará si apre una powershell all'interno della cartella del laboratorio, dopo aver avviato docker, e si avvia con il comando:

```
prompt> kathara lstart
```

Se viene modificato il file in `lab.conf` bisogna riavviare Kathará con il comando `lrestart`. Si chiude invece con il comando `lclean`.

All'avvio apre due terminali per entrambe le macchine, e vengono specificati i domini di collisioni e le connessioni definite nel file `.conf`. Per determinare la configurazione di un certo livello si utilizza sempre il comando `ip`, seguito dal livello che s vuole analizzare:

```
prompt> ip link
```

Al'interno di ciascuna delle macchine viene installato uno strumento chiamato `scapy`, una libreria in python utilizzata per creare pacchetti, ed in particolare gestire e modificare pacchetti. In questo modo si avvia il prompt di `scapy`, e si esce con il comando `exit`.

Questo terminale permette di creare un pacchetto ed inviarlo. Si crea una variabile `p` a cui assegnare il pacchetto. Si indica che si tratta di un pacchetto ethernet con `Ethernet()`, dove bisogna specificare gli indirizzi MAC del mittente e del destinatario:

```
prompt> p=Ethernet(dst:'00:00:00:00:00:01', src='00:00:00:00:00:02')
```

In questo modo è possibile specificare indirizzi MAC arbitrari, anche non presenti nella rete, e Kathará permette di analizzare questi comportamenti anomali. Il resto dei campi non specificati vengono inizializzati ad informazioni di default.

Per inviare un pacchetto si utilizza la funzione `send()`, che prende come argomenti il nome della variabile a cui è stato assegnato il pacchetto `p` ed il protocollo a cui viene inviato assegnato ad `iface`:

```
prompt> send(p, iface='eth0')
```

Esistono delle tecnologie chiamate "packet sniffer", come Wireshark, per controllare il traffico di rete. Questo viene specificato nelle ultime righe del file di configurazione, questa macchina tuttavia non è collegata, si dice fluttuante e sarà utile per analizzare il traffico su queste reti emulate.

```
wireshark[bridged]=true
wireshark[port]="3000:3000"
wireshark[image]="lscr.io/linuxserver/wireshark"
wireshark[num_terms]=0
```

L'ultima configurazione determina quanti terminali di Wireshark aprire all'avvio di Kathará. Si specifica 0, poiché si vuole utilizzare l'interfaccia grafica, e poiché in questi laboratori non si utilizzeranno comandi sul terminale i Wireshark.

Contiene un'immagine di Wireshark, chiamata analogamente per semplicità. Quest'applicazione viene avviata all'avvio del laboratorio. Questa macchina ha un'interfaccia grafica disponibile. Con il comando `lconfig` è possibile aggiungere un'interfaccia ad una macchina ad uno specifico link. Si specifica il nome della macchina con `-n` e si può specificare di aggiungere o rimuovere l'interfaccia con `--add` o `--rm`:

```
prompt> kathara lconfig -n wireshark --add A
```

Questo comando viene eseguito all'interno dell'hub del laboratorio, si utilizza per collegare questa macchina al dominio di collisione A. Nello stesso terminale dove è stato eseguito il laboratorio.

Il comando `bridged` connette la macchina all'host, con il comando `port` si specifica la porta dove la macchina condivide l'interfaccia grafica. Si accede tramite l'indirizzo `localhost:xxxx` dove viene specificata la porta inserita nella configurazione. Cliccando due volte sul nome dell'interfaccia, si possono analizzare i pacchetti inviati su quella connessione. Bisogna analizzare la connessione `eth1`, poiché la `eth0` viene inizializzata all'avvio di Kathará ed è connessa all'host. Il protocollo su vengono spediti i pacchetti creati, non essendo specificato.

Si può utilizzare la cartella `shared` costruita ogni volta che viene costruito il laboratorio, condivisa tra la macchina dispositivo e la macchina host. In questo modo è possibile utilizzare uno sniffer diverso da Wireshark per catturare i pacchetti. Si può effettuare quest'operazione tramite i comandi Linux `tcpdump`, con l'opzione `-tenny`, una composizione di tutte le flag necessarie per configurare il comando. Quando si manda un pacchetto, si può vedere il pacchetto sul terminale, direttamente. Aggiungendo l'opzione `-w` si può creare un file della cattura, di estensione `.pcap` "Packet Capture". Per salvare questo file nella cartella `shared`, condivisa, bisogna effettuare il comando in questa cartella. Ed è possibile aprirlo tramite Wireshark nella macchina host, per analizzare i pacchetti offline rispetto alla cattura.

2.3 Laboratorio del 25 Ottobre: "One Bridge"

In questo laboratorio si utilizza un bridge che collega quattro macchine su quattro domini di collisione differenti:

```
pc1[0]="A/00:00:00:00:00:01"
pc1[image]="kathara/base"
pc1[ipv6]="false"
pc2[0]="B/00:00:00:00:00:02"
pc2[image]="kathara/base"
pc2[ipv6]="false"
pc3[0]="C/00:00:00:00:00:03"
pc3[image]="kathara/base"
pc3[ipv6]="false"
pc4[0]="D/00:00:00:00:00:04"
pc4[image]="kathara/base"
pc4[ipv6]="false"
```

Ed una macchina che si comporta come bridge b1:

```
b1[0]="A/00:00:00:00:00:b1"
b1[1]="B/00:00:00:00:00:b2"
b1[2]="C/00:00:00:00:00:b3"
b1[3]="D/00:00:00:00:00:b4"
b1[image]="kathara/base"
b1[ipv6]="false"
```

Per permettere questo comportamento bisogna aggiungere un'interfaccia bridge con il seguente comando, che utilizza software per realizzare bridge, già presenti nei sistemi Linux. Si utilizza il software `ip link`, già utilizzato precedentemente:

```
root@b1:~$ ip link add name mainbridge type bridge
```

In questo modo si crea l'interfaccia di nome `mainbridge`, e di tipo `bridge`, all'interno della macchina b1.

Dopo aver creato il bridge bisogna connettere le diverse interfacce della macchina al bridge appena creato. Questo processo prende il nome di "enslaving", si realizza impostando il bridge come il master di quell'interfaccia, tramite il seguente comando:

```
root@b1:~$ ip link set dev eth0 master mainbridge
```

Questo va effettuato su ogni dominio di collisione a cui è connesso il bridge.

Quando viene realizzato il bridge, di default è spento e per attivarlo, o per fermarlo `set down`, si utilizza un ulteriore comando:

```
root@b1:~$ ip link set up dev mainbridge
```

Per controllare i bridge si utilizza un'altra serie di comandi già presenti in Linux, chiamata `brctl`, per "Bridge Control". Quando un bridge riceve un pacchetto, il MAC address del mittente viene salvato per un certo periodo di tempo nel suo filtering database. Dato che questo database è dinamico, il MAC address viene rimosso dopo un tempo di invecchiamento, di default di 5 minuti, o 300 secondi. Per modificare questo tempo si inserisce nel seguente comando come parametro, in secondi:

```
root@b1:~$ brctl setageing mainbridge 600
```

Bisogna specificare il nome del bridge precedentemente creato `mainbridge` ed il tipo di operazione da effettuare `setageing`.

Considerando questi comandi, il file `b1.startup`, permette di avere un bridge funzionate ed attivo ad avvio del lab:

```
ip link add name mainbridge type bridge
ip link set dev eth0 master mainbridge
ip link set dev eth1 master mainbridge
ip link set dev eth2 master mainbridge
ip link set dev eth3 master mainbridge
ip link set up dev mainbridge
brctl setageing mainbridge 600
```

Durante il lab per osservare il filtering database del bridge si può utilizzare il comando `showmacs`, che restituisce una tabella, contenente la porta, l'indirizzo MAC corrispondente. Inoltre contiene un'indicazione se quell'indirizzo MAC è locale, all'avvio infatti conosce automaticamente gli indirizzi MAC delle sue interfacce locali; ed il tempo di invecchiamento. Se un MAC è locale, il suo tempo non viene aumentato:

```
root@b1:~$ brctl showmacs mainbridge
port no mac addr            is local?   ageing timer
 1      00:00:00:00:00:b1   yes             0.00
 1      00:00:00:00:00:b1   yes             0.00
 2      00:00:00:00:00:b2   yes             0.00
 2      00:00:00:00:00:b2   yes             0.00
 3      00:00:00:00:00:b3   yes             0.00
 3      00:00:00:00:00:b3   yes             0.00
 4      00:00:00:00:00:b4   yes             0.00
 4      00:00:00:00:00:b4   yes             0.00
```

Il primo parametro indica il numero di porta del bridge, su un kernel Linux, il massimo numero di porte disponibili è di 1024, queste vengono assegnate sequenzialmente a partire da 1, nell'ordine in cui sono state connesse.

Se viene inviato un pacchetto da una delle stazioni, il bridge è in grado di imparare il suo MAC address:

```
root@pc1:~$ scapy
>>> p=Ether(dst='00:00:00:00:00:02', src='00:00:00:00:00:01')
>>> sendp(p, iface='eth0')
Sent 1 packets.
>>>
```

Dopo l'invio di questo pacchetto, il bridge conosce la posizione nella rete della stazione `pc1`:

```
root@b1:~$ brctl showmacs mainbridge
port no mac addr            is local?   ageing timer
    1      00:00:00:00:00:01   no             18.54
    1      00:00:00:00:00:b1   yes             0.00
    1      00:00:00:00:00:b1   yes             0.00
    2      00:00:00:00:00:b2   yes             0.00
    2      00:00:00:00:00:b2   yes             0.00
    3      00:00:00:00:00:b3   yes             0.00
    3      00:00:00:00:00:b3   yes             0.00
    4      00:00:00:00:00:b4   yes             0.00
    4      00:00:00:00:00:b4   yes             0.00
```

Se si invia un pacchetto con l'indirizzo MAC sorgente errato, allora il bridge non è in grado di imparare la posizione delle stazioni e quindi invierà pacchetti nei domini errati, impedendo ai pacchetti di arrivare a destinazione.

2.4 Laboratorio del 15 Novembre: "Basic IPv4"

In questo laboratorio si studieranno i comandi di base per IPv4, i comandi di ping, di rotta di indirizzamento, del protocollo ARP ed il loro funzionamento. Su questo laboratorio verranno utilizzate

cinque macchine, per configurare indirizzi IPv4 e le netmask, assegnarli agli host per raggiungere l'internet e LAN anche distanti tra di loro.

In questa rete una prima macchina `pc1` con indirizzo MAC `0:1` è connessa ad un dominio di collisione A, connesso al primo router `r1`, con indirizzo MAC `0:a1`. I due router sono connessi sul dominio di collisione B. Il router `r2` è connesso al domino B sull'indirizzo MAC `0:b2`, e connesso al domino di collisione C sull'indirizzo MAC `0:c1`. Sul dominio di collisione C sono presenti due macchine `pc2` sull'indirizzo MAC `0:2`, e `pc3` sull'indirizzo MAC `0:3`

Si utilizzano due /24 per le LAN con gli host per definire gli indirizzi. Per il dominio di collisione A si ha l'indirizzo 195.11.14/24, il dominio di collisione B ha l'indirizzo 100.0.0.8/30, per identificare solamente due indirizzi IP assegnati alle due macchine, l'indirizzo IP dell'ultimo domino di collisione è `200.1.1.0/24` Le macchine sono identificate `pc1` da `.5` e `r1` da `.1` sul dominio A. `r1` da `0.9` e `r2` da `.10` sul dominio di collisione B. Sul dominio di collisione C, la macchina `r2` su `.1`, `pc2` su `.7`, `pc3` su `.3`.

Per ogni macchina si vuole determinare un indirizzo IP per poter comunicare tramite il comando `ip address add`, seguito dall'indirizzo IP e dall'interfaccia `dev`. Inoltre bisogna specificare una rotta di default, il default gateway, tramite il comando `ip route add default via` seguito dall'indirizzo IP di default, seguito dalla specifica interfaccia dopo `dev`.

```
# pc1:
ip address add 195.11.14.5/24 dev eth0
ip route add default via 195.11.14.1
# pc2:
ip address add 200.1.1.7/24 dev eth0
ip route add default via 200.1.1.1 dev eth0
# pc3:
ip address add 200.1.1.3/24 dev eth0
ip route add default via 200.1.1.1 dev eth0
```

Per ogni router bisogna impostare due indirizzi IP per ogni dominio di collisione connesso. Quando si assegna un indirizzo IP su di un'interfaccia di un router, il sistema operativo assegna automaticamente quei prefissi nella tabella di instradamento per essere direttamente connessi. Ma bisogna aggiungere gli indirizzi che non sono direttamente connessi manualmente, per permettere al router di inoltrare pacchetti verso altri domini di collisione.

```
# r1:
ip address add 200.1.1.1/24 dev eth0
ip address add 100.0.0.10/30 dev eth1
ip route add 195.11.14.0/24 via 100.0.0.9 dev eth1
# r2:
ip address add 195.11.14.1/24 dev eth0
ip address add 100.0.0.9/30 dev eth1
ip route add 200.1.1.0/24 via 100.0.0.10 dev eth1
```

Per ottenere una descrizione della configurazione IPv4 di una macchina si utilizza il comando `ip address`. Il comando `routel` mostra tutte le destinazione che la macchina è in grado di raggiungere, mostrando la tabella di instradamento.

SI inserisce un link alla macchina wireshark sul dominio di collisione C, e si effettua un ping tra `pc3` e `pc2`, tramite i seguenti comandi:

```
> kathara lconfig -n wireshark --add C
```

Prima di effettuare una comunicazione tramite IPv4, la macchina deve individuare l'indirizzo MAC del destinatario, tramite il protocollo ARP. Per controllare gli indirizzi MAC memorizzati si può ursae il comando `arp`, si può inserire la flag `-n` per non risolvere gli indirizzi.

Wireshark fornisce anche l'informazione su quale pacchetto di risposta è legato a quale pacchetto di richiesta.

In questo modo si può analizzare il comportamento di una connessione diretta, per analizzare una connessione ad una LAN remota si può collegare la macchina wireshark al dominio B, ed effettuare un ping tra la macchina `pc3` e `pc1`. In questo modo si può osservare che i pacchetti "sniffati" da wireshark contegno solamente gli indirizzi MAC dello specifico dominio di collisione, non gli indirizzi MAC appartenenti al dominio A o C. Invece si utilizzano gli indirizzi IP reali del destinatario e del mittente.

Si esegue il comando `traceroute` con la flag `-z 1` per semplificare la visualizzazione. Su wireshark saranno visibili tutti questi pacchetti inviati ed il loro ttl progressivamente maggiore. Si esegue questo comando da `pc2` a `pc1`:

```
root@pc2:~# traceroute 195.11.15.5 -z 1
```

In caso si indicasse un indirizzo IPv4 inesistente effettuando un ping, vengono inviati i pacchetti di richiesta, ma la ARP request non riceve risposta quindi inoltra un errore al mittente. Il router che ha fallito la consegna inoltra il messaggio ICMP di errore. Se invece si prova a effettuare un ping su un indirizzo non presente su questa rete, viene sollevato un errore di tipo Destination Net Unreachable. Questi pacchetti di errore devono essere mandati da una macchina all'interno di questa rete che è in grado di determinare che l'indirizzo non appartiene alla rete. Questo indirizzo non è nella tabella di instradamento, quindi viene mandato al default gateway, ma non essendo nemmeno nella sua tabella di instradamento, viene scartato ed invia il messaggio ICMP al mittente.

Ora si prova ad effettuare un traceroute verso una macchina che non esiste su una LAN esistente, copia l'ultima riga del traceroute prima della macchina inesistente indicando il carattere `!H`, per specificare che manca la macchina host. Altrimenti se si specifica una LAN inesistente, produce solamente una riga con il carattere `!N`, per indicare che non è presente il network.

2.5 Laboratorio del 22 Novembre: "Basi IPv6"

2.6 Laboratorio del 6 Dicembre: "DNS"

In questo lab sono presenti 5 zone: uniroma3, it, "." (zona della root), net e startup. Quest zone sono gestite rispettivamente dai NS localuni e dnsuni, dnsit, dnsroot, dnsnet, dnsstart e localstart.

La topologia è piatta, tutte le macchine sono legate allo stesso dominio di collisione 192.168.0.0/24. Sono presenti altre due macchine pc1 e pc2, contenute rispettivamente in uniroma3 e startup.

Tutti questi NS sono configurati per poter gestire richieste iterative o ricorsive. L'immagine docker utilizzata per gestire i DNS si chiama docker bind 9.11. Non bisogna configurare alcuna rotta, poiché sono tutti direttamente connessi. In questa configurazione per la prima volta ogni macchina contiene una sua cartella, contenente un file `resolv.conf`, questo file indica alla macchina dove si trova il suo Local Name Server. Si utilizza il comando `search` per indicare per cercare un sotto-dominio di un dominio, poiché quando una ricerca fallisce, invia di nuovo la richiesta, appendendo questo parametro, in modo da cercare all'interno del suo dominio.

Kathará all'avvio copia nel percorso specificato i contenuti di queste cartelle. I DNS invece hanno una configurazione diversa, il nome del processo che parte e si occupa della gestione del DNS si chiama `named`, questo si trova all'interno della cartella bind, dal comando Linux utilizzato per generare questo processo DNS. Si indica nelle opzioni in questa cartella l'indirizzo dove salvare in cache gli indirizzi cercati. Le configurazione del DNS comprendono sempre un comando `include` passando coe parametro la posizione delle opzioni di questo DNS, nella cartella bind descritta precedentemente ed un comando per assegnare la zona `zona`, che indica il nome della zona, e determina se la macchina è root. Nelle opzioni la prima riga indica il tempo per cui i valori memorizzati in cache verranno mantenuti, indicato dal comando `$TTL`. In seguito ad una chiocciola, che utilizzato per non dover ripetere la zona del NS, si inseriscono le opzioni per la gestione della relazione master-slave e si indica con `SOA` che questo NS è l'inizio di una nuova zona. È presente la mail di posta del gestore del DNS, per operatori umani in caso qualcosa vada storto, in seguito tra parentesi è presente la configurazione tra slave-master del DNS, con un numero seriale, indicato da `serial`, un tempo di refresh, un tempo di retry per provare a ricontattare il master, in tempo secondo dopo il quale non prova più a contattare con il master, ed un tempo di mantenimento in cache delle informazioni negative. In ognuno di questi comandi si indica il tipo della connessione `IN`, in questi casi poiché si tratta sempre di connessioni internet.

In seguito si indicano le autorità per il dominio `.it` e `.next` indicando che questo NS conosce informazioni su queste due zone. Si indica che questi NS sono collegati allo stesso dominio di collisione A, con un indirizzo 192.68.0.5 per il NS della radice, 192.168.0.1 per dnsit e 192.168.0.2 per dnsnet.

Sono presenti configurazioni analoghe per lo SOA delle altre zone

2.7 Laboratorio del 13 Dicembre

3 Modello ISO-OSI

Per il funzionamento della rete gli standard sono strettamente necessari, altrimenti non sarebbe possibile una comunicazione tra un mittente ed un destinatario qualsiasi, alcuni di questi standard vengono imposti dalla case costruttrici, altri vengono definiti da organizzazioni internazionali, nell'ambito informatico o delle telecomunicazioni. Alcune di queste associazioni come IETF sono indipendenti da stati nazionali, dove varie aziende o istituzioni propongono modifiche di vecchi standard o introduzione e definizione di nuovi.

Il modello ISO-OSI rappresenta un importante strumento di classificazione nel modo delle reti. Venne realizzato in parte e sostanzialmente dismesso, ma nonostante questo viene utilizzato a livello globale.

Questo modello si basa sull'architettura stratificata di hardware o software, dove partendo da un nucleo centrale il sistema viene diviso in livelli o strati indipendenti dal livello inferiore, ed uno strato fornisce servizi solamente allo strato immediatamente superiore. Avanzando da uno strato al superiore i servizi vengono mostrati in modo sempre più astratto ed il sistema aumenta progressivamente di utilità. Per la sua utilità questo tipo di architettura stratificata permane molti campi dell'informatica.

