Concetti di Reti di calcolatori

Livello Rete

Il pacchetto IP (Internet Protocol):

Il pacchetto IP definito datagram è costituito da un'intestazione e da un'area dati. L'intestazione ha una lunghezza fissa di 20 bytes e una parte variabile. Il campo IHL contiene la dimensione in bytes della lunghezza totale dell'header. La massima dimensione del datagram sono 64Kbyte.

Gli altri campi sono:
Time to live: esprime il tempo di vita del datagram, rappresentato da un numero decrementato di una unità ogni volta che attraversa un router. Durante la sosta nei routers ogni secondo di attesa viene decrementato di un'altra unità.
Identification: numero che identifica il datagram.
Fragment offset: offset del primo byte di dati del datagram rispetto a quello originale.
Source address: indirizzo IP di partenza su 32 bit.
Destination address: indirizzo IP di destinazione su 32 bit.
Options: opzioni, *record route, source route, sicurezza, timestamp, id. del datagram*.
Version: versione del protocollo IP utilizzato.
Protocol: protocollo di livello superiore (trasporto) che ha generato il datagram, normalmente TCP.
MF: more fragment, flag che se posto a 0 identifica che si tratta di un unico pacchetto o dell'ultimo pacchetto di una serie.
DF: don't fragment, flag che identifica l'impossibilità di frammentare il pacchetto in datagram di dimensioni minori. Se un router non è in grado di accettare questa dimensione del pacchetto, il datagram non viene spedito.
Header checksum: controllo di errore sull'header.
Total lenght: lunghezza totale del datagram compresi dati e intestazione.
Type of service: identifica il tipo di servizio espresso in velocità ed affidabilità richieste per il datagram, non viene rispettato!

Formato degli indirizzi internet (indirizzi IP):

Il numero di rete è assegnato dal gestore di Internet, ICANN, mentre il numero dell'host è assegnato dal gestore della rete locale. Gli indirizzi che iniziano con 1111 sono riservati per usi futuri. Ci sono 5 classi di indirizzi IP:

CLASSE A

0	7 bit – NETID	24 bit - HOSTID

CLASSE B

1	0	14 bit – NETID	16 bit - HOSTID

CLASSE C

1	1	0	21 bit - NETID	8 bit - HOSTID

CLASSE D

1	1	1	0	28 bit – MULTICAST ADDRESS

Il protocollo ICMP (Internet Control Message Protocol):

Il protocollo ICMP è utilizzato per scambiare informazioni tra gli IMP ed i Gateways, in particolare serve per segnalare particolari situazioni sui datagram spediti. L'informazione viene racchiusa in un pacchetto IP con intestazione. I messaggi utilizzati sono:

Destination unreachable: indica che il datagram non è consegnabile perché:
- il gateway è sconosciuto
- l'host o il SAP è sconosciuto
- il pacchetto ha dimensioni elevate e non è frammentabile.

Time exceeded: datagram scartato perché il suo tempo di vita è finito prima che sia spedito.
Parameter problem: errore di sintassi nell'header.
Source quench: si richiede all'host di limitare il traffico generato.
Redirect: avvisa che si è cambiato percorso al datagram perché probabilmente prima è stato instradato male.
Echo: messaggio che deve essere ritrasmesso indietro dal destinatario con un echo reply.
Timestamp, timestamp reply: serve a verificare i tempi di ritardo, il primo porta il tempo di generazione e si attende una risposta col tempo di ricezione.

Il protocollo OSI Internet:

Il protocollo OSI Internet è connectionless, derivato dall'IP, per il sottolivello superiore del livello rete dell'OSI. Realizza servizi di livello 3 senza connessioni :
- N-UNITDATA: trasferimento dati
- N-FACILITY: richiesta di informazioni sul livello 3 (analogo all'ICMP).

L'intestazione è divisa in quattro parti: Fixed part, Address part, Segmentation part, Options part. Le prime 2 sono sempre presenti a meno che non si debbano attraversare gateways. I parametri opzionali sono codificati come:
- codice del parametro
- lunghezza del campo per il parametro
- valore del parametro.

I campi dell'header sono:
PDU lifetime: tempo di vita della PDU, unità 500ms.
SP: segmentation flag, se a 1 è permessa la segmentazione.
MS: more segment flag, se a 0 indica che si ha un solo pacchetto o è l'ultimo pacchetto.
ER: error report flag, se a 1 indica, in caso di errore che si vuole ricevere un pacchetto IP con l'indicazione di questo evento.
Checksum: codice di controllo header.
Total length: numero di byte totali, compresi dati e header.

Instradamento in Internet:

Per quanto riguarda l'instradamento sulla rete di Internet, è necessario conoscere oltre all'indirizzo IP anche il NPA (Network Point of Attachment), ovvero l'indirizzo del destinatario della sottorete o del gateway a cui passare l'informazione. Per questo motivo esistono diversi meccanismi di routing per mantenere aggiornate le informazioni.

Address Resolution Protocol (ARP):

Il protocollo che permette agli host di colloquiare con i gateway interno alla scopo di instradare i datagram è l'ARP. Altri protocolli, gli IGP sono invece utilizzati per instradare datagram tra gateway appartenenti al medesimo Autonomous System.
Il funzionamento è piuttosto semplice. I gateway devono conoscere la coppia HOSTID e NPA sia di tutti gli host che di tutti i gateway delle sottoreti che gli sono direttamente collegate. Per consentire al gateway di avere queste informazioni si possono utilizzare delle tabelle precaricate all'interno di una memoria del gateway stesso, oppure utilizzare un meccanismo che permetta agli host di inviare periodicamente queste informazioni al gateway. Gli host dunque per motivi di efficienza, non contengono tutte le informazioni sugli altri host, ma si rivolgono direttamente al gateway per conoscere gli NPA degli altri host, ciò in assenza di un meccanismo di broadcast. Utilizzando il protocollo ARP dunque l'host viene a conoscenza del NPA del destinatario che conserverà in una cache interna per utilizzarlo quando ce ne sarà bisogno. La sequenza delle operazioni che un host deve effettuare per conoscere l'NPA del destinatario di un datagram IP sono le seguenti:

1. Si rivolge al gateway se non esiste meccanismo di broadcast inviando la sua copia di IP/NPA e l'indirizzo IP del destinatario utilizzando NPA del gateway che già conosce.
2. Il gateway, ricevuta la richiesta, cerca nella sua tabella con chiave IP del destinatario il suo NPA. Spedisce un messaggio al destinatario usando il suo NPA utilizzando come contenuto la coppia IP/NPA del mittente.
3. Il destinatario, ricevuto il messaggio dal gateway, copia la coppia IP/NPA del mittente nella sua cache e risponde al mittente inviandogli il suo NPA.
4. Il mittente infine ricevuto il messaggio dell'host a cui voleva spedire il messaggio, inserisce nella sua tabella in cache la coppia IP/NPA del destinatario e passa l'NPA alla procedura che inizialmente ha provocato questa routine.

Il formato dei messaggi ARP è semplice e consta in 4 campi principali. I primi due riguardano la **lunghezza dell'IP**, perché vi sono vari tipi di indirizzi per l'HOSTID, la **lunghezza dell'NPA**, che dipende dalla sotto-rete. Gli altri due campi, ovvero **Hardware Type** e **Protocol Type** dipendono il primo dal tipo di rete ed il secondo dal tipo di protocollo che ha generato il messaggio.

IGP – Il RIP (Routing Information Protocol):

Il RIP, Routing Information Protocol, è uno degli IGP più comuni e si basa su un algoritmo distribuito. Le informazioni sono diffuse tra i gateways sotto forma di vettore di distanze. Le distanze possono essere misurate sia come numero di sotto-reti da attraversare che come ritardo misurato per l'invio di un datagram. Ogni gateway, inizialmente possiede informazioni inerenti esclusivamente alle sotto-reti a lui direttamente collegate, organizzate in una tabella detta d'instradamento costituita da due colonne. La prima contiene le informazioni sul nome delle sotto-reti e la seconda contiene la distanza rispetto ad un determinato gateway. Periodicamente, i gateways si scambiano i vettori delle distanze così da mantenersi aggiornati sui diversi percorsi della rete. Ogni gateway, quando riceve questo pacchetto di informazioni, valuta se conosce già un certo percorso e qual è il costo del percorso e se più vantaggioso del suo, lo sostituisce. Ogni riga della tabella contiene un timer che permette di mantenere abbastanza aggiornate le informazioni registrate. Anche se questo algoritmo è molto diffuso, esistono due problematiche che in determinate situazioni possono diventare rilevanti. Purtroppo i percorsi memorizzati nelle routing tables non sono necessariamente ottimi, in quanto percorsi di costi uguali a quelli contenuti nelle tabelle vengono scartati. Il secondo problema riguarda i loop. Infatti a volte un pacchetto, prima di arrivare a destinazione percorre un percorso chiuso.