La rete viene divisa in 7 livelli numerati dal basso verso l'alto, il livello indica la funzione delle tecnologie che vi appartengono e questo fornisce uno strumento di classificazione per analizzarle senza dover conoscere i loro meccanismi interni.

Ogni strato rappresenta un diverso livello di astrazione ed offrono funzioni ben definite. Poiché ognuno di questi strati è indipendente dal livello inferiore, viene minimizzato lo scambio di informazioni tra strati. Il numero dei livelli venne scelto in base alle funzioni distinte di una rete da descrivere e dalla realizzabilità.

All'interno di ogni strato si possono individuare diverse "entità", hardware o software dove sono contenuti i protocolli di quel livello. Per offrire servizi allo strato superiore, è presente un punto logico chiamato "Service Access Point" (SAP) al quale può accedere il livello superiore. L'unico punto di contatto tra livelli e quello inferiore è la loro interfaccia. Un protocollo è un linguaggio utilizzato da entità dello stesso livello, quindi entità di uno stesso strato possono comunicare con le adiacenti tramite protocolli e con superiori tramite SAP, ed inferiori tramite interfaccia.

Secondo questo modello i pacchetti sono contenuti in altri pacchetti, destinati a livelli inferiori, per cui quando vengono ricevuti da un livello $n-1$, viene letto il pacchetto di livello $n-1$ ed estratto il pacchetto di livello n contenuto ed inviato all'entità di livello n. I protocolli su uno stesso strato possono comunicare con altre entità dello stesso livello, e possono inviare indicazioni o conferme a richieste di entità utenti del servizio del livello superiore.

I dati generati da un protocollo di livello n sono detti n-pdu, "Protocol Data Unit", composti da un header indirizzato all'entità di livello n ed una payload, contente un $n+1$-pdu, destinata al livello superiore. Per cui all'aumento dei livelli aumenta l'overhead.

3.1 Livelli

I diversi livelli di questo modello presentano le seguenti funzioni:

1. Il primo strato della pila ISO-OSI rappresenta lo strato fisico, che si interfaccia direttamente con il mezzo trasmissivo della rete e quindi rappresenta il livello di natura fisica della trasmissione. Offre al livello superiore una comunicazione indipendente dal mezzo trasmissivo. Fornisce allo strato di collegamento servizi di trasmissione di bit a tra sistemi adiacenti, consegna in sequenza di bit o notifiche di malfunzionamenti.

2. Il secondo livello rappresenta lo strato data-link per risolvere eventuali malfunzionamenti dello strato fisico, rilevando e correggendo errori, tramite algoritmi di correzione, come i bit di parità. Offre allo strato superiore la possibilità di trasmettere pdu a sistemi adiacenti utilizzando due code nelle due direzioni.

3. Il terzo livello è lo strato di rete e conosce la topologia completa della rete, per effettuare operazioni di instradamento. Contiene i protocolli come IPv4, progressivamente sostituito da IPv6, e permette il trasferimento di pdu da estremo ad estremo. Inoltre permette la commutazione di circuito o di pacchetto a datagramma e a circuito virtuale.

4. Il quarto livello di trasporto, divide il messaggio in pacchetti, prova a colmare fluttuazioni della qualità del servizio dello strato di rete in modo trasparente rispetto agli strati superiori. In caso manchino dei pacchetti prova a recuperarli attraverso algoritmi di correzione, è il primo strato che risiede solamente nei terminali. Offre allo strato superiore la possibilità di instaurare una connessione e gestione della stessa, una trasmissione affidabile, ed il rilascio della connessione.

5. Il quinto livello di sessione sincronizza e struttura il dialogo tra due processi.

6. Il sesto livello di presentazione permette uno scambio di messaggi indipendentemente dalla sintassi della trasmissione.

7. Il settimo livello di applicazione offre un mezzo per accedere alla rete tramite un processo, interfacciando l'utente alla rete.

Per tutti i livelli superiori a quello fisico si possono definire due modalità operative ed associati servizi e protocolli, connessi e non connessi. Nei servizi o protocolli connessi, si instaura una connessione o dialogo tra le entità, e termina solamente dopo convenevoli finali. La modalità non connessa non ha bisogno di una connessione costante tra le due entità, viene instaurata senza un dialogo da una delle entità senza una terminazione.

Nella prima modalità l'entità non ha bisogno di ascoltare tutto il traffico per determinare quali pdu sono indirizzati alla stessa, ma necessita di una connessione continua anche se vengono trasmessi una piccola quantità di dati. Mentre nella seconda modalità possono essere inviate pdu indipendente dalla connessione e dalla distanza temporale tra le due, ma le entità che offrono questo servizio o protocollo devono costantemente analizzare il traffico per individuare le pdu a loro indirizzate. I protocolli non connessi sono quindi più efficienti, ma mancano di affidabilità, poiché manca una conferma di ricezione dei dati come nei protocolli connessi. quest'ultimi sono quindi più affidabili, ma meno efficienti, poiché dopo aver instaurato il dialogo non è possibile terminarlo preventivamente, e ciò può causare uno spreco di risorse.

In un servizio connesso sono presenti primitive per instaurare una connessione, inviare messaggi e una conferma di ricezione o ricevere messaggi, specificare l'indirizzo o nome della connessione ed abbattere la connessione. Mentre per servizi non connessi sono presenti solo primitive per inviare messaggi separatamene. Nelle reti LAN sono disponibili servizi connessi, solamente sul quarto strato, mentre nelle reti WAN è possibile siano offerti anche nel primo strato.

I primi tre livelli della pila ISO-OSI sono presenti su ogni nodo della rete non solo sui calcolatori, poiché rappresentano i livelli di trasmissione dei pacchetti, necessari anche nei nodi intermedi per poter trasmettere i pacchetti.

Nei protocolli di livello 2,3 e 4 si utilizzano meccanismi di riscontro o acknowledgment e tecniche di controllo a finestra, a riga indice e puntatore in avanti per correggere eventuali errori nei pacchetti.

Gli ultimi tre strati della pila si interfacciano con le applicazioni e lavorano generalmente in parallelo invece che in serie come il resto della pila.

Una singola connessione di livello n può essere sfruttata da più connessioni di livello $n+1$, interne in modo che gli n-pdu contengono entrambe le $n+1$-pdu delle due connessioni. Può essere il caso di connessioni tra più processi diversi sulle stesse due macchine, dove una singola connessione tra queste due macchine contenga numerose connessioni processo-processo tra le due.

Inoltre una singola connessione $n+1$ può utilizzare più di una connessione di livello n, per parallelizzare la trasmissione e velocizzarla partizionando i dati da inviare, oppure per implementare una resistenza ai guasti. La connessione $n+1$ utilizza più canali di comunicazione di livello n non in competizione.

Una singola connessione di livello $n+1$ nel tempo, può utilizzare più di una connessione di livello inferiore; è possibile che il terminale si sposta durante la trasmissione e si aggancia a reti diverse da quella iniziale, senza interrompere la connessione. Nello stesso caso, una stessa connessione di livello n continua nel tempo può essere utilizzata da diverse connessioni di livello $n+1$. La connessione originale dell'esempio precedente vede uscire il primo terminale e quindi la prima connessione $n+1$ per poi vedere accedere un altro terminale ed un'altra connessione $n+1$.

4 Livello 2: Standard IEEE 802

Lo standard IEEE 802 riguarda i primi due livelli del modello ISO-OSI, ovvero il livello fisico, ed il livello data link. Le tecnologie definite in base a questo standard quindi hanno come obiettivo la trasmissione e la rivelazione o correzione di bit attraverso un mezzo trasmissivo. Si occupano della connessione e quindi comunicazione tra macchine adiacenti, per una qualche definizione di adiacenza. Altri protocolli e standard noti sono l'IPv4 ed IPv6, protocolli di routing di livello tre, protocolli TCP ed UDP di livello quattro, ed il protocollo HTTP di livello sette, ma si occupa anche da solo delle funzioni dei livelli 5 e 6.

Questi standard vengono realizzati dall'organizzazione IEEE, Institute of Electrical and Electronics Engineers, organizzazione indipendente da stati sovrani. Il progetto IEEE 802 venne definito con l'obiettivo di realizzare una serie di standard di livello fisico e data-link per permettere la comunicazione di calcolatori sulla stessa rete locale, LAN, personale, PAN, o rete metropolitana, MAN, di grandezza intermedia tra le reti LAN e WAN. Ha avuto successo sopratutto per le reti LAN e MAN, ma per le reti personali si utilizza uno standard diverso basato sul bluetooth. Questi standard riguardano tecnologie con pacchetti di lunghezza variabile.

Le specifiche tecnologie vengono individuate tramite una notazione puntata, con $802.x$, dove x rappresenta un numero, ed identifica la tecnologia. I numeri precedenti al punto individuano lo standard dove è stata introdotta questa tecnologia. Ma le tecnologie non rimangono invariate nel tempo, per cui si possono assegnare delle lettere dopo il numero per specificare la versione o tipo di quella specifica tecnologia.

Lo standard IEEE 802.2 divide il livello due in due sottolivelli: "Logical Link Control" (LLC) e "Media Access Control" (MAC), questi gestiscono due tipologie diverse di pacchetti. Per le diverse tecnologie dello standard, il livello MAC è diverso, mentre il livello LLC è comune a tutti. Il sottolivello MAC è specifico per ogni tipo di LAN, si suppone che tutti i calcolatori che devono comunicare siano nella stessa LAN. Data questa ipotesi il sottolivello MAC risolve il problema di determinare il destinatario in ricezione, e di verificare la disponibilità della LAN in trasmissione, in caso la LAN sia a singolo canale condiviso. Quindi bisogna evitare che il canale sia utilizzato da più utenti.

4.1 802.2: Sottolivello MAC

Poiché il canale è condiviso, tutti gli utenti possono vedere i pacchetti inviati, sono quindi necessari protocolli di sicurezza e cifratura per impedira che sia possibile a chiunque connesso alla rete leggere il contenuto dei pacchetti. Tecniche che non verranno trattate in questo corso. Inoltre si utilizza un canale condiviso poiché se ci fosse un malfunzionamento fisico, una sola connessione verrebbe compromessa e non l'intera rete.

Per determinare il destinatario di un pacchetto nella MAC pdu è presente un campo per definire il tipo di trasmissione:

- Punto a Punto: da un calcolatore ad un altro nella LAN;
- Punto a Gruppo: da un calcolatore a diversi altri nella LAN;
- Broadcast: a tutti gli utenti connessi alla LAN.

Per permettere di identificare univocamente un unico elemento nella rete, gli indirizzi MAC devono essere univoci nella rete considerata. Dato che è possibile connettersi ad una LAN dall'esterno senza conoscere gli indirizzi MAC utilizzati, servirebbe un gestore di rete per assegnarli ad ogni nuova connessione, ma questo è un approccio inefficiente. Si utilizzano quindi indirizzi MAC univoci a livello mondiale, in questo modo nell'intera rete esisteranno solo indirizzi MAC differenti. Questa condizione vale anche su VPN o LPN, inoltre se su una stessa macchina vengono simulate diverse macchine virtuali, ognuna di esse dovrà avere un indirizzo MAC differente, all'interno della rete locale che utilizzano per comunicare tra di loro. Se due macchine avessero lo stesso MAC, riceverebbero gli stessi pacchetti, e verrebbero riconosciuti da entrambe le macchine come propri.

La MAC pdu è composta da diversi campi, che possono variare in base alla tecnologia con l'aggiunta di campi specifici. Ma per ogni tecnologia aderente allo standard IEEE 802 sono presenti sicuramente questi quattro campi per la MAC pdu:

- MAC-dsap (Destination Service Access Point): indirizzo di destinazione;
- MAC-ssap (Source/Send Service Access Point): indirizzo di partenza;
- Info: LLC pdu;
- FCS (Frame Check Sequence): per identificare e correggere eventuali errori.

Per ottenere l'indirizzo del destinatario, si utilizzano protocolli di acquisizione descritti in seguito.

Gli indirizzi MAC sono composti da 6 byte, in base allo standard EUI-48, "Extended Unique Identifier", per indirizzi a 48 bit, ma esiste anche uno standard a 64 bit non utilizzato. Questi byte vengono rappresentati in forma esadecimale, separati da due punti o trattini. I primi tre byte dell'indirizzo MAC vengono assegnati al costruttore e rappresentano gli OUI "Organization Unique Identifier", gli ultimi 3 byte vengono scelti dal costruttore. Per cui dato un indirizzo MAC, è sempre possibile determinare il costruttore della macchina a cui appartiene.

Esistono diversi tipi di indirizzi MAC, in base al valore di determinati bit:

- Unicast: indirizzi che individuano le singole schede di rete dei calcolatori; se l'ultimo bit del primo byte ha valore zero;
- Multicast: indirizzi che identificano gruppi di schede di rete; se l'ultimo bit del primo byte ha valore uno;
- Broadcast: identificano tutte le schede di rete; se l'indirizzo è `FF:FF:FF:FF:FF:FF`.

Inoltre è possibile assegnare indirizzi MAC non unici a livello mondiale, specificando il valore del penultimo bit del primo byte, in modo che abbia valore uno. In questo modo è possibile gestire localmente indirizzi MAC nella stessa LAN.

Per risolvere i conflitti in trasmissione si utilizzano nelle rete WiFi degli algoritmi distribuiti sulle singole macchine in contemporanea, che collaborano per determinare a chi abilitare gli accessi alla rete.

4.2 802.2: Sottolivello LLC

Il sottolivello LLC consegna al livello MAC un pacchetto da spedire, questo pacchetto è uguale per ogni tecnologia aderente allo standard e presenta campi analoghi al sottolivello MAC. I campi LLC-dsap/ssap individuano gli indirizzi LLC del mittente e destinatario, il campo info contiene il tipo di pdu per gestire diverse tipologie di pacchetto, e l'ultimo campo contiene la pdu di terzo livello. Gli indirizzi LLC non individuano macchine come gli indirizzi MAC, ma vengono utilizzati per identificare i protocolli di livello 3 a cui sono indirizzati i pacchetti. Consentono la convivenza di diversi protocolli di livello 3 sulla stessa macchina e sulla stessa LAN, dove sono presenti diverse pile protocollari con obiettivi e versionatura diversa.

Gli indirizzi LLC vengono attribuiti dall'IEEE solo a protocolli "ufficialmente" standard, ma questa è una classificazione che ignora molti dei protocolli più utilizzati a livello globale come TPC-IP, il protocollo più utilizzato al mondo. Vengono identificati da un byte in esadecimale, se non sono protocolli standard, il loro indirizzo è `AA`, ed il pacchetto subisce una variazione con la snap-pdu, "SubNet Access Point", dopo il campo control di 5 byte per identificare il protocollo. Questo rappresenta un ulteriore livello di overhead, già elevato per il modello ISO-OSI, per i pacchetti.

4.3 802.3: Ethernet

La prima tecnologia ethernet nacque nel 1970 dal consorzio DIX, dalle iniziali delle tre più grandi case produttrici informatiche dell'epoca: Digital, Intel e Xerox. In seguito venne revisionato negli anni '80 due volte, nel 1989 lo standard IEEE 802.3 diventa lo standard ISO 8802.3, e negli anni '90 ebbe talmente successo sulle reti LAN e WAN che divenne essenzialmente lo standard di tutte le trasmissioni su filo, completamente retrocompatibile. Nel tempo ha usato diversi mezzi trasmissivi, da cavi di rame intrecciati a coppie, alla fibra ottica moderna. I cavi di rame venivano avvolti da una isolante dielettrico ed una schermatura esterna, dove erano presenti quattro coppie di cavi di rame intrecciati. Questo permette connessioni punto-punto bidirezionali e contemporanee.

Dagli anni '90 in poi la banda massima possibile è aumentata fino ad un massimo di 100-400 Gb/s.

Il formato del pacchetto ethernet è rimasto sostanzialmente invariato nel tempo, nonostante le sue revisioni, nello standard IEEE 802.3 presenta i seguenti campi:

- Preambolo (56 bit): veniva utilizzato per sincronizzare in fase i pacchetti, modalità di trasmissione non più in uso;
- SFD "Start Frame Delimiter" (8 bit): indica l'inizio del pacchetto;
- Indirizzi MAC di sorgente e destinazione (96 bit);
- Lunghezza del campo dati (16 bit);
- Dati: di lunghezza variabile con un massimo di 1500 Byte, contiene una LLC-pdu;
- Pad: eventuale riempimento, da 0 a 46 Byte, la somma con il campo dati deve essere compresa tra 46 e 1500 Byte;

- FCS "Frame Check Sequence" (32 bit): contiene il valore del codice di ridondanza ciclica (CRC) calcolato.

Non è presente invece un delimitatore finale del pacchetto. I pacchetti hanno una lunghezza minima di 512 bit, mentre una lunghezza massima di 1512 Byte, esclusi il preambolo ed il SFD. Si definisce per rete ethernet l'"Inter-Frame Space" (ITR) o "Inter-Packet gap" (IPG) come il tempo minimo tra due pacchetti consecutivi. Questo viene definito indipendentemente dalla velocità della rete, come il tempo necessario per inviare 96 bit, è necessario per permettere di distinguere due pacchetti inviati consecutivi e determinare si tratti di spazio tra i due.

Agli albori di questa tecnologia si utilizzavano reti condivise, quindi bisogna effettuare una connessioni a turni, e per occupare la connessioni su reti di 5 km, si scelse la lunghezza minima di 512 bit. Analogamente se il pacchetto è troppo grande, occuperebbe il mezzo trasmissivo per troppo tempo, quindi si è imposta una lunghezza massima.

Nella trasmissione ethernet il livello MAC in trasmissione riceve la LLC-pdu, inserendolo nel pacchetto di livello MAC e convertendolo in bit da passare al livello fisico. In ricezione, converte i bit in un pacchetto MAC, se questo è indirizzato ad un altro oppure contiene errori, calcolando il CRC, viene scartato, altrimenti viene rimossa la parte MAC ed inviato al livello LLC. Scartando pacchetti in questo modo si perde di affidabilità del sistema, ma si guadagna efficienza, poiché si suppone queste informazioni perse vengano recuperate ad un livello superiore (livello di trasporto), senza che sia l'ethernet ad inviare una richiesta di ritrasmissione. Ogni terminale che riceve pacchetti deve quindi ricalcolare il CRC, quest'operazione potrebbe rappresentare il collo di bottiglia e deve essere il più veloce possibile. Questo livello garantisce una distanza minima tra pacchetti rispettando l'IPG e verifica la lunghezza minima del pacchetto. Se un pacchetto non rispetta la lunghezza minima, viene anch'esso buttato. Vengono inoltre generati, in trasmissione, e rimossi, in ricezione, il preambolo e lo SFD dal pacchetto.

In passato le reti ethernet erano formate da connessioni sullo stesso filo condiviso non era punto-punto e bidirezionale, chiamato dominio di collisione, tra varie stazioni ed usato a turno. I pacchetti inviati da due o più macchine potevano collidere e si poteva richiedere una ritrasmissione da queste. Questo mezzo condiviso era realizzato da un repeater o ripetitore o hub, questo ripete il segnale su tutti i canali connessi e li amplifica, a causa della lunghezza delle reti infatti, il segnale poteva perdere di potenza durante la trasmissione. A livello fisico si comporta come un filo, non memorizza infatti i pacchetti che riceve, ma li trasmette, non è una macchina "Store & Forward". Ripetitori del genere sono ancora diffusi in alcuni contesti. Si trova al livello fisico ed ha diverse porta su cui può ricevere dati e trasmetterli su tutte le altre.

Il mezzo trasmissivo per le connessioni ethernet viene realizzato in cavi di rame o fibra ottica, al massimo di 100 m di lunghezza, progettati senza amplificatori. Si usano coppie separate per trasmissione e ricezione per mantenere una connessione bidirezionale. Si usano connettori RJ-45. Reti a fibra ottica invece possono percorrere distanze molto significative rispetto a cavi in rame.

Ethernet II e ethernet IEEE 802 sono diversi tra di loro, lo standard ethernet II non ha uno sottolivello LLC ed è in generale molto più snello. Nelle reti locali convivono connessioni di entrambi gli standard, e sono tendenzialmente molti di più nella vecchia versione di ethernet. Sono necessari quindi meccanismi per mantenere la retrocompatibilità completa dei pacchetti. Non essendoci lo strato LLC, il pacchetto di terzo livello viene contenuto direttamente nel pacchetto MAC, poiché non esiste il livello LLC per questi pacchetti, si utilizza un altro campo type, per specificare a

quale protocollo di livello superiore inviare il pacchetto. Questo sostituisce il campo lunghezza di IEEE 802.3. Viene riconosciuto se il campo lunghezza ha un valore maggiore di 1500, lunghezza massima del campo data, e viene interpretato come un codice identificativo di un protocollo di livello superiore.

Poiché ethernet ha vissuto un enorme crescita tecnologia, sono presenti diversi sottolivelli dello standard per classificarle, da versioni 802.3u da 100 Mb/s alle ultime versioni 802.3ba/bg/bm dai 40 ai 100 Gb/s. Sono inoltre in corso di standardizzazione tecnologie ethernet nell'ordine dei terabit al secondo.

Generalmente se nel sottolivello dello standard è presente una lettera maiuscola, questo rappresenta un tecnologia estremamente importante.

Sono presenti inoltre molte versioni differenti per scopi diversi, esiste uno standard specifico per il mercato automobilistico, dove è presente poco spazio interno, e quindi si utilizzano cavi da una singola coppia di cavi intrecciati: 802.3bw e 802.3bp (2015/2016) permettono una banda nell'ordine dei gigabit sulla singola coppia. Esiste inoltre uno standard per inviare corrente elettrica su ethernet ed alimentare tramite ethernet telefoni IP a bassa tensione: 802.3af e 802.3at (2003/2009).

4.4 802.1D: Bridge-Switch

La connessione instaurata tramite ethernet è bidirezionale simultanea solamente su due calcolatori, ma su una stessa connessione può inviare i dati un solo calcolatore, quindi sono necessarie altre componenti. Un bridge è una componente che consente di connettere tra di loro più di un computer tramite ethernet, comportandosi come se fosse un calcolatore intermedio ai calcolatori della rete, connesso a ciascuno di questi tramite una connessione ethernet.

Inoltre connettendo tra di loro diversi bridge è possibile creare una struttura più articolata, creando una struttura simile ad un albero. La parte wired o cablata delle connessioni LAN vengono instaurate in questo modo.

I bridge svolgono una prima funzione di rendere possibili topologie articolate, effettuando un'operazione di "filtering", per separare tra di loro porzioni di rete che non devono dialogare tra di loro in modo diretto

I bridge sono delle macchine "store & forward", ovvero quando ricevono un pacchetto, prima di essere inviato su altre porte, viene memorizzato e trasmesso su altre porte, analogamente come se fosse un calcolatore, in caso le altre porte siano impegnate a trasmettere altri pacchetti, quindi in caso di traffico. Si può quindi immaginare una coda di pacchetti sulle porte del bridge per essere trasmesse.

Il bridge sono delle tecnologie di livello 2, ed utilizzano algoritmi di instradamento per inviarli ad un MAC address specifico, ma questo tipo di algoritmo viene effettuato a livello 3. Questo non sorge problemi, poiché quest'operazione di instradamento è interna alla LAN, e non coinvolge alcun'altra componente della rete. I bridge devono essere conformi allo standard IEEE 802.1D. Gli standard comprendenti il carattere "D", sono di grande importanza. I sistemi connessi a reti LAN ignorano i bridge, si dicono quindi trasparenti, poiché i calcolatori connessi alla rete non conoscono la loro posizione all'interno della rete.

Un calcolatore per inviare un messaggio ad un altro calcolatore su una rete LAN, invia il suo pacchetto ad un bridge attraverso il sul MAC address. Il bridge quindi utilizza in principio un

diverso MAC address, per spedire questo pacchetto al computer di destinazione tramite il suo MAC address. Tra questi due MAC è presenta un componente di relay, per trasmettere il pacchetto tra porte diverse del bridge.

Le porte di un bridge possono avere lo stesso MAC o MAC differenti. Poiché il pacchetto è specifico al MAC del bridge, deve ricostruire il pacchetto scartando i campi specifici al MAC address del bridge. Inoltre poiché i pacchetti non sono tutti conformi allo standard IEEE 802.3, deve ricostruire anche il campo LLC. I MAC address dei computer nella rete sono realizzati in modo da poter essere connessi a ciascun tipo di MAC.

Si vuole che il modello di rete sia "plug & play", ovvero non deve essere dipendente da un intervento umano. I bridge costruiscono la loro tabella di instradamento per identificare dove sono presenti i diversi MAC address, autonomamente attraverso un meccanismo di "learning", salvando questa tabella nel "filtering database". Ogni porta del bridge rappresenta una linea ethernet diversa, identificando un loro dominio di collisione, a cui possono essere connessi diversi calcolatori.

Si considera una rete dove ogni componente connesso è spento, ed una tabella vuota. Appena si accende un calcolatore ed invia un pacchetto da un dominio di collisione, allora il bridge capisce a quale porta corrisponde il MAC address del mittente. Ma ancora non conosce dove si trova il destinatario, quindi lo invia su tutte le sue porte disponibili, su tutta la rete. Invece se conosce la porta dov'è presente il destinatario lo invia solamente su quella porta.

Il learning permette di costruire autonomamente il filtering database di un bridge, questo meccanismo tuttavia non funziona quando al rete presenta una topologia diversa dalla topologia ad albero. Per esempio se è presente un ciclo all'interno della rete, il bridge si vede arrivare un pacchetto dallo stesso MAC address su porte diverse. Ma un albero è una topologia contenente solo "Single Points of Failures" (SPoF) e quindi fortemente sconsigliata, poiché un singolo malfunzionamento causerebbe la perdita di funzionalità dell'intera rete. Per cui data una topologia a grafo, un bridge è in grado di calcolare autonomamente un albero ricoprente della rete, ad ogni cambio di topologia della stessa. I bridge inoltre vengono collegati tra di loro per più di una connessione per evitare altri SPoF, ed evitare che a un singolo guasto la rete venga tagliata in due.

Tramite un meccanismo progressivo i bridge individuano la loro posizione nella struttura dell'albero ricoprente e sono in grado di staccare alcune porte e rimanere collegati sull'intera rete; quando questi bridge rilevano un guasto su una di queste connessioni, riattivano una delle porte disattivate per mantenere in funzione la rete, ottenendo una significativa resistenza ai gusti. Questo tipo di algoritmo di spanning tree verrà trattato in corsi più avanzati di reti di calcolatori.

Le prestazioni di un bridge influenzano le prestazioni dell'intera rete locale, vengono identificate da una serie di parametri. Il numero massimo di pacchetti al secondo processabili dal bridge, rappresentano un collo di bottiglia per i pacchetti che possono essere presenti sulla rete in ogni singolo momento, se vengono inviati un numero superiore ci pacchetti, alcuni pacchetti verranno scartati. Un altro parametro caratteristico è il tempo medio di latenza, ovvero il tempo in cui il bridge prende le sue decisioni ed invia il pacchetto alla porta giusta. Per cui è preferibile avere bridge full speed, ovvero con una velocità pari al massimo teorico. Più corti sono i pacchetti maggiore è il numero di decisioni effettuate nell'unità di tempo. Nello standard IEEE 802.3 a 10 Mb/s, un bridge si definisce full speed, se è in grado di processare 41880 pacchetti al secondo, per ogni porta. Questi esperimenti di verifica a parità di frequenza devono essere effettuati utilizzando pacchetti di lunghezza minima, così ad ogni porta è presente la massima frequenza di funzionamento del bridge.

Il pacchetto più piccolo che può essere inviato è da 512 bit, per cui il numero massimo di pacchetti al secondo ad una velocità di 10 Mb/s è di circa 19500 pacchetti, ma il pacchetto comprende anche il preambolo ed lo SFD, per cui vanno aggiunti altri 64 bit, numero di pacchetti al secondo scende quindi a 17300. Tuttavia tra un pacchetto ed il successivo in ethernet è presente l'"interpacket gap" di 96 bit. Per ciascuna tipologia di porta del bridge si effettua questa analisi e si verifica nel caso peggiore quanti pacchetti è in grado di gestire.

Il bridge è un calcolatore, con una CPU, RAM ed interfacce per le diverse LAN, in ROM le funzionalità dello standard Per bridge più potenti, le porte vengono realizzate tramite schede ASIC, per risolvere il problema dell'instradamento localmente. Le porte vengono realizzati tramite diversi slot che possono essere inseriti o rimossi in base al tipo di porta necessaria. Inoltre sono necessari massicci impianti di raffreddamento per riuscire a mantenere la temperatura del data center. A differenza di un server che può essere rallentano in caso di traffico allentato e quindi diminuire la temperature, le apparecchiature di bridge non possono spegnersi, e quindi comportano una temperatura costantemente elevata. Bridge di fascia alta sono in grado di effettuare bilanci sulle prese di corrente.

Logicamente sono presenti almeno due porte una "MAC relay entity", per trasmettere i pacchetti tra le varie porte, ed un'entità di livello superiore, per la gestione del bridge, degli algoritmi e dei protocolli. Queste entità di alto livello comunicano con altri bridge attraverso pacchetti per realizzare lo spanning tree.

Le porte del bridge possono essere abilitate o disattivate dall'amministratore di rete. Una porta attiva può essere in stato di "forwarding" o di "blocking", se sono bloccate lo spanning tree lo ha bloccate. Ogni porta ha un indirizzo MAC univoco, e sono numerate progressivamente a partire da uno. Convenzionalmente l'indirizzo MAc del bridge corrisponde all'indirizzo MAC della porta numero uno.

La tabella di instradamento contiene "entries" (righe) statiche o dinamiche. Le righe statiche vengono inserite dall'amministratore a causa di esigenze di sicurezza importanti, altrimenti la posizione del MAC address vengono mantenuti per un tempo finito, configurabile, e di default di 5 minuti. Infatti è possibile che il calcolatore venga spostato spazialmente attraverso la rete, è quindi possibile si colleghi ad una porta differente.

Lo sviluppo di ethernet ha portato alla creazione di meccanismi di controllo di flusso, soprattutto per gli switch. Una richiesta attraverso il bridge può essere di pochi byte verso un server, ma può provocare un trasferimento notevole di dati verso il client, quindi attraverso il bridge ad una porta ad alta velocità, ma questi pacchetti da ritrasmettere verso il cliente passano attraverso una porta di banda minore, quindi la porta più lente può andare facilmente in saturazione. Viene introdotto quindi tramite lo standard IEEE 802.3x e 802.3bd un controllo di flusso tramite dei "pause frame" un MAC control frame di 512 bit, per fermarsi prima di riprodurre traffico I pause frame non contengono dati ma contengono informazioni di controllo, e rappresentano una novità, viene quindi implementato attraverso un nuovo sottostato di MAC chiamato MAC control. Il supporto allo standard 802.3x è opzionale e viene negoziato tra le schede alle due estremità del filo.

Prima dello standard 802.3x poiché ethernet era una comunicazione a turni, per impedire la saturazione i bridge potevano inviare pacchetti senza dati prendendo il controllo della connessione.

4.5 802.11: WiFi

A differenza di ethernet, dove si ha una connessione punto a punto bidirezionale e simultanea, attraverso un bridge permette di comunicare a diversi computer con una singola connessione tra tutti i calcolatori ad un bridge. Per cui i MAC ethernet è molto semplice, ma questo non si può dire per la rete WiFi. Una rete locale è un ambiente dove tutti parlano contemporaneamente con tutti, e ciò non può avvenire su di un filo condiviso tra due calcolatori. Quindi le connessioni WiFi presentano un indirizzo MAC molto più complesso.

Ethernet è molto versatile, ma su alcuni edifici non è possibile effettuare un cablaggio economicamente, oppure sono presenti uffici nei quali gli impiegati sono presenti occasionalmente. Oppure nel caso di reti offerte pubblicamente, con utenti occasionali. Per questi motivi di praticità nei portatili moderni, sono presenti schede di rete WiFi e non ethernet.

Nonostante questi benefici, il mezzo trasmissivo non è affidabile, e comporta un costante consumo elettrico per connettersi alla rete. Comprende una zona di copertura limitata. Esistono diversi studi sull'uso delle frequenze o micro-frequenze sulla salute.

Il sistema WiFi nel venne definito nel comitato IEEE 802, dove forma il working group 902.11 dedicato alle LAN senza fili. Il primo standard ad affermarsi è 802.11b. Nel 1999 si forma il consorzio Wireless Ethernet Compatibilità Alliance, successivamente denominato Wi-Fi (Wireless Fidelity) Alliance per certificare i prodotti IEEE 802.11.

Una rete WiFi può presentare architetture di due tipi, ad hoc o strutturate. In una rete ad hoc le stazioni comunicano direttamente l'una con l'altra.

Quest'architettura è prevista dallo standard, ma le reti WiFi non funzionano in questo modo. Quest'architettura è stata progettata per permettere di comunicare dispositivi personali, ma su questo ambiente applicativo ha prevalso bluetooth.

Sono realizzate tramite architettura strutturata, tutte le stazioni, astrazione di calcolatore poiché funziona su tante tipologie di dispositivi, possono accedere solamente tramite punti di accesso (Access Point o AP). Questi punti di accesso sono interconnessi mediante fili, quindi non rappresenta una rete completamente senza fili. Questi collegamenti vengono effettuati tramite ethernet, rappresentato come un unico domino di collisione, nonostante sia presente una topologia più complessa. In queste reti non è presente un master, ma tutti gli AP sono allo stesso livello, in modo che non siano presenti SPoF.

Quest'architettura permette una semplice scalabilità, infatti è sufficiente connettere altri AP alla rete ethernet. L'informatica deve essere scalabile, con costi limitati deve poter aumentare le sue infrastrutture.

Ciascuno di questi AP ha un MAC di tipo IEEE 802.11, e si comporta come un bridge, presenta almeno due interfacce, una 802.11 per comunicare con i dispositivi wireless, ed un'altra porta 802.3 per comunicare sulla rete ethernet. Ma questi pacchetti inviati con due MAC differenti, necessitano di un'entità di relay per poter gestire questi diversi tipi di MAC. Ogni AP controlla un "Basic Service Set" (BSS), che estende la rete cablata. Ogni BSS ha un identificatore (BSSID), può essere utilizzato il MAC della scheda AP corrispondente. Ma è possibile modificare questo BSSID. La parte cablata si chiama "Distribution System" (DS) e rappresenta l'infrastruttura portante della rete.

Sono presenti delle eccezioni, infatti è possibile costruire dei collegamenti tra AP che estendono il DS tramite WiFi. Ma dal punto di vista dell'architettura si comporta come un filo, collegando solamente due AP.

Per gestire il sottolivello MAC, bisogna gestire il problema dell'accesso al mezzo trasmissivo condiviso. Inoltre bisogna avere una spedizione affidabile dei pacchetti su un mezzo trasmissivo poco affidabile. Livelli fisici differenti utilizzano frequenze e bande diverse. Il sottolivello MAC viene a sua volta diviso in due sottolivelli, questo viene effettuato per permettere di realizzare architetture modulari. Un sottolivello "Distributed Coordination Function" (DCF) e "Point Coordination Function" (PCF), questi due livelli indipendentemente sono connessi al LLC. L'accesso distribuito al mezzo trasmissivo viene realizzato nel DCF, questo è il modello effettivamente utilizzato. Rappresenta il MAC vero e proprio, e tramite questo le macchine tentano di accedere al mezzo trasmissivo, ma non è garantito questo accesso, e può comportare ritardi. In alcuni ambienti applicativi si vuole accedere al mezzo trasmissivo con garanzia di un tempo di ritardo massimo che non superi una certa soglia. In queste applicazioni si sceglie una stazione di riferimento che in ogni intervallo temporale indica chi può trasmettere con il mezzo trasmissivo. Ambienti "Mission Critical", dove una macchina deve necessariamente effettuare un azione in un certo intervallo di tempo.

La probabilità che il ritardo sia elevato in una rete WiFi è molto bassa, per cui questa tecnologia non viene quasi mai utilizzata. Poiché in questo caso bisognerebbe utilizzare un controllore unico che si cerca molto spesso di non utilizzare in una rete.

Per qualunque MAC IEEE 802.2 deve essere presente l'indirizzo del destinatario, del mittente, il campo dati e l'FCS. Esiste un altro campo "Frame Control" nello standard 802.11 che indica il tipo di pacchetto, di controllo o contenente dati, e fornisce anche informazioni sul mittente ed il destinatario del DS, sulla frammentazione e sulla riservatezza. Contiene un campo per indicare di durate che indica il tempo necessario in cui deve essere effettuata la trasmissione. Sono presenti fino a quattro indirizzi MAC. Il "Sequence Control" contiene informazioni utili per la frammentazione o il riassemblaggio dei pacchetti.

Si parla quindi di indirizzamento per spiegare il motivo per cui sono presenti fino a quattro indirizzi MAC in un unico pacchetto. Ogni scheda di rete wireless contiene un suo indirizzo MAC, e questa raccoglie i pacchetti e verifica che sia diretto alla stessa macchina dal suo indirizzo MAC. Nel pacchetto di livello MAC sono presenti quattro indirizzi numerati da 1 a 4, più due bit. Questi bit sono ToDS e FromDS, se il primo bit vale 1, questo pacchetto è spedito all'AP per essere smistato dal DS, il secondo vale uno quando il pacchetto proviene dal DS. In funzione di questi valori i quattro indirizzi hanno significati diversi.

Quando due computer comunicano direttamente, un una rete ad hoc, il primo indirizzo del pacchetto corrisponde all'indirizzo del destinatario ed il seguente all'indirizzo del mittente. Il terzo campo di indirizzo corrisponde al BSSID, ma questo si riferisce solamente alle reti strutturate. In questo modello le macchine comunicano tra di loro in gruppi chiusi, quindi condividano un identificatore chiamato BSSID. La connessione è quindi lecita solamente se il BSSID è lo stesso. Poiché si tratta di una comunicazione diretta, i suoi bit hanno valore nullo.

In una trasmissione da una stazione ad un AP, il primo indirizzo è l'indirizzo dell'access point, il secondo è quello del mittente vero. Il terzo indirizzo è quello del pacchetto vero. Questi AP sono

comunque trasparenti rispetto alle macchine, poiché alla prima connessione il calcolatore memorizza l'indirizzo del BSSID a cui può richiedere accesso per inviare e ricevere pacchetti.

Quando un pacchetto viene inviato dal DS, il primo indirizzo è il MAC del destinatario vero, il secondo è l'indirizzo dell'AP che lo invia, ed il terzo è l'indirizzo del vero mittente.

NNell'ultimo caso, il pacchetto è inviato e ricevuto dal DS, è il caso di due AP che comunicano tra di loro tramite WiFi. Sono necessari i due indirizzi degli AP e gli indirizzi del destinatario e del mittente vero.

Per realizzare dei turni senza una stazione di coordinamento centralizzata si utilizza il sottolivello DCF. Il DCF ha come requisiti principali di evitare interferenze di trasmissioni simultanee, contendo il maggior numero possibile di connessione e gestendo il canale trasmissivo in modo equo. Non si vuole utilizzare un controllore centralizzato, senza un clock. Si utilizza un algoritmo csma/ca, "Carrier Senza Multiple Access/Collision Avoidance". Questo meccanismo determina se il mezzo trasmissivo è disponibile, in caso una stazione ha un pacchetto da spedire "Carrier Sense". Se il mezzo è occupato aspetta che la stazione sia libera prima di trasmettere. È possibile che due stazioni provino a trasmettere contemporaneamente, in questo caso si verifica una collisione edi pacchetti diventano intellegibili per i destinatari, e dovranno essere ritrasmessi. Per evitare le collisioni si utilizzano degli strumenti per evitarle il quanto possibile. Gli algoritmi backoff, acknowledgment,

DCF per effettuare il suo lavoro utilizza gli intervalli tra due pacchetti consecutivi per gestire delle priorità. Questo tempo è l'interpacket gap, che in ethernet è lungo 96 bit-time. Ci sono due tipi principali di IFS, "Inter-Frame Space", di tempo variabile tra i vari pacchetti: DIFS, "DCF IFS", in generale quello standard, e SIFS, "Short IFS" di lunghezza minore. Se un pacchetto consecutivo può essere trasmesso immediatamente, allora si utilizza il SIFS, altrimenti si utilizza il tempo di standard DIFS.