IGP – L'OSPF (Open Shortest Path First):

L'OSPF è un protocollo IGP nato per garantire alcune specifiche molto utili all'interno degli autonomous systems. Innanzi tutto, si basa su un algoritmo pubblico in grado di garantire una certa sicurezza e il supporto per l'incapsulamento (TUNNELING). Altre caratteristiche rilevanti del protocollo sono la capacità di gestire e supportare diverse metriche (lunghezza dei percorsi, ritardi), di adattarsi dinamicamente alle configurazioni di rete e la possibilità di instradare su percorsi differenti valutando l'intensità di traffico su un determinato percorso. Anche il supporto per organizzazioni gerarchiche e la possibilità di smistare il traffico su più percorsi diversi rendono OSPF un buon candidato come protocollo interno per un autonomous system. I routers OSPF considerano la rete come un grafo costituito da nodi e da archi pesati secondo una metrica opportuna.

IGP – Messaggi scambiati fra routers:

I routers appartenenti allo stesso Autonomous System, si scambiano informazioni per individuare i percorsi ottimi e per stabilire verso quali vicini bisogna instradare un determinato messaggio affinché arrivi ad un determinato host di una particolare sotto-rete. Un Autonomous System è una regione in cui tutti i suoi router utilizzano un algoritmo IGP comune e controllata da un unico gestore o gruppo di gestori. I messaggi scambiati sono:

Hello: messaggio inviato verso tutti i canali punto a punto non appena viene attivata l'apparecchiatura di rete. Serve a conoscere e comprendere chi sono i vicini.

Link State Update: messaggio che contiene tutto il grafo con gli archi pesati secondo una determinata metrica e la situazione attuale delle linee, visto dal router mittente ed inoltrato a tutti quelli vicini. E' confermato e numerato.

Link State Request: messaggio con cui si esorta il router destinatario ad inviare messaggi di tipo Link State Update.

Data Base Description: messaggio che viene inviato dal mittente per descrivere in sequenza tutti i messaggi ricevuti di tipo Link State Update.

Utilizzando questi messaggi, i router comprendono quale vicino ha le informazioni più aggiornate.

L'utilizzo di questi messaggi serve per individuare i percorsi ottimi e sub-ottimi. I routers backbone, calcolano questi percorsi per tutte le aree dell'autonomous system utilizzando i dati che provengono da tutte le aree, comprese quelle di confine. Attraverso i routers di confine, le informazioni sui percorsi elaborate dai backbone raggiungono tutte le aree dell'a.s.

EGP – Il comportamento dei routers:

I routers EGP, Exterior Gateway Protocol sono le apparecchiature di rete incaricate di connettere diversi autonomous systems. L'EGP deve assolutamente utilizzare un algoritmo comune. Inizialmente, ogni router EGP contiene una tabella in cui sono descritti tutti i routers vicini. Utilizzando queste informazioni, ogni routers può iniziare o interrompere relazioni con i vicini, scambiare messaggi d'errore, verificare la presenza dei vicini, inviare e richiedere informazioni per l'aggiornamento delle routing tables e comprendere come instradare un messaggio per raggiungere una determinata sottorete di un determinato autonomous system. Il protocollo di routing utilizzato dai gateways esterni è il **BGP** (Border Gateway Protocol). Questo protocollo permette ai routers esterni di scambiarsi messaggi contenenti vettori di distanza con in più l'informazione sull'intero percorso in uso per raggiungere ciascuna destinazione. Per il BGP esistono tre tipi di autonomous systems: STUB, DI TRANSITO e MULTICONNESSA.

IP versione 6:

Ultimamente è stata introdotta la nuova versione del protocollo IP che varia di molto le capacità e le prestazioni della versione 4. Innanzi tutto lo spazio di indirizzamento passa da 32 a 128 bit. In più, il protocollo è in grado di avere più intestazioni con un formato più semplice ma allo stesso tempo più flessibile per future estensioni. La possibilità di gestire diversi pacchetti come unico flusso di dati e la sicurezza degli stessi garantiscono una qualità maggiore rispetto all'edizione precedente. I campi dell'intestazione sono:
Versione: in questo caso 6.
Flow label: serve ad identificare un flusso di pacchetti.
Prossima intestazione: identifica il tipo della prossima intestazione.
Lunghezza di carico o payload length: identifica la dimensione del carico dati.
Priorità: priorità nell'instradamento del datagram.
Limite di salti: come il TTL.
Le intestazioni possono essere multiple, non esiste la possibilità di frammentare il pacchetto perché i gateways Ipv.6 non lo permettono. Per definire la lunghezza di carico del percorso si utilizza un meccanismo che prevede di diminuire la dimensione del datagram finchè non si ricevono più messaggi ICMP del tipo destination unrechable.
Non esistono classi di indirizzo, ma solamente tre tipologie fondamentali:
Unicast: verso un unico destinatario.
Multicast: verso diversi destinatari appartenenti tutti allo stesso gruppo ma anche geograficamente molto lontani.
Anycast: verso diversi destinatari che però condividono tutti lo stesso prefisso. Il datagram viene consegnato al destinatario più vicino.
Per convertire un protocollo IP da versione 6 a versione 4 e viceversa si utilizza un meccanismo di TUNNELING.