Per ora si considera il DCF senza il RTS/CTS, "Request To Send"/"Clear To Send" per evitare collisioni. Quando una stazione vuole trasmettere, ed il canale è occupato, capisce che è presente traffico nel canale, e si autolimita scegliendo un numero random, di backoff, nell'intervallo $[0, cw]$, ed incrementa il timer quando il timer solo quando è libero. La trasmissione può essere effettuata solamente quando il timer raggiunge il termine.

Quando una stazione rileva una collisione, utilizzando vari meccanismi, duplica il cw in modo che l'intervallo di casualità sia duplicato, in modo da rendere più improbabile collisioni future. Limitato superiormente al valore $cw_{\max}$. Se è un pacchetto viene inviato con successo, il valore di cw viene posto al valore minimo $cw_{\min}$. Valori ragionevoli per questo intervallo $[cw_{\min}, cw_{\max}]$ sono $[7, 31]$ e $[255, 1023]$.

Per determinare se il canale trasmissivo è libero, si utilizza un'altro meccanismo. In ogni pacchetto viene specificata la sua durata, nel campo "duration". Poiché il messo trasmissivo è condiviso, tutte le stazioni connesse ascoltano questo campo e si segnano la durata della trasmissione corrente nel vettore NAV, "Network Allocation Vector". Rappresenta un contatore per sincronizzare le stazioni. Ogni stazione lo decrementa con il passare del tempo, ed ogni stazioni può trasmettere solamente se questo campo vale zero. Questo campo deve essere presente come primo campo del pacchetto, per essere letto.

Ogni pacchetto spedito da una stazione deve essere riscontrato dalla stazione destinataria tramite un acknowledgment. Quindi una macchina quando calcola la durata della trasmissione, calcola il tempo di trasmissione, nota la banda ed il numero di bit del pacchetto, una durata SIFS, più corta

di DIFS, prima dell'invio dell'acknowledgment, e la durata di questo piccolo pacchetto. Il SIFS è l'interpacket space che separa un pacchetto dal suo acknowledgment. -

La verifica della disponibilità della rete può essere effettuata anche a livello fisico, controllando se nel canale trasmissivo è presente del segnale attivo. Se rileva un segnale attivo aspetta di trasmettere, quindi solo quando entrambe le condizioni fisiche e logiche sono verificate. Una trasmissione non si interrompe fino alla fine, e termina quando viene ricevuto il pacchetto ack. Se non viene ricevuto, oppure è intelligibile, allora si è verificata una collisione, e deve essere ritrasmesso il pacchetto. La stazione mittente si calcola quando dovrebbe ricevere l'acknowledgment. Di tutte le stazioni nella rete, solo la stazione trasmittente è in grado di accorgersi che si è verificata una collisione. In questo caso duplica il valore di *cw* ed aspetta prima di inviare nuovamente il pacchetto.

Se una stazione trasmettesse senza interruzioni, allora il mezzo trasmissivo non sarebbe mai libero. Questo rappresenta un attacco di tipo DOS "Denial Of Service".

Se il pacchetto che viene trasmesso è molto lungo, e si effettua una collisione, la stazione mittente se ne può accorgere solamente dopo il periodo di trasmissione di questo lungo pacchetto. Si perde molto tempo prima dell'identificazione della collisione. Per risolvere questo problema la dimensione massima dei pacchetti WiFi è di 1500 byte come in ethernet. Inoltre è possible che tra le stazioni ci sia visibilità parziale, ovvero può vedere solo una parte della rete.

Quando una stazione vuole trasmettere, invia un frame al destinatario, di breve lunghezza, chiedendo l'autorizzazione alla trasmissione. Se il destinatario è disponibile emette un breve frame di conferma. Alle stazioni vicine è richiesto di non interferire per l'intera durata della trasmissione che sta per avvenire. Questo meccanismo di prenotazione del canale tra due stazioni permette di evitare le collisioni. Se il canale è libero, a stazione mittente invia un pacchetto RTS per richiedere l'autorizzazione, e viene concessa alla stazione con un pacchetto CTS, e tutte le altre stazioni aspettano per il tempo presente nel campo duration dei pacchetti RTS e CTS, invece che nel pacchetto da trasmettere.

Una collisione si verifica quando all'invio del RTS, non viene inviato il CTS. Analogamente alle condizioni precedenti, dove al pacchetto non viene seguito l'ack.

Il valore di *cw* viene decrementato solamente se il canale è libero, quando NAV è pari a zero. L'algoritmo di backoff rappresenta un algoritmo distribuito su molte macchine.

Su ogni stazione WiFi è presente un parametro ch indica quale pacchetti da utilizzare. Viene utilizzato RTS/CTS per pacchetti di lunghezza maggiore a questo parametro s. In queste reti senza fili l'insieme delle stazioni appartenenti ad uno stesso BSS cambia continuamente. Per poter accedere ad un BSS si possono utilizzare protocollo di handshake in due modalità.

Si utilizza un protocollo di handshake per scambiare informazioni tra l'AP ed il BSS. Ogni stazione AP presente il proprio indirizzo MAC, potrebbe inviare un messaggio al BSS per indicare la sua presenza alla rete, tramite un pacchetto broadcast chiamato "beacon frame". Altrimenti la stazione potrebbe inviare pacchetti sonda di "probe", trasmettendo pacchetti "probe request", per esplorare la rete, ed attende un pacchetto di "probe response. "

Un amministratore può definire varie reti logiche sulla stessa rete fisiche, ciascuna identificata a un suo SSID. Solo chi ha gli opportuni permessi può accedere ad una carta rete logica. Si definisce ESS, "Extended Service Set" l'insieme delle stazioni appartenenti ai BSS di una rete e con lo stesso SSID.

Le stazioni nello stesso ESS, possono muoversi cambiando BSS.

Il livello MAC può decidere se frammentare un pacchetto e si occupa anche del riassemblaggio. In queste reti è consigliabile diminuire l'overhead di un pacchetto e ridurre la probabilità di collisione, diminuendo la dimensione dei singoli pacchetti. Ogni frammento del "MAC service data unit" viene frammentato e ciascuno di questi frammenti viene trattato come un pacchetto e viene riscontrato singolarmente. La dimensione dei frammenti può essere modificata dall'utente per guadagnare un vantaggio in trasmissione sulle altre stazioni. In ricezione il livello MAC ricompone questi frammenti in modo che gli strati superiori non si accorgono della frammentazione, né gli enti MAC che non sono coinvolti nella trasmissione.

In questo momento storico i riscontri di LLC non vengono utilizzati, quindi anche se nello standard è possibile che l'invio e ricezione di pacchetti ack sia effettuata a livello LLC.

Per inviare pacchetti di tipo broadcast, dove ToDS sia pari a zero, non si utilizzano ack e RTS/CTS, le eventuali collisioni non vengono quindi rilevate. Quando ToDS è pari ad uno, si utilizzano pacchetti di ack, ed in caso RTS/CTS, per i pacchetti diretti verso gli AP. Si può configurare l'AP in modo che ogni pacchetto broadcast ricevuto venga inviato o meno a tutto il BSS.

5 Livello 3: Il Livello di Rete

Il livello di rete sceglie un percorso per i pacchetti, conoscendo la topologia della rete, attraverso passaggi intermedi chiamati salti o hop tra varie stazioni.

Se il destinatario è nella stessa LAN del mittente, allora lo raggiunge direttamente, altrimenti deve percorrere diverse reti geografiche effettuano questi salti. Fino ad ora ci si è occupati di ogni salto separatamente tra la rete locale o la rete geografica attraverso un singolo filo, o mezzo trasmissivo. Per connettere tra di loro LAN e WAN si utilizzano reti di raccolta o di accesso.

5.1 Reti di Raccolta e Reti di Accesso

Nella LAN la trasmissione tar due stazioni, per quanto sia complessa la topologia della rete, è sempre diretta. Nelle reti geografiche (WAN), la trasmissione tra due stazioni a livello due avviene su un canale punto-punto. Viene realizzato solamente tramite un filo, non sono necessarie altre componenti della rete. Tipicamente questo filo viene realizzato da ethernet, anche se sviluppato principalmente per reti locali.

I collegamenti tra questi due tipi di rete vengono realizzati dall'ISP di cui si usufruiscono i servizi. Questa zona intermedia viene realizzata tramite reti di raccolta o di accesso. La rete di raccolta è la rete nella quale si raccoglie il traffico, mettendolo assieme, proveniente da tutte le stazioni coperte dal servizio dell'ISP. Le reti di accesso permettono, una volta raccolto il traffico, di accedere alla rete geografica vera e propria, composta da lunghi collegamenti punto-punto.

Per molti anni in quasi tutti i paesi del mondo, la rete di raccolta è stata la rete telefonica, due fili di rame, un doppino telefonico. Questa struttura molto pervasiva è stata il supporto delle comunicazioni di rete per molti anni. In seguito venne effettuata una progressiva sostituzione tra rame e fibra ottica, secondo una terminologia FTTx, "Fiber To The x", che distingue tra quanto vicino la stazione è vicina alla fibra ottica. La linea di tendenza punta a FTTH, "Fiber To The Home", dove la fibra ottica arriva direttamente alla stazione. Un altro metodo di raccolta simile è FTTB, che utilizza le esistenti strutture telefoniche per trasmettere i dati all'interno dell'edificio.

FTTN, "Fiber TO The Node", e FTTC, "Fiber To The Center", sono difficili da distinguere, questi rappresentano gli accessi xDSL o aDSL, quelli normalmente venduti, il caratteri prefisso di DSL descrive la trasmissione in termini di bilanciamento e traffico disponibile. In generale il cavo in fibra ottica raggiunge una stazione centrale che utilizza le preesistenti

Tipicamente gli accessi FTTH e FTTB sono a 1 Gb/s, recentemente si stanno vendendo accessi a velocità superiore. Queste strutture utilizzano la tecnologia GPON, "Gigabit-capable Passive Optical Network", il dispositivo vero e proprio di cui fanno uso è l'OLT, "Optical Line Termination", essenzialmente è uno switch. Permette di collegare fino a 64 clienti sullo stesso OLT. Al livello del cliente è presente un dispositivo router su cui si può connettere per accedere, questi sono collegati a vari livelli di splitting ottico, per connettere tutti i router dei clienti, connessi tutti su una porta di uno switch OLT. Questi dispositivi di splitting ottico non necessitano di alimentazione, ma unisce tra di loro il traffico proveniente da più clienti.

La banda di questi OLT arriva fino a 1 Gb/s, ma se più clienti vengono connessi alla stessa porta allora la banda effettiva per clienti diminuisce all'aumentare dei clienti, fino a 64 per porta. Questa è la rete con cui i provider attualmente raccolgono il traffico.

Per essere collegati alla WAN, si utilizza la rete di accesso, realizzata tramite switch OLT tutti collegati tra di loro, con un estensione geografica significativa. Ma è consigliabile non estendere troppo questa struttura, generalmente si indica come MAN. La maggior parte dei provider realizza la propria rete di accesso tramite un anello che connette le varie reti di raccolta e ciò che lo collega alla spina dorsale, la WAN, del provider è un nodo di aggregazione. Vengono realizzati in cicli per mantenere una certa resistenza ai guasti, e si utilizzano alberi ricoprenti di anello e loop-avoidance con tecnologie analoghe alle reti locali, per mantenere attivi solo i collegamenti strettamente necessari. Se si rompe un singolo switch, solo i clienti attestati a quello switch subiranno un disservizio. Generalmente vengono utilizzati più aggregation node, per mantenere un'importante resistenza ai guasti. Questo meccanismo di duplicare componenti può essere attuato a vari livelli per mantenere alta la resistenza del provider.

Molto spesso per aumentare la robustezza si utilizzano più anelli con doppi collegamenti, anche in fibra, che effettuano percorsi diversi. Si utilizza una nello poiché è la struttura contenente cicli più semplice possibile.

5.2 Indirizzo ed Instradamento

Il livello di rete fornisce servivi al livello di trasporto, che non è interessato della topologia delle varie reti attraversate per raggiungere una destinazione. Inoltre non conosce le tecnologie attraversate al livello due. Può offrire servizi connessi o non. Il protocollo di rete più diffuso IPv4 utilizza un protocollo di rete non connesso, utilizzano instradamento a datagramma, mentre un livello di rete connesso utilizza la commutazione a circuito virtuale. Ma questa distinzione non è strettamente mantenuta.

Il livello di trasporto deve conoscere degli indirizzi distribuiti in modo consistente su tutta la rete, in modo da poterle identificare univocamente tra rete locale e geografica. Una primitiva di servizio non connesso offerta al livello di trasporto indica l'indirizzo livello 3 del destinatario ed il suo payload, questo sarà ricevuto dal livello quattro corrispondente.

Dal punto di vista livello tre i sistemi possono essere "End System", ES, o "Intermediate System", IS. Ad ogni sistema viene associato un indirizzo numerico per poterlo identificare. Ad ogni sistema spesso viene associato un nome, la cui corrispondenza all'indirizzo viene gestita da server appositi presenti sulla rete.

Spesso queste apparecchiature intermedia si chiamano router o gateway, contengono almeno i primi tre strati della pila ISO-OSI, talvolta per motivi di gestione devono contenere anche livelli superiori. Alcuni casi di instradamento avvengono anche a livello due, anche se principalmente a livello 3.

L'indirizzo di livello due identifica un destinatario solamente all'interno di una LAN, mentre l'indirizzo di livello tre deve poter identificare il destinatario all'interno del'intera rete. Un sistema necessita di tanti indirizzi MAC quante sono le schede di rete al suo interno, ma è possibile associare una singola macchina ad un singolo indirizzo di livello tre, ma è possibile per alcuni protocolli, come IPv4 e IPv6, di essere associati ad indirizzi di livello tre differenti.

Esistono protocolli appositi per gestire e stabilire la corrispondenza tra gli indirizzi di livello due e livello tre. Ma gli indirizzi di livello due, i MAC address, sono univoci a livello globale, quindi si potrebbe utilizzare questi indirizzi. Il problema degli indirizzi MAC è che sono distribuiti

casualmente, ma sarebbe utile ai fini di individuare le stazioni se gli indirizzi associati a macchine vicine avessero caratteristiche simili. Poiché gli indirizzi MAC individuano univocamente la scheda di rete, mente gli indirizzi di livello tre possono cambiare, utilizzare solo i MAC comporterebbe problemi di privacy.

In linea di principio si possono individuare tre tipi di instradamento principali:

- Routing by Network Address: nel pacchetto c'è l'indirizzo del sistema destinatario, potrebbe non essere l'indirizzo dell'ES, ma di una sua interfaccia. Con commutazione a datagramma su questo indirizzo;
- Label Swapping: nel pacchetto non c'è l'indirizzo, ma un'etichetta che individua un cammino virtuale, con commutazione a circuito virtuale basata su queste etichette;
- Source Routing: nel pacchetto è indicata la lista ordinata di tutti gli IS da attraversare per raggiungere la destinazione.

5.2.1 Routing by Network Address

Quando il pacchetto raggiunge un IS, con varie linee di inoltro, questo guarda la sua tabella di instradamento locale e cerca come chiave di ricerca l'indirizzo del destinatario e restituisce la linea dove deve essere inoltrato il pacchetto. La commutazione è a datagramma poiché questa tabella può variare nel tempo. Questo rappresenta il modo più semplice di instradamento, operato dagli switch, dove la tabella di instradamento viene chiamata filtering database.

Se l'indirizzo non fosse presente nella sua tabella di instradamento, allora il pacchetto viene buttato, a differenza dei bridge, poiché su reti grandi l'inoltro di tutti i pacchetti sconosciuti su tutte le linee comporterebbero un aumento considerevole del traffico.

Sono necessari quindi dei protocolli per poter definire gli indirizzi e conoscere la topologia della rete, essendo dinamica. Questi meccanismi permettono di compilare automaticamente le tabelle di instradamento dell'intera rete. Ma è possibile che siano compilate manualmente, anche se non realistico. Ma questo processo di instradamento è indipendente dal chi ha compilato la tabella.

5.2.2 Label Swapping

Se due stazioni vogliono comunicare, in una rete a circuito virtuale, viene stabilito il percorso che deve attraversare il pacchetto. Le apparecchiature intermedie vengono quindi informate sul percorso che deve essere effettuato. Questo percorso viene identificato da un'etichetta in modo che all'arrivo del pacchetto presso un IS, si utilizza una tabella dove sono presenti le etichette ed i possibili percorsi, e quindi viene inviato su una certa linea. Questa tabella è locale e molto piccola, e determina la linea di ritrasmissione del pacchetto.

Deve essere stabilito il percorso tra le stazioni, analogamente alla tabella di instradamento per l'instradamento precedente. La differenza tra queste due tabelle è nelle loro dimensioni, infatti è necessaria una tabella che contenga tutti i destinatari per il meccanismo di instradamento precedente, mentre per una tabella delle etichette deve contenere solamente i possibili percorsi. Il numero di righe non è funzione del numero di stazioni globalmente presenti nella rete, ma dal numero di

circuiti virtuali che la attraversano in ogni dato istante. É irragionevole che in una rete tutte le stazioni trasmettano a tutte le altre stazioni.

Per evitare di dover generare etichette uniche in tutta la rete, e quindi verificarne la disponibilità, ad ogni tratto del percorso viene assegnata un'etichetta diversa localmente. In questo modo numero di etichette disponibili non è un limite superiore al numero dei circuiti virtuali attivabili. Queste etichette sono contenute in un campo nell'intestazione del pacchetto.

5.2.3 Source Routing

Nel pacchetto viene specificata la lista delle stazioni da attraversare per raggiungere la destinazione. Il ruolo dell'IS deve solo leggere questa lista e determinare la stazione successiva al quale inoltrare il pacchetto.

5.3 Router

L'architettura di un router è formato da un algoritmo per calcolare la tabella di instradamento, la tabella stessa ed il processo di inoltro dei pacchetti. In questo corso non si occupa dell'algoritmo di creazione automatica della tabella di instradamento.

Ciò che costituisce la tabella di instradamento ed il processo di ritrasmissione di un pacchetto rappresenta il "data plane", mentre l'algoritmo di costruzione della tabella di instradamento consiste nel "control plane".

In un router multiprocollo sono presenti diversi di queste pile di data e control plane per ogni protocollo, mantenendo le stesse interfacce. Un router può partecipare contemporaneamente a diverse tecnologie, senza che queste diverse tecnologie e protocolli conoscano la loro presenza. Se ad un router arriva un pacchetto destinato ad un protocollo che non può gestire, viene buttato.

5.4 Il Protocollo TCP/IP

Il protocollo di rete IPv4, "Internet Protocol version 4" appartenente alla "suite TPC/IP" o "Internet Protocol Suite", è il principale protocollo di rete di livello 3. Quest Internet Protocol Suite è la pila protocollare utilizzata dall'internet, implementata nella quasi totalità dei computer. Questi protocolli sono descritti nei documenti RFC.

Alla fine degli anni '70 nasce l'Internet Protocol Suite con "Transmission Control Protocol" o TCP, ed il protocollo internet IPv4 con una prima versione di internet chiamata arpanet. Da allora è in costante crescita. Già dai primi anni della sua storia arpanet si permetteva connessioni tra le due coste oceaniche degli Stati Uniti. Tra protocolli di TCP/IP abbiamo IPv4, IPv6, ICMP ed ARP, protocolli di supporto, e protocolli di routing, di control plane per costruire le tabelle di instradamento automaticamente.

Le architetture che si possono realizzare utilizzando l'internet protocol suite sono possibili in vari tipi di stratificazione.

QUando nasce la TCp/IP, nasce secondo i principi progettuali della semplificazione estrema del data plane, in modo che l'inoltro dei pacchetti sia effettuato nel modo più semplice possibile. Tutte le funzioni complesse vengono spostate sugli es. Il routing è effettuato dall'indirizzo di rete e viene realizzato tramite commutatone a datagramma. Il servizio offerto al livello 4 è non connesso. Il

servizio è di tipo "best effort", ovvero un pacchetto può essere corrotto, perduto o consegnato in ordine errato.

Il principio della vita stretta, o "narrow waist", dove i pacchetti a livello 3 sono tutti realizzati con IPv4 in modo molto snello, per permettere un'enorme interoperabilità tra molti apparati, poiché è il protocollo parlato da tutti.

IPv4 è attualmente, insieme ad IPv6, il protocollo principale di livello tre, descritto nell'RFC 791. I pacchetti di questo livello si chiamano datagrammi o datagram. Provvede a funzioni di instradamento, il motivo per cui nasce il livello tre, ma svolge anche la funzione di frammentazione, riassemblaggio e rilevazione degli errori. Per i pacchetti la frammentazione era compito del livello quattro, ma già nelle reti WiFi i pacchetti venivano frammentati, ma nessuno all'esterno della rete WiFi può rilevare la frammentazione. La frammentazione offerta dal livello quattro compie una riframmentazione, per i pacchetti già frammentati al livello tre poiché troppo grandi. Al contrario di una rete WiFi, nella rete di livello tre i frammenti vengono rilevati.

I pacchetti di livello tre devono essere contenuti all'interno di un pacchetto di livello due, di campo dati di dimensione 1500 byte. Per cui datagram più grandi di questa dimensione devono essere frammentati. Questo rappresenta un vincolo sulla dimensione del datagram, quindi vengono frammentati solo se devono frammentarlo. Provvedere al routing deve essere semplice, quindi all'arrivo di frammenti di un datagram, questi vengono inoltrai per semplificare le azioni di un router.

I pacchetti di livello datagram Ipv4 è diviso in due parti, il datagram header ed il campo dati.

La convenzione per rappresentare i pacchetti in RFC è di rappresentare i pacchetti striscia per striscia, di lunghezza di 32 bit. Queste strisce da 4 byte vengono chiamate "long word". Il primo campo che si trova è la versione del protocollo che ha generato il pacchetto, ma è possibile che venga utilizzato da versioni successive, ma questo non è possibile per i pacchetti IPv6.

l'identificatore ident, flag e fragment offset sono campi appositi per frammentar eil pacchetto. Il campo protocol indica il protocollo di livello superior, il campo header checksum è analogo al campo RCS del livello 2. In seguito si specificano il mittente ed il destinatario del pacchetto. In seguito si possono specificare campi addizionali opzionali nel pacchetto. Le opzioni devono obbligatoriamente essere specificate in multipli di 32 bit. Per cui si può aggiungere del padding per allinearsi a 32 bit. In seguito vengono inseriti i dati del pacchetto.

Questo formato è il linguaggio di quasi tutti gli es ed is del mondo. Non tutti poiché insieme ad Ipv4 si sta affermando il protocollo IPv6.

I campi versione e ihl sono entrambi di 4 bit. Il primo campo dovrebbe permettere la migrazione tra due versioni del protocollo chiara, lenta e controllata. Dato un protocollo che evolve nel tempo deve essere effettuata un'operazione di parsing, strutturando il pacchetto in campi, questo parser è diverso per ogni versione. Quest'idea di utilizzare un campo versione viene utilizzata anche per altri pacchetti di rete. L'"Internet Header Length" o ihl specifica la lunghezza dell'header, multipli di 32 bit. Al minimo contiene 5 long word. In un pacchetto ethernet ci sono al massimo 1500 byte, di cui 20, al minimo, vengono utilizzati dall'header, i restanti 1480 sono disponibili a protocolli di livello superiore. Il campo "type of service" specifica la priorità del datagramma rispetto ad altri, ed è diviso in due parti DSCP, "Differentiated Services Code Point", e ECN, "Explicit Congestion Notification", utilizzati per definire la priorità del pacchetto.

QUando i pacchetti arrivano in una macchina sono tutti uguali, ma per differenziarli tra di loro si utilizza questo campo per indicare la priorità del pacchetto in modo che gli apparati intermedi siano in grado di effettuare le adeguate. Questi campi priorità vengono specificati da chi ha realizzato quel pacchetto di livello 3, ma dipende dal provider. Non tutti i provider sfruttano questa possibilità, se la sfruttano non permettono all'utente di specificarlo, ma la impostano loro la priorità. I provider che utilizzano questo strumento possono realizzare offerte commerciali utilizzando questo approccio, assegnando livelli di priorità diversi. Per l'utenza domestica questo tipo di offerta non è convenzionale, ma tipicamente per aziende questo tipo di offerte è molto comune.

La possibilità di rispettare la priorità può essere garantita solamente dagli apparati di loro appartenenza, i router, ma questo non è garantito per tutto il percorso del pacchetto attraverso la rete. Quando questo pacchetto con priorità arriva in un'area geografica gestita da un altro ISP, questo non ha obbligazioni commerciali rispetto all'utente, e quindi questi pacchetti vengono gestiti indipendentemente dalla loro priorità. Quindi queste offerte sono valide solamente per pacchetti che rimangono all'interno della zona di interesse del ISP.

I campi ident, flags e fragment offset hanno scopi diversi. Il livello quattro per realizzare pacchetti di una certa dimensione, richiede al livello tre che notifica il livello due e viene restituito il valore massimo di MTU, "Maximum Transmission Unit". Quindi il livello di trasporto realizza pacchetti in base alla tecnologia immediatamente sottostante. Ogni livello due attraversato da uno stesso pacchetto potrebbe avere diversi valori di MTU, quindi il pacchetto di livello tre potrebbe essere più grande ed è possibile sia necessario frammentare il pacchetto. Questi frammenti attraversano gli is, senza che questi vengano rilevati dagli is, e verranno riassemblati solamente all'arrivo all'es. IPv6 effettua una scelta radialmente diversa, ed alcune di queste scelte migliori vengono implementate anche da IPv4.

Occorrono dei campi nel pacchetto per riconoscere i pacchetto come frammento, riconoscere frammenti generati dallo tesso pacchetto e deve essere possibile determinare l'ordine dei frammenti per poterli riassemblare. In caso il frammento di livello due deve essere nuovamente frammentato si utilizza una semplice tecnica di realizzare un offset rispetto al pacchetto originale. Si inserisce la posizione del pacchetto originale del byte. Questo valore è univoco, se ci saranno ulteriori frammentazioni questi avranno numeri di offset diversi. Il campo ident, di 16 bit, identifica un datagramma e serve all'es per determinare il pacchetto di appartenenza del frammento. Se il pacchetto non è frammentato questo rappresenta bit sprecati. Il campo di 13 bit può specificare solo un offset multiplo di otto, con la convenzione che tutti i frammenti sono multipli di otto, in seguito nel campo flag di 3 bit, specifica se il pacchetto può essere frammentato. Il primo bit deve esser e riservato e sempre pari a zero, altri bit DF e MF permettono di non frammentare, il primo e di specificare che sia l'ultimo pacchetto, l'ultimo. Se un pacchetto non può essere frammentato, allora si preferisce buttarlo. Oltre all'identificatore per stabilire l'univocità del pacchetto si può utilizzare il pacchetto mittente.

Sull'ultimo frammento si può inserire un numero di byte arbitrario, poiché non è necessario l'offset, ma specificato nel campo MF. Quando vengono realizzati protocolli vengono inseriti bit o gruppi di bit non utilizzati, per permettere ad evoluzioni future per cogliere circostanze non ancora rappresentate dal protocollo.

Il campo "total length", di 16 bit, indica la dimensione massima del pacchetto, al massimo di 65553 byte, ma molto spesso è frammentato e minore di questo valore. Il campo "time to

live", di 8 bit, rappresenta il tempo di vita del pacchetto, e quando raggiunge zero viene scartato, ed il router invia una avvertimento al mittente. Il valore contenuto viene decrementato ad ogni salto o hop. Quindi un pacchetto in internet può effettuare al massimo 255 salti. Nonostante la dimensione della rete internet, l'elevata connettività permette di effettuare pochi salti. Viene imposto un limite per impedire che alcuni pacchetti entrino in loop. Non deve essere possibile in nessun caso che un pacchetto giri in eterno sulla rete. Il campo protocol di 8 bit identifica il protocollo del livello superiore che è nella payload del pacchetto IIP, i possibili valori vengono definiti dalla IANA. Molto spesso il pacchetto di livello superiore contenuto è di protocollo TCP, con valore 6. Altri 16 bit identificano il campo checksum, di intestazione e deve essere ricalcolata ad ogni hop. Le tecnologie più recente impedisce che venga ricalcolata ad ogni hop. I campi opzionali possono specificare quando sia segreto il pacchetto, si può specificare il source routing, specificando il cammino completo, senza interpellare la tabella di instradamento. Si può specificare un'opzione meno stringente per indicare la sequenza di router obbligatoria da attraversare. Si può indicare che il router attraversato specifica il proprio indirizzo oppure che specifici anche un timestamp. Non è obbligatorio per gli is di rispettare queste operazioni, poiché sono operazioni aggiuntive che richiedono calcolo ulteriore e potrebbero rallentare le operazioni.

5.4.1 Indirizzamento IPv4

Gli indirizzi sono associati alle schede di rete e non agli indirizzi host, sono di 4 byte ed univoci a livello mondiale. Per comodità sono rappresentati da quattro numeri decimali, separati da una virgola, spesso gli vengono assegnati nomi simbolici.

Gli indirizzi vengono assegnati alle macchine in un modo specifico per facilitare la commutazione. Indirizzi sulla stesse rete locale condividono il prefisso, un gruppo di macchine è quindi identificato dal suo prefisso. Non sono assegnati in modo casuale, ma le stesse zone geografiche hanno gli stessi indirizzi. Nei router quindi si possono inserire delle tabelle di indirizzi, raggruppati in "net". Indirizzi nella stessa net sono raggruppati su una stessa rete fisica.

La convenzione consiste di completare i prefissi con tutti zeri per realizzare un indirizzo IPv4 legittimo da poter utilizzare. Questo valore ottenuto si utilizza per denotare l'intera net e LAN, e non può essere utilizzato per assegnare alcuna delle interfacce della LAN.

L'indirizzo IPv4 ha due funzioni, di identificare un'interfaccia nella rete e di locazione di gruppi di indirizzi in posizione geografiche omogenee nella rete. Nei primi anni di internet gli indirizzamenti IPv4 erano divisi in cinque classi, con prefissi di diversa lunghezza, rappresentate da cinque lettere A, B, C, D, E. Se l'indirizzo iniziava con il primo bit a zero, allora la sua parte prefisso si estendeva al primo byte, mentre i restanti sono un identificatore delle macchine host sulla LAN. Se il primo bit è uno ed il secondo è uno zero, allora il prefisso si estende al primo ed al secondo byte. Quando il terzo bit è a zero il prefisso si estende fino al terzo byte. Questi tipi di indirizzi si chiamano di classe A, B e C. Gli indirizzi di classe D hanno il terzo bit pari a zero e individuano indirizzi di tipo multicast. La classe D con il quarto bit a zero viene riservata ad usi futuri.

Il problema di questo approccio consiste nello spreco di byte per la prima classe, poiché non esiste una LAN contenente 2^{24} indirizzi. Per cui si è introdotto l'indirizzamento IPv4 classless che utilizza un'altra sequenza di bit accanto all'indirizzo chiamata "netmask", questo consente una lunghezza arbitraria della parte prefisso. Ad ogni indirizzo viene associata un'indicazione che

specifica la lunghezza del prefisso attraverso il numero di uno che la compongono e quella delle macchine dal numero di zeri che la compongono. Una netmask è una sequenza di 4 byte. Queste vengono normalmente scritte nella documentazione di rete come numeri decimali, con la stessa convenzione degli indirizzi IPv4. La netmask di un prefisso si può rappresentare come una barra ed un numero specificando dopo l'indirizzo la lunghezza del prefisso. Due indirizzi IPv4 che si trovano sullo stesso prefisso hanno la stessa netmask, generano lo stesso valore se messi in and bit a bit con la netmask relativa. Due indirizzi appartenenti a diverse LAN producono risultati diversi se messi in and bit a bit sulla stessa netmask, se gli indirizzi vengono assegnati in modo che siano tutti disgiunti tra di loro. La responsabilità di assegnare gli indirizzi viene assetata ad organizzazioni internazionali come al RIPE.

La configurazione delle interfacce degli es prevede la specifica della netmask della net di appartenenza, per cui mettendo in and l'indirizzo del pacchetto in arrivo, l'es è in grado di identificare se si tratta di un indirizzo locale o remoto. Se il destinatario è locale viene inviato tramite trasmissione diretta al destinatario, se è remoto invece, viene inviato al default gateway della LAN, un router particolare. Il problema in questo approccio è che non si conosce la netmask del destinatario e si ricava il prefisso del destinatario dalla netmask dell'es, l'ipotesi che gli indirizzi vengono assegnati in modo disgiunto permette di risolvere questo problema di ricavare il prefisso sbagliato applicando l'and bit a bit tra la netmask e l'indirizzo destinatario. L'unico caso che quest'ipotesi non risolve è quando il prefisso calcolato del destinatario è diverso da quello reale, ma comunque diverso da quello locale. Questo non rappresenta un problema poiché il prefisso individuato è remoto, quindi verrà inviato indipendentemente al router. Non da errore dal punto di vista della località o non località del destinatario di un pacchetto.

Dentro ad un router è presente una tabella di instradamento che controlla tramite un'azione chiamata "IP lookup". Questa tabella contiene singoli prefissi, che vengono confrontati bit a bit con il prefisso locale, sulla relativa netmask. Non appena si rileva un riscontro il pacchetto viene inviato su quella linea, altrimenti il pacchetto viene buttato. I prefissi vengono ordinati rispetto alla loro lunghezza. Sulla riga 0.0.0.0/0, chiamata rotta di default, è presente viene prodotto un riscontro con ogni indirizzo, e nessun pacchetto verrà mai buttato. Si utilizza questa rotta di default poiché la tabella di instradamento non può contenere tutti i prefissi dell'intero internet. Oltre alla linea di inoltro rappresentata dall'interfaccia del router, è presente un altro campo chiamato "next-hop" che specifica a quale router inviare il pacchetto in base all'indirizzo del destinatario, per distinguere pacchetti destinati alla stessa linea di inoltro. Tuttavia sono presenti eccezioni notevoli su questo schema.

Per realizzare un indirizzo non si può utilizzare né il numero 0 né il numero 255.

Le operazioni eseguita da un es per l'inoltro di un pacchetto possono essere pensate in termini di accesso ad una tabella di instradamento. Può essere considerato come una macchina con una tabella di instradamento con informazioni veramente essenziali.

Su ogni macchina l'interfaccia lo di loopback appartiene ad un indirizzo 127.0.0.0/8, ed il pacchetto rimbalza e torna indietro. Tramite quest'interfaccia due applicazioni su una stessa macchina possono comunicare disaccoppiandole fortemente, quindi senza utilizzare l'API, solamente attraverso dei pacchetti. Questo indirizzo è un prefisso, non è un vero indirizzo, l'indirizzo attribuito a quest'interfaccia di loopback corrisponde di default al primo indirizzo disponile: 127.0.0.1. si possono inserire più interfacce di loopback su una stessa macchina, sempre all'interno dello stesso

prefisso.

Alcuni indirizzi hanno un significato particolare, il primo consiste da tutti 1, dopo il prefisso, e rappresenta un indirizzo broadcast sulla LAN, per questo non si può utilizzare solamente uno all'interno di un indirizzo. Indirizzi di prefisso 10.0.0.0/8 sono indirizzi privati non validi in internet, che possono essere utilizzati per comunicare all'interno di una LAN, possono essere associati ad indirizzi validi per comunicare verso l'esterno. L'indirizzo 172.16.0.0/12 rappresenta macchine visibili solo localmente, analogamente indirizzi 192.168.0.0/16 sono indirizzi utilizzabili nella comunicazione tra macchine nella stessa LAN. Il provider trasforma gli indirizzi privati in indirizzi validi per comunicare in internet.

In un processo chiamato "sub-netting" è possibile spezzare l'indirizzo privato 10.0.0.0/8 per permettere di comunicare tra di loro varie macchine di una stessa azienda o organizzazione.

Le organizzazioni che assegnano gli indirizzi utilizzano un meccanismo che divide gli indirizzi in grandi pezzi da /8, assegnati ad aree geografiche e delle organizzazioni delegate ad ogni area geografica assegnano gli indirizzi a chi lo richiede. Quando finiscono la loro sezione di spazio di indirizzi, richiede all'IANA, "Internet Assigned Numbers Authority" l'organizzazione globale per assegnare gli indirizzi IPv4. Attualmente questi indirizzi si stanno esaurendo quindi queste organizzazioni potrebbero non essere in grado di assegnare gli indirizzi richiesti.

Gli es tendono ad avere più schede di rete, sulla stessa macchina, ma anche se sono presenti più schede l'IPv4 ne utilizza una sola alla volta, mentre le altre sono inattive. Questo poiché alla sua creazione, non era immaginabile che una stessa macchina potesse contenere più schede di rete. Per cui i costruttori di macchine inseriscono una gerarchia tra le varie schede di rete. Anche se tante applicazioni cercano di utilizzare più schede di rete in parallelo su un singolo es.

5.4.2 ARP

Questo protocollo di terzo livello, definito nella RFC 826 per associare gli indirizzi MAC ad indirizzi IPv4. Questo protocollo risolve il problema della traduzione tra questi due indirizzi.

Se bisogna inviare un pacchetto all'intero di una rete locale bisogna effettuare una consegna diretta, ovvero spedire il pacchetto direttamente alla macchina del destinatario. Questa consegna viene effettuata specificando al livello 2 l'indirizzo MAC del destinatario, di cui è noto solamente l'indirizzo IPv4, quindi si utilizza il protocollo ARP. ARP invia un pacchetto broadcast, "ARP request", su tutte le macchine della LAN, e risponderanno solamente le macchine aventi quell'indirizzo MAC. Tutte le macchine sulla LAN sono costrette ad aprire i pacchetti, quindi viene generata un'interruzione hardware e deve essere gestita dal sistema operativo anche se il pacchetto non appartiene alla macchina. Ogni scheda di rete è costretta ad analizzare ed aprire i pacchetti che arrivano ad essa.

Questo processo di broadcast provoca un'interruzione a bordo di tutti gli host della LAN, e passano attraverso gli hub e gli switch, vengono fermati solamente dai router.

Per evitare questo processo costantemente si utilizza un processo di caching, dove gli indirizzi noti vengono memorizzati. Quando un host riceve un pacchetto ARP request dedicato a sé stesso invia un'"ARP reply". Gli host utilizzano ARP hanno a disposizione una cache dove si possono salvare queste corrispondenze tra IPv4 e ARP.

Su una macchina Linux il comando `arp -a` permette di visualizzare la lista di queste associazioni.

Questi pacchetti sono contenuti direttamente nel pacchetto di livello 2, quindi sono presenti gli indirizzi IPv4 e MAC dei destinatari e mittenti. I campi hardware e protocol specificano il tipo di indirizzi di livello due e di livello tre, mentre o campi hlen specificano la lunghezza dei due indirizzi, definito in questo modo modulare poiché si pensava che la lunghezza degli indirizzi fosse cambiata. Si utilizzano quindi campi di lunghezza variabile per memorizzare questi indirizzi che possono variare.

Mentre se l'indirizzo è remoto, il protocollo ARP allo stesso modo deve individuare l'indirizzo MAC del router. Questo protocollo ARP appartiene al livello tre della pila ISO-OSI, ed è un protocollo di supporto per l'IPv4. Quando un pacchetto viene ricevuto da un router, vengono effettuate le stesse operazioni descritte per gli es, per individuare l'indirizzo MAC della macchina o del router, se il pacchetto è in connessione diretta oppure è per l'esterno.

Per ogni salto del pacchetto è presente un ARP request, ARP reply e l'invio del pacchetto, se l'indirizzo MAC non è contenuto nella cache della macchina host.

5.4.3 Il Protocollo ICMP, per IPv4

Il protocollo ICMP, "Internet Control Massage Protocol" per IPv4 realizza due operazioni per le reti, ed ha un approccio best effort, effettua dei tentativi al meglio delle sue capacità, altrimenti il pacchetto viene buttato.

Provvede ad un piccolo servizio di rilevazione degli errori. Inoltre consente di inviare, e richiedere pacchetti ed ottenere informazioni. Mentre ARP viaggia direttamente dentro il livello 2, i pacchetti ICMP viaggiano dentro pacchetti IPv4.

Le regole sul comportamento di questo protocollo vennero definite nella RFC 792.. La prima regola impone che nessun messaggio ICMP viene generato a seguito di rilevati errori su altri messaggi ICMP. La seconda regola impone che se il pacchetto viene frammentato, solo il primo frammento può generare un messaggio di errore ICMP. La terza regola impone che i pacchetti broadcast e multicast non generano pacchetti ICMP. Si utilizzano queste regole per non generare

Il pacchetto ICMP viaggia all'interno del pacchetto dati di un pacchetto IPv4, poiché se viene sollevato un pacchetto di errore, deve poter tornare, quindi deve necessariamente viaggiare dentro un pacchetto IPv4.

I messaggi di primo tipo sono messaggi di "destination unreachable", in questo caso i pacchetti vengono scartati poiché la destinazione non è raggiungibile per diversi motivi:

- "network unreachable": un gateway deve la rete dov'è destinato il pacchetto a distanza infinita, un router sta entrando nella tabella di instradamento e l'indirizzo non effettua alcun match;
- "host unreachable": l'host a cui è destinato il pacchetto non risponde ad una chiamata ARP. Il router a cui è arrivato il pacchetto tenta di effettuare una chiamata ARP per inviare direttamente il pacchetto, ma questa fallisce quindi scarta il pacchetto;
- "protocol unreachable": dentro il pacchetto IPv4 è presente un campo che identifica il protocollo a cui va mandato il pacchetto, se questo non è individuato allora viene sollevato questo messaggio di errore;

- "port unreachable": la "port" a cui è spedito il pacchetto non è raggiungibile;
- "fragmentation needed and DF set": se il pacchetto non può essere frammentato, anche se deve essere frammentato per passare attraverso la MTU, quindi viene scartato.

Messaggio di tempo scaduto, se il valore del TTL ritorna a zero, il pacchetto viene scartato con un errore di tipo `TIME_EXCEEDED`. Messaggio di redirezione, può essere inviato da un router per indicare dove bisogna reindirizzare il traffico, indicando all'es una strada più favorevole. Messaggio di eco, ad un messaggio di `ECHO_REQUEST` può essere inviato un `ECHO_REPLY`, una richiesta ed una relativa risposta edi eco, si utilizza per controllare la raggiungibilità, dove sono presenti dei `TIMESTAMP` e `TIMESTAMP_REPLY` per fornire informazioni sull'orario di invio e misura la velocità del collegamento.

Il comando `ping`, seguito da un indirizzo, permette di effettuare quest'ultima operazione: invia un pacchetto ICMP di eco ad una macchina e questa, se è raggiungibile, tenta di inviare i pacchetti di ritorno ICMP. Questo comando è disponibile nella quasi totalità dei sistemi, ed ha l'obiettivo di verificare se un dato indirizzo IP è raggiungibile ed il ritardo necessario per raggiungerlo.

Quando termina il comando di ping viene visualizzato un report che indica il numero di pacchetti inviati, persi, il tempo totale impiegato, ed il tempo di andata e ritorno in media, al massimo e la deviazione media. Il comando ping è realizzato con una `fork()`, dove il primo si occupa di lanciare i pacchetti, mentre il secondo riceve i pacchetti. Quando i pacchetti vengono inviati o ricevuti questi vengono memorizzati. I pacchetti inviati si possono visualizzare con il comando `tcpdump` specificando solamente i pacchetti di tipo ICMP:

```
> tcpdump -n "icmp"
```

Se si effettua un ping su un indirizzo broadcast, considera solo la risposta della prima macchina, mentre tutti gli altri messaggi di ritorno provenienti dalle altre macchine vengono considerati come duplicati. Per chiudere l'esecuzione del comando si utilizza la sequenza "Ctrl + C".

Il traceroute è disponile nella quasi totalità delle macchine, attraverso questo comando si vuole determinare quali sono i router effettivamente attraversati per raggiungere una destinazione IP. Quest'operazione viene eseguita inviando pacchetti di ttl molto basso, incrementandolo ad ogni pacchetto, in modo che i router che scartano il pacchetto inviano il pacchetto `TIME_EXCEEDED` alla macchina mittente. In questo modo si può ottenere progressivamente la sequenza di router che il pacchetto deve attraversare per arrivare all'indirizzo IP destinatario. Nella commutazione a datagramma, i diversi pacchetti potrebbero attraversare strade diverse, ed i vari pacchetti della sequenza potrebbero attraversare router diversi. Quindi non è garantito che l'informazione così ottenuta sia corretta. Sono sicuramente corretti per i percorsi dei singoli pacchetti, ma potendo variare nel tempo il percorso, non è garantito sia lo stesso per tutti i router.

Il comando nei sistemi Linux si chiama `traceroute`, e `tracert` su Windows, seguito dall'indirizzo IP di destinazione. Il tempo di ritardo è variabile poiché i router non hanno come compito principale l'inoltro dei messaggi di errore. Quindi quando il router non sta inoltrando altri pacchetti può gestire i pacchetti scartati ed inoltrare i pacchetti ICMP.

Quando i pacchetti di risposta non vengono rilevati dopo un tempo definito di pochi secondi, vengono segnati tramite il carattere `*`. I router che non inviano i pacchetti ICMP sono router che non si vogliono mostrare, quindi scartano i pacchetti senza inviare ICMP. Questi router generalmente

si trovano tra i primi hop, di accesso diretto alla rete dell'ISP. Il contenuto dei pacchetti inviati è arbitrario, poiché sono destinati ad essere buttati. Ma generalmente si utilizzano due tipi di pacchetti, una echo request, a cui il destinatario risponde con una echo reply, oppure viene inviato un pacchetto UDP, un protocollo di livello quattro.

Effettuando un'operazione `tcpdump` si possono osservare i a gruppi di tre i pacchetti inviati con i rispettivi ttl. Studiare le reti corrisponde a studiare le varie apparecchiature e gli strumenti di supporto che popolano le reti.

5.4.4 IPv6 e ICMPv6

Gli indirizzi IPv4 si stanno esaurendo, già nel 2010 la IANA ha allocato le ultime /8 disponibili attribuendole ai registri continentali. Alla fine del 2020 il registro europeo ha allocato i suoi ultimi indirizzi disponibili. Gli ultimi indirizzi sono essenzialmente esauriti, ma ci sono registri più avanti di altri nella distribuzione. IPv6 è alternativo rispetto a IPv4, per loro natura realizzano due reti disgiunte, che non possono comunicare tra di loro. IPv6 ha varie caratteristiche interessanti, inoltre ha indirizzi di 128 bit, e sta progressivamente sostituendo gli indirizzi IPv4. IPv5 non è mai esistito, poiché venne realizzato un protocollo specifico dal punto di vista applicativo, ed utilizzava un nuovo protocollo di livello 3 il cui nome poteva sembrare IPv5 quindi per evitare confusione venne chiamato IPv6. IPv4 e IPv6 sono destinati a convivere per anni, entrambi protocolli di livello tre, i loro pacchetti vengono gestiti separatamente dagli es e is, e per mantenere l'internet rimarrà per molto tempo IPv4.

Un pacchetto IPv6 ha un header di struttura fissa, per evitare di creare problemi ad eventuali router attraversati, di 40 byte. Alcune caratteristiche di IPv4 spariscono o cambiano nome e vengono introdotti nuovi campi. Il campo "Ver" versione di 4 bit, nei pacchetti IPv4 non ha funzione, poiché non vengono distinti con questo campo da IPv6, ma dal campo LLC. Il campo "Traffic Class" di 8 bit ha sostanzialmente ele stesse funzioni di Type of Service di IPv4. Il campo "Flow Label" di 20 bit viene utilizzato per distinguere un flusso, ma ancora non ha uno scopo definito. I router attraversati dovrebbero gestire pacchetti dello stesso flusso allo stesso modo. É un campo il cui utilizzo non è ancora chiaro a livello applicativo, ma è utile per certi aspetti di sicurezza. Il campo "Payload Length" di 16 bit è analogo al campo Length in IPv4. Il campo "Next Header" di 8 bit individua il protocollo di livello superiore contenuto, simile al campo Protocol in IPv4. Si utilizza in IPv6 anche per specificare delle opzioni, inserendo in questo campo una codifica per indicare le opzioni contenute all'interno del pacchetto. Il campo "Hop Limit" è il ttl, di 8 bit, come in IPv4. I pacchetti IPv6 non possono essere frammentati, quindi non sono presenti i campi in IPv4 corrispondenti alla frammentazione. Se questi pacchetti incontrano un router con un MTU minore vengono scartati. La frammentazione deve essere effettuata a livello degli es, esistono IPv6 protocolli che individuano in un cammino la MTU minima, e quindi l'es è in grado di creare pacchetti di lunghezza adeguate per passare attraverso l'intero pacchetto. Si suppone che i controlli di livello 2 e 4 siano sufficienti, quindi sparisce la checksum di IPv4, non è presente il campo che specifica la lunghezza dell'intestazione, poiché la lunghezza è fissa in IPv6. In termini di intestazione è un protocollo semplice, rispetto ad IPv4.

Gli indirizzi IPv6 sono molto diversi da IPv4, avendo molti bit in più sarebbe più difficile rappresentarli in decimale, quindi vengono rappresentati in 8 numeri esadecimali separati da due punti,

ogni numero rappresenta 16 bit. Questa rappresentazione consente di non effettuare conversioni di base, da binario a decimale e viceversa.

La scrittura può essere semplificata, rappresentando zeri consecutivi con un singolo zero, solo se antecedenti da altri zeri. Generalmente si omettono gruppi consecutivi di 16 bit contenenti soltanto zeri, si omettono anche i due punti di separazione. Questa notazione è utilizzabile una sola volta all'interno di un indirizzo, per evitare ambiguità. L'interfaccia di loopback è ::1, e rappresenta l'indirizzo 0:0:0:0:0:0:0:1, analogo di 127.0.0.1 in IPv4.

Per i prefissi si utilizza la stessa notazione "/" usata in IPv4, con un prefissi di lunghezza massima di 64 bit. Una netmask non è rappresentata mai nel formato dove sono espliciti i suoi bit. Si usa solamente la notazione con la barra /.

Anche in IPv6 ci sono indirizzi unicast e multicast, ma non sono presenti indirizzi broadcast. Si cerca di stabilire con chi si vuole dialogare in maniera più selettiva. Tuttavia questa regola ha delle interpretazioni piuttosto lasche. Due tipi fondamentali di indirizzi unicast sono "Global Unicast", indirizzi utilizzabili i internet. esistono indirizzi simili agli indirizzi IPv4 non utilizzabili in internet, con qualche differenza significativa, chiamati "Link-Local", utilizzabili solo nell'ambito della LAN. In IPv6 indirizzi Link-Local sono usati sempre, a differenza degli indirizzi privati in IPv4, e permettono soltanto di dialogare con altre macchine sulla stessa LAN. Nel mondo IPv6 link e LAN sono sinonimi, quindi non vengono chiamati LAN-Local.

Indirizzi multicast importanti sono "Solicited Node", ff02::01, tutte le macchine in una LAN, essenzialmente un indirizzo broadcasts, e ff02::02, tutti i router in una LAN.

I primi 64 bit sono il prefisso, mentre i seguenti 64 bit vengono chiamati "Interface Identifier", o "Interface ID", per identificare univocamente un'interfaccia. Questo comporta uno spreco, poiché anche se su una singola LAN sono presenti poche macchine, vengono comunque utilizzati 64 bit per indirizzarle.

Indirizzi Global Unicast vengono assegnati da organizzazioni e registri internazionali, analogamente ad indirizzi IPv4.

Indirizzi Link-Local hanno un prefisso fe80::/64

In IPv6 un'interfaccia di rete può avere diversi indirizzi, e gli spazi di indirizzamento attribuiti a due LAN devono essere disgiunti. Gli indirizzi possono essere attribuiti alle interfacce in modo manuale oppure automatico, tramite vari meccanismi built-in nel protocollo per assegnare questi interfacce. Uno di questi è particolarmente interessante e si chiama auto-configurazione "stateless". Con questo meccanismo una macchina si attribuisce un indirizzo per comunicare nella LAN, senza dialogare con il router della LAN. L'interfaccia collabora con i router della LAN per assegnare uno o più indirizzi Global Unicast per poter comunicare in internet. Anche in IPv4 sono presenti protocolli simili in modo che una macchina si attribuisce un indirizzo IPv4, questi meccanismi provengono da servizi esterni, mentre in IPv6 questi meccanismi appartengono al protocollo.

L'auto-attribuzione dell'Interface ID può essere causale oppure basata sull'indirizzo MAC dell'interfaccia, anche se ha un impatto negativo sulla privacy. Ma permette di realizzare indirizzi univoci, dato che lo sono tutti gli indirizzi MAC. L'interfaccia divide l'indirizzo MAC in due parti, ciascuna di 24 bit, viene inserita la sequenza di 16 bit ff:fe alla metà, tra le due parti. Al settimo bit della prima parte viene attribuito 1, poiché se l'indirizzo MAC è unico a livello globale quel bit vale 0. Questi 64 bit identificano l'indirizzo nella LAN. Gli indirizzi MAC studiati si riferiscono ad uno standard EUI-48, e per passare ad indirizzi MAC dda 64 bit si utilizza il meccanismo descritto nello

standard EUI-64. Se viene utilizzato questo indirizzo per comunicare in internet, viene mostrato il proprio indirizzo MAC, questo rappresenta un aspetto molto negativo in merito alla privacy. Altrimenti meccanismi di scelta casuale possono essere utilizzati, ed includono meccanismi per evitare collisioni. Un'interfaccia costruisce il proprio indirizzo Link-Local anteponendo a questo Interface ID il prefisso fe80::/64. Prima di considerare questo come indirizzo, svolge un'attività di "Duplicate Address Detection", controllando se nella stessa LAN è presente un'interfaccia di stesso indirizzo, inviando un pacchetto all'indirizzo costruito. Questo indirizzo è Link-Local quindi deve essere unico solamente all'interno della LAN. I router presenti sulla LAN inviano periodicamente dei pacchetti di "Router Advertisement" destinati ad ff02::1, tutte le interfacce presenti sulla LAN. Le LAN a loro volta possono sollecitare questi pacchetti tramite pacchetti "Router Solicitation". Nei pacchetti di Router Advertisement, fornisce un elenco di prefissi utilizzabili sulla LAN, un tempo di validità ed una specifica se può essere utilizzato come router di default, fornisce altre informazioni utili alle interfacce ed il proprio indirizzo MAC.

Un'interfaccia che riceve questi prefissi può attribuirsi indirizzi ulteriori rispetto a quelli Link-Local. Esistono protocolli appositi distribuendo prefissi salvati su server, ed attribuibili da remoto ai vari router della LAN. La spedizione di un pacchetto avviene come in IPv4, controllando se il destinatario è sulla stessa LAN, si effettua quindi spedizione diretta o spedizione al router di default. Le tabelle di instradamento dei router hanno lo stesso significato delle tabelle IPv4, utilizzate allo stesso modo. In IPv6 non c'è il protocollo ARp per individuare l'indirizzo MAC del destinatario. Il protocollo ARP, invia continuamente richieste broadcast che invia molte richieste sulla LAN. In IPv6 si utilizza un meccanismo ICMPv6, svolge le stesse funzioni di ICMPv4, ed altre funzioni, sopratutto svolge le funzioni svolte da ARA in IPv4.

Per cercare un indirizzo MAC si spedisce un pacchetto ICMPv6 ad un gruppo di multicast, non broadcast. Quando un'interfaccia si assegna un indirizzo ICMPv6, assume anche di appartenere ad un uno specifico gruppo multicast, chiamato "Solicited Node". Questo indirizzo è composto da 24 bit IID sono gli ultimi 24 bit del'indirizzo. I primi 16 bit sono ff02, i seguenti 72 bit sono pari a zero, i seguenti 16 sono 01:ff, seguiti dai 24 dell'IID. La probabilità di due macchine di avere lo stesso IID sulla stessa macchina è molto bassa, quindi questi gruppi multicast sono molto piccolo, a volte composti da una singola macchina.

Per ottenere un indirizzo MAC di un sistema, un nodo di una macchina calcola dall'indirizzo IPv6 del destinatario la sua IID, e determina il suo gruppo multicast a cui appartiene. A questo indirizzo invia un pacchetto ICMPv6 di tipo "Neighbor Solicitation". Se il destinatario è presente invia un pacchetto unicast, con il suo indirizzo MAC specificato nella parte dati del pacchetto di tipo "Neighbor Advertisement". Questo indirizzo viene specificato nella "Neighbor Cache", equivalente alla ARP Cache.

Si utilizza Solicited Node poiché un pacchetto multicast è considerato solo da un gruppo di macchine e non dall'intera LAN, richiede un gruppo multicast corrispondente a livello MAC, per evitare di dover inviare pacchetti broadcast a livello due invece che a livello tre. Visto che l'indirizzo contiene gli ultimi bit dell'indirizzo è probabile che venga processato da poche schede. Per individuare

6 Livello 4: Strato di Trasporto TCP e UDP

Da questo livello in poi si possono sviluppare applicazioni, il servizio a questo livello deve essere affidabile, poiché si suppone che la rete sia affidabile.

Offre servizi contesi con una connessione bidirezionale e contemporanea tra le due parti. I processi che usano queste primitive assumono quindi che questo livello sia affidabile.

Le primitive sono di diversi tipi, offerte ad una popolazione molto ampia di utenti-programmatori:

- `listen`: Mette la macchina in attesa di ricevere una richiesta di instaurazione di una connessione;
- `connect`: Tenta di instaurare una connessione;
- `send`: Invia dati;
- `receive`: Riceve dati;
- `disconnect`: Rilascia una connessione.

Il nome della primitiva dipende dal linguaggio di programmazione utilizzato per accedere a questa primitiva.

La contemporaneità nella rete non è effettivamente garantita, è molto improbabile che nello stesso istante due es si comunichino a vicenda, ma sono in grado di ascoltare la rete ed aspettare di ricevere o inviare messaggi. Affinché la connessione sia affidabile, ogni dato inviato viene riscontrato dall'es di destinazione.

Le primitive per l'instaurazione ed il rilascio di connessioni devono essere realizzati in modo affidabile, più facile da risolvere per l'instaurazione che per il suo abbattimento.

I pacchetti scambiati in una connessione sono sempre numerati a entrambi le parti in modo sequenziale, e si possono riscontrare i pacchetti inviati tramite acknowledgment, analogamente ai casi precedenti. La numerazione inizia da un numero arbitrario scelto dai due es. L'instaurazione della connessione si basa sui numeri iniziali della sequenza scelta degli es. Questo metodo viene chiamato "three way handshake".

Dati due calcolatori, il primo sceglie un numero di sequenza arbitrario per il numero dei pacchetti x ed invia una Connection Request al secondo con il numero scelto, a questo punto il secondo calcolatore ricevuto il pacchetto sceglie il uso numero iniziale di sequenza y. Questo invia un pacchetto di tipo "Connection Accepted", contenente x e y. A questo punto il primo calcolatore suppone che questi due valori siano validi ed invia un pacchetto di riscontro al secondo calcolatore con un pacchetto contenente y. Questo pacchetto di riscontro finale ha un valore x, ma queste sono questioni di convenzioni che verranno trattate successivamente. Spesso il primo pacchetto dell'handshake si chiama Data, poiché potrebbe già contenere dei dati dal primo al secondo calcolatore. Questo è sicuramente non vero per il primo ed il secondo pacchetto della connessione.

Numerare i pacchetti è molto utile, per permettere di riscontare esattamente quali pacchetti sono stati ricevuti.

Sarebbe semplice numerare i pacchetti da zero per entrambi gli es, ma utilizzare numeri diversi è possibile distinguere pacchetti scambiati da due stessi calcolatori, anche in tempi diversi, su connessioni diverse tra di loro.

Rilasciare una connessione è un processo complesso, se non viene effettuato correttamente si potrebbe rischiare la perdita di alcuni dati. Se la connessione viene terminata in modo unilaterale, il rilascio è rudimentale, poiché il secondo calcolatore connesso potrebbe aver inviato pacchetti, nonostante la connessione sia stata interrotta. Questi pacchetti verranno quindi persi. Si potrebbe utilizzare un rilascio simmetrico, specificando all'altro calcolatore che si ascolta la connessione per un certo periodo di tempo per continuare a ricevere dati inviati dopo il termine della connessione. Questo è un problema molto complesso, poiché non è definito in quale istante la connessione si dice completamente terminata.

Questa situazione si può rappresentare in modo semplice con il problema dei due eserciti, o due generali. Si suppone esistano due eserciti, il primo numericamente superiore al secondo, ma diviso in due parti minori del secondo. Queste due parti sono disgiunte tra di loro, e per poter battere il secondo esercito devono necessariamente attaccare in contemporanea, altrimenti non riuscirebbero a vincere. I generali di queste due parti del primo devono comunicare spedendo emissari tra di loro, ma questi devono attraversare il territorio dell'esercito nemico, e potrebbero essere catturati. Si potrebbero emissari tra le due parti per riscontrare, inviando un numero arbitrario di riscontri, l'ultimo generale ad inviare il suo emissario come riscontro, non saprà mai se il messaggio è stato ricevuto. Non esiste infatti protocollo di comunicazione per garantire a questi due generali una vittoria.

Si inserisce un timer, se entro un certo periodo non succede niente, allora viene rilasciata la connessione. Si vorrebbe avere un tempo di timeout molto ristretto, ma questo rappresenta un'euristica, non garantisce la perdiate di pacchetti.

6.1 TCP

Il protocollo TCP viene definito negli standard RFC 793, 1122 e 1323. TCP offre un servizio di trasmissione di dati affidabile bidirezionale e contemporaneo, chiamato full-duplex, oppure punto-punto, "host-to-host" o "end-to-end". Connessioni di tipo multicast o broadcast non sono supportate, può usare protocolli di livello tre diversi, sia IPv4 che IPv6, tra gli altri.

È il primo protocollo a garantire l'affidabilità della connessione, identifica i problemi dei livelli sottostanti, se un pacchetto viene scartato nel percorso lo richiede nuovamente, e se i pacchetti arrivano disordinati li riordina. Utilizza meccanismi di acknowledgment, ogni pacchetto viene seguito da riscontri. Una connessione non avviene tra due calcolatori, ma tra due processi attivi dentro una macchina. Ogni processo in una connessione è identificato dall'indirizzo IP della macchina, ma questo non è sufficiente, quindi si utilizza un numero assegnato a ciascun processo per distinguerli, chiamato numero di "port" o porta. Una volta stabilita la connessione i dati allo strato TCP vengono inviati correttamente allo strato TCP del ricevente, offerti al processo destinazione. Dal punto di vista dell'applicazione la rete non esiste, si interfaccia solamente con il livello TCP del destinatario. I port i TCP sap, sono numeri da 2 byte, non possono essere minori di 1024, riservati a servizi standard, anche se molti altri numeri sono riservati ad altri servizi con un port superiore. Un processo potrebbe chiedere al protocollo un certo numero di port specifico da utilizzare e se non è già utilizzato viene assegnato questo numero.

I pacchetti TCP si chiamano "segment", segmenti, costituiti da un intestazione e dai opzionali. C'è un limite per i dati IP di 65553 byte, un altro limite più stringente è dato dalle MTU. I byte a

disposizione dell'applicazione per ogni pacchetto sono 1460 byte, poiché 20 riservati all'intestazione del livello tre, ed altre 20 riservate all'intestazione IPv4. I pacchetti non sono numerati, ma vengono numerati i byte dei pacchetti. Con un numero di sequenza di 32 bit, e consente di avere reset dei numeri di sequenza infrequenti, rappresenta una sequenza circolare. Questi vengono riscontrati byte per byte utilizzando acknowledgment.

I segment vengono utilizzati per instaurare connessioni, spedire dati, spedire acknowledgment o chiudere connessioni.

Ogni riga è composta da 4 byte. Nella prim sono presenti il numero di porta del sorgente e del destinatario, la seconda contiene il numero di sequenza del byte spedito, ed il seguente il numero di sequenza di acknowledgment. Se si usasse un intestazione per spedire ogni byte sarebbe troppo inefficiente, quindi questo numero di sequenza di byte si riferisce al primo byte del campo data. Questi byte vengono numerati a partire da questo numero di sequenza specificato. L'acknowledgment specifica l'ultimo byte che è stato riscontrato. Il campo hlen specifica la lunghezza dell'intestazione, res è un campo riservato, il campo code contiene la funzione del pacchetto, il campo windows regola il controllo di flusso, lungo 2 byte. Nella riga sequenza è presente una checksum, importante per capire se il pacchetto è integro, segue un "urgent pointer" per specificare un dato importante da leggere appena viene ricevuto il pacchetto. Segue una riga con le opzioni ed eventuale padding.

Quando si riscontra un numero di sequenza in un acknowledgment si specifica il numero del prossimo byte atteso, questo campo quindi si riferisce ai dati che viaggiano in direzione opposta. Il numero di sequenza inoltre stabilisce la posizione del pacchetto nel flusso di dati generando un certo numero di pacchetti.

Il nome del pacchetto per la richiesta dell'instaurazione della connessione si chiama "syn", mentre "ack" per il pacchetto di riscontro. Il significato applicativo dei byte trasportati dal protocollo TCP non sono noti al protocollo stesso, o a qualunque protocollo di livello quattro, questi verano interpretati da protocolli al livello superiore.

Il campo hlen contiene il numero di parole di 32 bit dell'intestazione. Il campo code determina il tipo di messaggi contenuto nel segmento:

- URG: Il campo urgent pointer è valido;
- ACK: Il campo ack è valido, ovvero il riscontro contenuto deve essere considerato;
- PSH: Specifica che il pacchetto deve essere inviato velocemente;
- RST: Specifica di voler resettare la connessione;
- SYN: :Sincronizza i numeri di sequenza;
- FIN: Il mittente ha raggiunto la fine del byte stream e rilascia la connessione.

La checksum intessa l'intero segment e gli indirizzi IP ed il campo protocol dei campi al livello tre. Questa checksum provoca un'invasione della gerarchie sui campi del terzo livello. Le opzioni specificano la negoziazione sulla massima ampiezza del campo dati nell'instaurazione della connessione. Gli host sono obbligati ad accettare almeno 536 byte.

7 Livelli Applicativi: DNS, HTML, HTTP

7.1 DNS

Il DNS o Domain Name System è un applicazione del livello ISO-OSI sopra al livello quattro. Dal livello quattro in poi le applicazioni

Poiché è scomodo utilizzare un numero per identificare un indirizzo, si vuole assegnare ad ogni interfaccia un nome, quest'applicazione si occupa della mappatura tra questi nomi assegnati a l'indirizzo IP. Agli albori dell'internet questo "Spazio dei Nomi" era "flat", quindi poteva essere assegnato qualsiasi nome arbitrario, e la mappatura veniva tracciata da una persona a mano su un file testuale. Al crescere della dimensione della rete tuttavia potevano sorgere facilmente conflitti. La possibilità di aggiungere e modificar Ei nomi necessitava di un accesso a questo file di testo centrale.

Per migliorare questo servizio si ha pensato di decentralizzare l'assegnazione dei nomi e la responsabilità del mapping tra nome ed indirizzo. Inoltre la più importante aggiunta è la possibilità di accedere al mapping con tecniche client-server.

Si utilizza un database distribuito per contenere queste corrispondenze, la robustezza e la resistenza di questo algoritmo sono favorite da meccanismi di caching e replicazione. Il "namespace" viene partizionato per garantire un controllo efficiente non centralizzato delle assegnazioni. I nomi non sono più sequenze di caratteri, ma sequenze di caratteri separati da punti. I nomi che hanno un suffisso comune possono essere rappresentati attraverso un albero. Il livello più alto della gerarchia, l'autorità "top level" delega la gestione dei nomi ad autorità assegnate a diverse partizioni dello spazio dei nomi. L'autorità top level non si occupa dell delle partizioni interne, e si suppone tutti questi domini siano figli di un dominio radice, cui corrisponde una stringa vuota. Questa partizione può somigliare all'indirizzamento IPv4, ma questo meccanismo di delega applicabile a più livelli può espandere lo spazio, mentre in IPv4 lo spazio è finito e definito a priori.

Un dominio è un sottoalbero del namespace, il cui nome è il nome della radice del sottoalbero. Le foglie di questo albero sono in corrispondenza di indirizzi IPv4 e IPv6, rappresentando interfacce. Inoltre è possibile che i nodi intermedi rappresentino ulteriori interfacce. Anche un singolo host è un dominio. In questo momento esistono 1500 domini top level, all'inizio questa distribuzione era molto ristretta. I domini nazionali sono stati standardizzati con sigle da ISO 3166, e recentemente questa restrizione è stata resa più lasca, ed esistono vari domini di recente definizione, come il dominio del cern. I domini su cui nasce internet sono principalmente domini nord americani, come com, edu, gov, etc.

L'organizzazione dei nomi di internet viene chiamata "Domain Name System", i sistemi che realizzano il mapping tra indirizzi e nomi sono chiamati Name Server (NS). Alcuni hanno al delega per una porzione del namespace, e questi sono in grado di dialogare tra di loro. Un NS ha informazioni complete su una parte del namespace, detta zona, sulla quale ha autorità. Una zona è un sottoalbero del namespace, eventualmente privata di alcuni sottoalberi.

Questi sottoalberi su cui non ha autorità sono altri domini delegati ad altri NS dal top level. Un NS può essere autorità di varie zone, inoltre per la stessa zona possono essere associati più NS autorità, per motivi di resistenza ai guasti. Poiché questo sistema è nevralgico per la gestione di internet, ed è necessario il suo funzionamento per l'accesso all'internet.

I NS possono essere primari o secondari, i server primari contengono le informazioni aggiornate sul mapping della zona, mentre gli NS secondari richiedono periodicamente il mapping della zona al server primario. Si lavora su modiche o assegnazioni dei nomi solamente sul NS primario. Un NS può essere contemporaneamente primario per certe zone e secondario per altre. Si hanno vari gradi di libertà sul numero dei NS, sulla possibilità di essere primari o secondari, oppure sulla loro posizione, possono essere anche lontani dalle macchine delle quali conservano i nomi. I client che usano i NS si chiamano resolver e sono all'interno degli host. Hanno la capacità di interrogare i NS ed interpretare le loro risposte. Sopratutto sono in grado di inviare le informazioni ricavate ai programmi che li utilizzano. Non necessariamente è un processo autonomo, i resolver sono all'interno di comuni librerie contenuti in alcuni browser.

Se il resolver ha già le informazioni necessarie le fornisce, alternativamente le chiede direttamente alla radice autorità dello spazio dei nomi, il quale fornisce al resolver un NS più specifico, scendendo l'albero fino alla raggiunta dell'indirizzo specifico. Questi NS assegnati alla radice sono costantemente sotto pressione, poiché l'intera rete si basa interamente su di loro per effettuare il processo di risoluzione.

Per la risoluzione sono possibili vari atteggiamenti, può essere ricorsiva oppure iterativa. Nella risoluzione ricorsiva il client chiede al server una risorsa, e pretende come risultato l'indirizzo, se il server non la conosce, è suo compito contattare altri server. Mentre nella risoluzione iterativa, è il client a dover contattare ulteriori server, specificati dal server iniziale. Nella risoluzione iterativa i serve non si devono tenere informazioni in memoria su cosa sta accadendo nella rete. Il resolver effettua una query ricorsiva al NS e questo effettua delle query iterative a vari NS, appunto per rendere più efficiente.

Le informazioni memorizzate tra le vari query vengono salvate su vari cache dentro i NS, che svolgono il ruolo di autorità per le varie zone, quindi in alcuni casi non è necessario scendere l'intera gerarchia della zona. Alcuni server possono memorizzare informazioni negative, la cache è presente anche al livello di applicazione, come i browser. L'informazione sulla cache scade dopo un certo intervallo di tempo. Il tempo di vita delle informazioni in cache si chiama TTL, "Time To Live" e viene scelto con opportune scelte per un compromesso tra consistenza ed efficienza, non è in relazione con il TTL dei pacchetti sull'internet.

Alcuni NS sono autorità per nessuna zona, e sono solamente a disposizione dei client e possono accettare query ricorsive ed effettuare query iterative. Non sono solo a disposizione dei client. Normalmente ogni IPS ne mette a disposizione uno per i propri clienti, sono chiamati a volte Local NS, o Caching NS, oppure Recursive NS.

Le informazioni DNS sono memorizzate su "Resource Record", ed ogni dominio è associato ad uno o più Resource Record, questi contengono gli indirizzi IP, ma anche altre informazioni che permettono di associare ai nomi altri servizi. Uno di questi servizi sono i servizi di posta elettronica.

Questi record contengono campi per il nome del dominio, il TTL, la classe, in internet è sempre `IN`, il tipo di record ed il valore, che dipende dal valore di tipo. I tipi principali sono

- `SOA`: "Start of Authority", contiene informazioni amministrative della zona;
- `A`: Contiene nel valore l'indirizzo IPv4 dell'host;
- `MX`: Specifica il nome dell'host che accetta le mail indirizzate al dominio del record;

- `NS`: Il NS per una zona;
- `AAAA`: Contiene nel valore l'indirizzo IPv6 dell'host;
- `CNAME`: "Canonical Name", contiene il nome corrisponde ad un altro nome o alias.

Si possono assegnare dei nomi ai computer che si comportano come NS. All'interno sono presenti dei record che si pensano nella posizione sbagliate, questi indirizzano al NS, per permettere di effettuare richieste ricorsive o iterative, questi si chiamano "Glue Record".

I messaggi DNS si realizzano in due tipi di formato, di richiesta e di risposta. Tra i campi dell'header:

- `QR`: indica s eil messaggio è una domanda o una risposta;
- `RD`: "Recursion Required", indica che la query è di tipo ricorsivo.

Tra i campi della question section:

- `NAME`: Nome della richiesta;
- `TYPE`: Tipo della richiesta.

Questi messaggi devono viaggiare su un protocollo di livello quattro affidabile, quindi sembra bisogna essere assegnato al TCP, ma in realtà viene assegnato all'UDP. È la logica applicativa che si prende carico del meccanismo dei riconto, in caso le richieste non ricevono risposta, allora inviano nuovamente la richiesta dopo un certo intervallo di tempo. I client si rivolgono alla porta UDP 53 "Well Known", la richiesta e la risposta viaggiano entrambe su singoli pacchetti UDP. Per dialoghi che richiedono risposte di grande dimensioni si può usare TCP, nel caso in cui server secondario si rivolge al primario per richiedere le informazioni aggiornate.

Sotto il DNS non ci sono solo messaggi UDP, ma ci sono protocolli di sicurezza per evitare vengano letti da terze parti, la risposta viene quindi considerata autentica. Questi meccanismi non sono trattati in questo corso.

7.2 HTML e HTTP

Il linguaggio HTML è il linguaggio che specifica le pagine web, si interessa il protocollo che porta queste informazioni attraverso la rete, ovvero HTTP. HTML, "HyperText Markup Language", è un testo che può contenere riferimenti ad altri ipertesti. Un marcatore è un codice che segnala l'inizio o la fine di una primitiva di formattazione del testo.

Le caratteristiche dell'ipertesto sono la concisione

Si può arrivare a livelli di completezza arbitrari affidando a subordinate distribuendo l'informazione nelle pagine. I dati vengono condivisi evitando le ridondanze

L'ipertesto costituito dall'insieme di tutte le pagine presenti su internet viene denominato World Wide Web, rete estesa al mondo intero. Bisogna distinguere tra internet e WWW. La parola internet è ulteriormente ambigua poiché il nome del protocol Internet Protocol e l'insieme dei protocolli Internet Protocol Suite. Inoltre si riferisce alla struttura fisica, ma spesso si riferisce a tutti i servizi

disponibili in questo ambiente distribuito. Mentre WWW si riferisce alla rete logica costituita dalle pagine web e dai loro hyperlink, queste costituiscono solo uno dei servizi offerti dall'internet.

HTML è un linguaggio di "markup", è un linguaggio che codifica delle informazioni tramite marcatori. Un linguaggio di marcatura ha senso quando si vuole rappresentare un'informazione che può essere strutturata in termini sequenziali.

Si possono identificare linguaggi di marcatura fisica o logica. I linguaggi della struttura fisica descrive dettagliatamente come apparirà il testo formattato. Queste ha una filosofia WYSIWYG, "What You See Is What You Get". Si vuole poter descrivere testi per schermi di dimensione e

Linguaggi di marcatura logica o semantica descrive il significato strutturale degli oggetti da formattare con una filosofia WYGIWYM, "What You Get Is What You Mean".

HTML usa una marcatura logica e permette di realizzare pagine web senza doversi preoccupare delle caratteristiche del dispositivo sul quale quel testo sarà offerto in consultazione. Ogni marcatore in HTML viene identificato all'interno di parentesi acute `< ... >`. I marcatori di fine formattazione hanno uno slash prima della parentesi angolata aperta `</ ... >`. Alcuni marcatori ammettono valori dopo un uguale. I commenti hanno una forma `<!-- ... --->`. Le parentesi angolate non sono quindi disponibili come caratteri si utilizzano caratteri speciali per indicare questi caratteri che non possono essere scritti direttamente. Iniziano con `&` o `;`:

- `&lt`: <;
- `&gt`: >;
- `&amp`: &;
- `&_grave`: lettera _ con accento grave.

Esistono marcatori meta che non necessariamente servono per la formattazione, ma possono inserire marcatori per identificare certe parole all'interno della pagina.

Un file HTML viene strutturato all'interno del marcatore `<html> ... </html>` e si può dividere in due sezioni `<head> ... </head>` che contiene informazioni relative all'intera pagina, e `<body> ... </body>` che contiene descrizioni relative agli oggetti da visualizzare nella pagina. Nell'intestazione si può trovare il titolo con `<title> ... </title>`, metadati `<meta ... = ... >`

Si può formattare il testo con i seguenti marcatori:

- `<center> ... </center>`: Centratura;
- `<h1> ... </h1>`: Titolo;
- `<h2> ... </h2>`: Sottotitolo;
- `<b> ... </b>`: **Bold**;
- `<i> ... </i>`: *Italic*;
- `<u> ... </u>`: <u>Sottolineato</u>;
- `<blink> ... </blink>`: Lampeggiante.

Si definiscono ancore con il marcatore `<a> ... </a>`, in base all'attributo si possono creare hyperlink con `href`, un ancora di un punto di arrivo con `name`. Si possono inserire hyperlink ad una posizione nella stessa pagina oppure ad un'ulteriore pagina.

Si possono inserire immagini con `<img src="URL">`, a capi di un paragrafo `<p>` o break `<br>`, linee orizzontali `<hr>` di cui si può modificare lo spessore con `size=...`. Si può colore del font con `<font color=...></font>`. Si possono realizzare liste non ordinate:

```
<ul>
    <li> ...
    <li> ...
</ul>
```

Oppure ordinate:

```
<ol>
    <li> ...
    <li> ...
</ol>
```

Si possono definire tabelle con:

```
<table>
<table border=...>
<tr> ... </tr> <!-- inizio-fine linea-->
<td> ... </td> <!-- inizio-fine cella-->
<th> ... </th> <!-- titolo cella-->
</table>
```

Si possono definire ulteriori marcatori ed estensioni, ma se il browser non conosce il marcatore non produce errore, ma semplicemente lo ignora. HTML è un linguaggio estremamente basilare che si è evoluto nel tempo, attualmente siamo alla versione HTML 5. Il codice HTML viene scandito riga per riga, partendo dall'alto ed eventualmente viene formattata la pagina. HTML viene spesso accompagnato da altre tecnologie come javascript o CSS, "Cascading Style Sheets", fogli di stile, elementi di formattazione che possono essere utilizzati epr rendere la pagina al meglio.

7.3 URL ed il Protocollo HTTP

Prima di descrivere il protocollo HTTP, bisogna descrivere l'URL, "Uniform Resource Locator", o l'URI, "Uniform Resource Identifier". Poiché in internet c'è la differenza tra lar risorsa e la sua posizione.

Un URL rappresenta una rappresentazione testuale e compatta di una risorsa disponibile, in modo sintetico realizzata da uno schema, ed una parte dipendente dallo schema, separati dal separatore `:`. Lo schema descrive il tipo di schema, e la parte seguente la individua secondo le caratteristiche di un dato schema. Tipicamente lo schema è `http` o `https`, questi sono sia risorse che protocolli per acquisire risorse. Schemi come `file` sono risorse presenti sul proprio calcolatore.

Alcuni schemi utilizzano la stessa sintassi per la parte dipendente, questa sintassi si chiama "Common Internet Scheme Syntax", CISS. SI utilizza un doppio slash iniziale seguito da un campo con l'userid e la password seguita da una chiocciola, ma questo è molto raramente utilizzato negli URL moderni. Il secondo campo dopo la chiocciola corrisponde all'indirizzo dell'host, il suo numero IP, o ad una stringa identificativa di questo indirizzo sull'internet. Il campo successivo specificato dopo `:` indica la porta dove è disponibile quella risorsa, si suppone la risorsa sia un livello quattro che dispone di porte, questa può essere omessa, ed in caso venga omessa si utilizza una porta di quelle note, associate a protocolli HTTP, come la porta 80. L'ultimo campo individua un file all'interno del filesystem, del server con `/` seguito dall'indirizzo del file. Alcuni sistemi presentano variazioni nella sintassi di questo campo, come HTTP e HTTPS. Questo identifica una risorsa, tipicamente attraverso l'indirizzo dell'host, ma l'indirizzo IP ha anche il ruolo di locazione, quindi è possibile localizzare la risorsa dato l'IP.

Quando un browser deve accedere ad una risorsa individuata da un URL etra i campi di indirizzi e la porta, se non è specificata utilizza la porta 80, per il protocollo HTTP. Questo lavoro lo effettua qualsiasi applicazione che tenta di accedere a duna risorsa. Il campo userid e password vengono utilizzati per gestire un colloquio con il server, senza coinvolgere l'utente. In coda al path può esserci una stringa preceduta dal carattere `#` per individuare la posizione all'interno della pagina HTML, utilizzata dal client e non inviata al server. Mentre una stringa preceduta da `?` viene inviata al server, per specificare parametri o altre opzioni.

Il protocollo HTTP, "HyperText Transfer Protocol", è stato progettato come un protocollo di livello applicativo per realizzare sistemi distribuiti, basati su ipertesti, anche se da molto non è più esclusivo per HTML.

HTTP venne inventato da Tim Bernes-Lee, un ricercatore del CERN nel 1989, e da HTTP/3, nel progetto QUIC, c'è una variazione molto grande rispetto alle reti precedenti, con lo standard RFC 9114.

Una sessione di HTTP/1.0 è composta da quattro fasi di apertura, richiesta, risposta e chiusura. Nel codice del server è stato specificato ad un processo di ascoltare sulla porta 80, oppure una porta specificata per ascoltare eventuali richieste di connessione. Un cliente quindi puà effettuare una connessione TCP con questa porta TCP del server. Una volta instaurata una connessione TCP viene inviata una richiesta al server, un pacchetto HTTP di richiesta al server con la specifica della risorsa alla quale si vuole accedere. Il server poi restituisce la risorsa, tipicamente un file, contenente una descrizione HTML di una pagina che si vuole visualizzare. Dopo aver terminato la trasmissione, il server chiude la connessione TCP. Avviene quindi un singolo scambio client-server in ogni sessione, questa sezione è completamente stateless.

Nella richiesta viene specificato uno di possibili metodi tra questi seguenti:

- `GET`: Richiede la risorsa indicata;
- `POST`: Invia dati alla risorsa indicata;
- `HEAD`: Chiede informazioni sulla risorsa indicata;
- `PUT`: Copia dei dati inviati sulla risorsa con una sostituzione di file;
- `DELETE`: Richiede la cancellazione della risorsa:

- `OPTIONS`: RIchiede di conoscere le opzioni disponibili per il trasferimento della risorsa specificata.

Un pacchetto HTTP è strutturato da un'intestazione composta da un metodo, risorsa e versione di protocollo seguita da informazioni aggiuntive. Una riga vuota quindi divide l'intestazione dal corpo del pacchetto dove sono presenti i dati. Questi sono pacchetti di richiesta, pacchetti di risposta hanno nell'intestazione la versione, ed in linguaggio naturale il codice e la spiegazione per il codice. Questi codici di stato possono essere positivi, di tipo `2xx` o `3xx` oppure negativi `4xx` o `5xx`. Non è necessario conoscere questi codici di risposta poiché nel pacchetto è presente un campo testuale contenente la spiegazione del codice. I codici di risposta sono tipicamente sempre leggibili, per molti protocolli che definiti da standard RFC.

Nelle versioni successive i pacchetti vengono inviati allo stesso modo, ciò che ca,bia sono le informazioni addizionali. Possono essere presenti alcuni campi di richiesta:

- `accept-charset`: Insieme di caratteri accettabili;
- `cookie`: Un cookie inviato precedentemente dal server;
- `content-length`: Lunghezza in byte del corpo;
- `host`: Il domain name del server e il numero di port TCP, utile per servire vari domini sullo stesso server.

Su uno stesso server possono essere presenti servizi diversi, si utilizza il campo `host` per specificare a quale di questi si vole accedere. Ulteriori campi di richiesta

-

Alcuni campi di risposta sono `ETag`, questo è un identificatore assegnato da un server ad una specifica risorsa, può essere generato da una funzione hash. Può offrire informazioni aggiuntive rispetto ad un semplice valore temporale. Un altro di questi capi di risposta è `set-cookie` questo contiene un cookie HTTP. Un cookie è un dato spedito dal server e memorizzato nel client, viene re-inviato dal client al server ogni volta che accede a certe risorse, definite dallo scope del cookie. Viene usato per gestire uno stato nelle sessioni HTTP. Il server può quindi riconoscere attività precedenti dello stesso client. L'applicazione più evidente è quella dell'autenticazione degli accessi ad un server, questo cookie permettere al server di identificare lo stesso client, quindi attribuisce i diritti per poter eseguire le operazioni successive. Può anche permettere di tracciare le pagine visitate, da uno stesso utente. Un cookie può essere utilizzato anche da altri servizi nonostante non sia stato rilasciato per un certo sito. Nell'EU all'accesso di un sito, questo è obbligato di specificare il motivo dei cookie che vengono utilizzati, il loro scopo o finalità, ed il modo in cui vengono salvati e mantenuti nel tempo. Inoltre è possibile accedere comunque al sito senza accettare all'utilizzo dei cookie.

Il protocollo HTTP 1.1 permette di effettuare diverse sessioni sulla stessa connessione TCP, ma questo dipende comunque dal server.

Questa possibilità è particolarmente importante se insieme a HTTP e TCP si utilizza un protocollo per rendere sicure le connessioni. Attualmente si utilizza il protocollo TLS; "Transport Layer

Security", dopo l00instaurazione della connessione TPC i due host si scambiano informazioni per mantenere la sicurezza delle connessioni. Quindi all'instaurazione della connessione TCP si effettua uno scambio per determinare il protocollo di sicurezza utilizzato. Questo permette di offuscare i pacchetti trasmessi tra queste due macchine mantenendo la connessione persistente. Se fosse possibile effettuare una singola connessione TCP per ogni richiesta verrebbero scambiati 7 pacchetti di apertura-chiusura per una singola sessione.

HTTP quindi rappresenta una sequenza di transizioni richiesta-risposta tra un client ed un server, tipicamente originate da un singolo utente, per accedere a risorse tra loro correlate.

Se si usano HTTP e TLS separatamente è opzionale l'utilizzo di TLS, mentre il protocollo HTTPS presenta l'utilizzo obbligatorio del protocollo di sicurezza TLS. Cambia anche la porta di ascolto nota e diventa la porta 443.

HTTP/3 invece utilizza un ulteriore protocollo di trasporto chiamato QUIC e non TCP. QUIC utilizza pacchetti UPD, basato su implementazioni nell'user-space e quindi non sono legate all'aggiornamento del sistema operativo. Si intreccia quindi l'evoluzione dei protocolli di trasporto e l'evoluzione di HTTP. Questo rappresenta un fenomeno di evoluzione molto recente.

In QUIC il three-way-handshake è mescolato con l'handshake del protocollo TLS 1.3, con un minore overhead della connessione. Per utilizzare questo protocollo, anche il server deve poterlo utilizzare, se il server è di Google allora non ci sono problemi, poiché questo protocollo è stato introdotto da Google. Se il server non appartiene a Google, allora è possibile che utilizzi il protocollo QUIC, circa il 20% dei siti Web utilizza il protocollo HTTP/3, mentre tutti i browser lo utilizzano, perché vogliono offrire sempre migliori prestazioni.

8 Posta Elettronica

La descrizione di questo servizio web consiste nel definire un criterio per la sua progettazione, utilizzato per realizzare nuovi servizi o modificarne di esistenti. La progettazione è divisa in quattro fasi, un'analisi dei requisiti, definendo le caratteristiche dei servizi. La specifica delle primitive del servizio, che rappresenta la sua interfaccia. Definizione dell'architettura e delle operazioni, e la definizione dei protocolli per ogni tipo di comunicazione, bisogna definire la struttura delle PDU, degli stati e delle procedure del protocollo.

I servizi di posta elettronica possono essere forniti da ISP, aziende come Google, che offrono altri servizi web.

Si effettua in seguito un'analisi dei requisiti di progettazione per un servizio di posta elettronica. Un servizio di posta deve poter gestire messaggi in partenza ed in arrivo, spedire e ricevere messaggi ad uno o più destinatari. Richiede ri preservare la riservatezza dei dati, e si vuole avere una certezza sulla consegna, è anche possibile avere una consegna differita nel tempo, per garantire la consegna. Quest'approccio duale rispetto ad altri servizi web. Si vuole che la configurazione sia poco onerosa da parte degli utenti finali con un'interfaccia utente elementare ed intuitiva.

Definiti i requisiti si definiscono delle primitive di servizio offerte agli utenti, per i messaggi in partenza:

- Composizione di un messaggio;
- Memorizzazione di un messaggio;
- Cancellazione di un messaggio;
- Caricamento di un messaggio.

Per i messaggi ricevuti:

- Lettura dei messaggi;
- Memorizzazione di un messaggio;
- Cancellazione del messaggio;
- Stampa di un messaggio.

Queste sono le primitive di base per poter almeno gestire la posta elettronica. Per progettare un servizio bisogna effettuare un'analisi dei requisiti e definire su di questi una serie di primitive di servizio. Dopo queste due fasi bisogna definire l'architettura del servizio.

Si considera l'architettura odierna della posta elettronica, descrivendo ulteriori architetture non realizzabili valutando variazioni. Si considera un possibile architettura realizzata da una connessione diretta tra il mittente ed il destinatario. Sono quindi presenti due processi su questi due host, attivi, che devono poter ascoltare e inviare dati su una connessione, potrebbero utilizzare TCP. Per definire la porta di ascolto si potrebbe definire uno standard globale, su una porta ben nota alla definizione dell'architettura. Questo approccio è molto semplice, il messaggio viene recapitato in tempo reale e si ha anche la certezza della consegna, basata sull'affidabilità di TCP. Ma il mittente non può spedire

il messaggio se il processo sul destinatario non è attivo, quindi non rappresenta un'implementazione realizzabile. Inoltre il mittente si deve ricordare il nome della macchina del destinatario, può essere il nome del DNS o dell'indirizzo IP, ma questo deve essere noto per poter instaurare la connessione. Questo stesso approccio non è ragionevole per molti servizi basati sulle reti. Inoltre non viene definito come il destinatario viene autenticato, poiché viene riconosciuto solamente dalla macchina.

Si considera quindi una variante dove un server intermedio disaccoppia le due macchine host, questo server rappresenta il dominio del destinatario, ovvero un nodo dell'albero del namespace. Si ipotizza di legare i destinatari a domini di internet, sotto qualche forma. Questo server di ricezione deve essere ben noto. I processi di invio e ricezione sono disaccoppiati, ed il destinatario può essere autenticato dal server, dove sono presenti servizi di autenticazione. Il mittente non deve ricordare la specifica macchina del destinatario, ma solamente il dominio. Ma questo server può essere impegnato, guasto o sovraccarico in un certo istante e questo non garantisce di ricevere il messaggio dal server. Se dopo aver inviato il messaggio il processo del mittente può essere chiuso e quindi il messaggio non è in grado di arrivare al server. Poiché non si ha autorità sulla gestione del server.

L'architettura moderna contiene un serve mittente per memorizzare il messaggio, con funzione di inoltro, e molti server duplicati, definendone uno primario ed altri ausiliari. In questo modo se il primo server non è raggiungibile si può tentare di raggiungere uno dei server primari, per inoltrare il messaggio al destinatario.

Questa rappresenta un'architettura diversa da quella che is potrebbe aspettare. Bisogna quindi specificare come si definiscono indirizzi di posta elettronica, in rifermento a questa architettura. Una prima parte dell'indirizzo identifica un'entità che riceve il messaggio, ed una seconda parte per definire il dominio dell utente, dopo un `@`, per ricavare il server che deve ricevere il messaggio.

La prima applicazione coinvolta nel servizio si chiama MUA, "Mail User Agent" e viene mandato in esecuzione quando si vuole accedere al servizio di posta elettronica. L'utente può chiudere questa applicazione quando lo ritiene opportuno e viene chiamata anche mailer. L'applicazione MTA, "Mail Transmission Agent", al contrario della MUA, il MTA è accessibile in modo stabile nle tempo, poiché si trova su di un server di inoltro per permettere al messaggio di essere trasmesso da una sorgente ad una destinazione.

Esempi di MUA sono applicazioni Desktop come Mozilla Thunderbird o Microsoft Outlook, altri servizi si basano sul web che trasformano il browser in un MUA, come Gmail, a cui bisogna però autenticarsi sul browser. Sono inoltre disponibili molte applicazioni mobile Android ed iOS.

MUA e MTA sono applicazioni, che vengono eseguite su delle macchine, quelle che ospitano gli MTA sono dei server che possono avere diversi ruoli nell'architettura appena definita. Un primo tipo di server si chiama OMS, "Outgoing Main Server", alla quale MTA si riferisce direttamente il processo MUA del mittente. Un server di tipo "Mail eXchanger", per ogni dominio DNS infatti esiste una lista di host, in ordine di priorità che ospitano gli MTA incaricati di ricevere posta per quel dominio. Un ulteriore server chiamato "Incoming Mail Server" è il serve che contiene la MTA che comunica direttamente con la MUA del destinatario, generalmente coincide con il Mail eXchanger primario del dominio.

Si vuole che l'utente possa configurare facilmente il servizio della posta elettronica. In quest'architettura l'utente deve ricordarsi semplicemente due server, il suo MUA si riferisce infatti ad un Outgoing Mail Server ed un Incoming Mail Server. Questi server invece di specificarli con un

indirizzo IP si possono specificare con un nome, in modo che se dovesse cambiare l'indirizzo IP della macchina non sarebbe necessario modificare il servizio.

Le MTA ospitate sull'Outgoing Mail Server sono suddivise in due processi chiamate MSA, "Message Submission Agent", quello a cui si rivolge la MUA per spedire la posta, ed il secondo è il vero e proprio MTA che cura la spedizione al MTA del dominio del destinatario. Lo stesso avviene per un IMS, dove esiste sempre un processo MTA, ed un ulteriore processo MDA, "Mail Delivery Agent", responsabile del recapito all'utente.

Definita l'architettura si definiscono una serie di azioni, a più livelli di dettaglio, scomposte in varie operazioni. Tutte le primitive di gestione dei messaggi in partenza e dei messaggi ricevuti sono delegati al MUA. Quando l'utente richiede il salvataggio della MUA, quando l'utente richiede il salvataggio del messaggio, il MUA accede al file system locale, per salvare la nuova mail, in caso non esista, accodandola al file delle mail salvate, con il rispettivo testo.

Quando si spedisce un messaggio il MUA invia il messaggio al suo OMS, che cerca il Mail eXchanger del dominio del destinatario utilizzando il DNS, se il primario non è disponibile si rivolge al secondario, richiedendo al proprio serve primario la lista dei MX del dominio destinazione, in ordine di priorità, tentando di trasmettere il messaggio ad intervalli regolari. Se fallisce per tre giorni consecutivi viene notificato l'utente del fallimento. Se viene inviato ad un MX secondario, questo tenta ad intervalli regolari di inviare i messaggi salvati al MX primario.

Sono presenti anche servizi ausiliari, ovvero il DNS, senza il quale la posta elettronica non funzionerebbe, inoltre è necessario un File System distribuito, il server deve poter accedere ai file delle mail tramite un sistema per il File System distribuito, poiché potrebbero non trovarsi sullo stesso server.

Esiste un comando chiamato `dig` che permette, sulle macchine Linux, di comportarsi come resolver, nel mondo Windows è presente un comando analogo chiamato `nslookup`, che fornisce record MX. In questo modo è possibile ottenere le informazioni relative ai server ed i loro indirizzi IPv4. Utilizzando questo comando bisogna specificare ulteriori parametri per leggere i record MX, e specificare l'indirizzo. Questo va effettuato anche con il comando `dig` specificando dopo la flag `-f` il tipo di record che si vuole leggere e l'indirizzo corrispondente.

Dato un MUA configurato per spedire mail bisogna conoscere l'indirizzo IP dell'OMS, questo si ottiene specificando un nome ed invocando un DNS, questo effettua la risoluzione dell'indirizzo, e rappresenti il primo punto di contatto tra la posta elettronica ed il DNS, mentre un punto di contatto diverso rappresenta l'OMS che notifica al DNS di voler conoscere l'indirizzo MX del destinatario. Questa seconda operazione infatti non rappresenta una risoluzione canonica. Il messaggio viene inviato dall'OMS al MX relativo all'IMS del destinatario. Il MX e l'IMS comunicano tra di loro tramite una memoria di massa condivisa sulla quale può scrivere il MX e leggere l'IMS.

Si possono distinguere due tipi di flussi di informazione, tra il primo MUA e l'MTA dell'OMS, che si trasmettono messaggi fino a quando non arrivano al MX primario del destinatario. Questo primo flusso di informazione è realizzato da un client che intende trasferire uno o più messaggi, tramite un servizio che il server potrebbe accettare o rifiutare di concedere il servizio. Non occorre autenticare gli interlocutori poiché o entrambi hanno i privilegi di amministratore, oppure uno di essi non è il mittente, e non si considera cruciale assicurarsi che il mittente corrisponda a quanto dichiarato. Poiché l'IMS ed il MX sono due macchine associate a compagnie diverse e non sarebbe realizzabile permettere a tutti i possibili OMS le autenticazioni del server MX, analogamente per

il riferimento ad utenti che inviano messaggi da spedire ai proprio OMS. Si suppone non ci sia una verifica di autenticazione in questo passaggio, e questo è la causa dello spam che invade la posta elettronica.

Mentre l'ultimo passaggio dall'IMS dal destinatario probabilmente è realizzato da un'interazione di richieste e risposte, e sembra essere intrinsecamente più complicato. Utilizza quindi un protocollo più articolato. Il client si informa sul numero e la dimensione dei messaggi contenuti nel server, e destinati allo specifico utente. In questo passaggio l'autenticazione è indispensabile, e si possono trasferire o non trasferire messaggi. Chi utilizza questi messaggi si deve autenticare poiché consente di leggere solo i messaggi espressamente indirizzati all'utente con le opportune credenziali.