Il meccanismo del DHCP (Dynamic Host Configuration Protocol):

Il meccanismo del DHCP è molto utile per assegnare dinamicamente ed in fretta gli indirizzi IP alle macchine di una rete locale in modo da evitare la configurazione manuale. Esistono due tipi di macchine: i server, a cui sono assegnati indirizzi IP permanenti o statici e che hanno il compito di distribuire gli indirizzi agli host detti client. L'indirizzo può essere permanente, ovvero una macchina ha sempre lo stesso indirizzo, o volatile, il suo IP può cambiare.
Il meccanismo di interrogazione dei server DHCP e di distribuzione degli indirizzi è semplice:

1. Appena attivo, l'host spedisce sulla rete locale (in broadcast) un segnale di discover per comprendere quali server DHCP siano disponibili.
2. Uno o più server DHCP, se disponibile risponde.
3. L'host può dunque effettuare la sua richiesta direttamente verso quel server DHCP.
4. Gli indirizzi del server scoperti con discover sono salvati su disco perché potrebbero ritornare utili.

Si applica questo meccanismo per evitare di intasare la rete solo con richieste DHCP.
I messaggi DHCP sono inclusi in messaggi UDP, la porta di ricezione del server è 67, quella del client è 68.

Livello Trasporto

Qualità del servizio:

La qualità del servizio è descrivibile attraverso l'utilizzo di diversi parametri. Quando si richiede un servizio si può indicare il valore accettabile, il valore desiderato ed il valore

inaccettabile. Per decidere se accettare il servizio offerto, si valutano le proprie funzionalità, quelle del livello rete e le risorse disponibili in quel momento.
Alcuni parametri della qualità del servizio sono:
Connec. Establishment delay: tempo per stabilire una connessione.
Connec. Establishment failure prob.: probabilità di non riuscire ad instaurare una connessione del tempo stabilito.
Throughput: n. di byte al secondo trasferibili.
Transit delay: tempo per uscire dell'entità trasporto ricevente.
Residual error rate: prob. di errore residua per le TPDU.
Transfer failure: probabilità per una TPDU di non soddisfare le caratteristiche di velocità promesse.
Connection release delay: tempo per chiudere una connessione.

Indirizzamento:

Per aprire un connessione o utilizzare un servizio senza connessione, è necessario conoscere il TSAP destinatario. Normalmente, è utile mettere in comunicazione due processi che si trovano su macchine diverse per eseguire determinate operazioni. Per fare ciò, però il processo mittente deve conoscere il TSAP di quello destinatario. Per localizzare questo indirizzo si utilizza questa procedura:

1. Il processo mittente, ovvero quello che vuole instaurare il rapporto, utilizzando il proprio TSAP e il TSAP di un processo che è definito Name Server si mette in contatto richiedendo il TSAP del processo destinatario. Ciò è possibile perché il servizio di name server contiene una base dati distribuita con tabelle che associano nome_processo-TSAP.
2. Il name server quindi spedisce al mittente il TSAP per il processo richiesto.

E' utile fare attenzione che per ogni processo generato o alterato è necessario aggiornare la base dati del name server. L'indirizzo del TSAP di solito è gerarchico, nell'OSI ind. TSAP = n. TSAP – n. NSAP – n. LSAP – ind. sul canale.

Meccanismo per l'apertura di una connessione:

Vi sono diverse problematiche nell'apertura di una connessione, prima fra tutte le possibilità di duplicare delle TPDU che come numero di sequenza potrebbero ancora essere disponibili in rete. Per questo motivo, esiste un meccanismo per permette di aggirare il problema.
Nella prima fase, ogni dispositivo presenta un clock, quando si apre una connessione, si utilizzano i bit bassi di questo clock per assegnare i numeri alle TPDU (ogni connessione parte con un numero diverso). Tuttavia, il numero di sequenza deve essere tale da ripetersi solo quando non siano più presenti altre TPDU con lo stesso n. di seq. sulla rete. Per questo motivo si utilizzano delle regioni proibite, in cui non è possibile utilizzare lo stesso numero per la TPDU e per risolvere il problema si risincronizzano i numeri di sequenza. Una problematica frequente può provenire dalla frequenza del clock, che influenza direttamente la capacità di trasmissione delle TPDU.

Meccanismo di chiusura di connessione:

Una connessione può essere terminata in tre modi distinti:

1. Uno dei due partecipanti decide di chiudere la connessione ed invia un DISCONNECT.REQUEST
2. Entrambi gli utenti inviano una DISCONNECT.REQUEST
3. Il livello 3 invia una DISCONNECT.INDICATION perché ha rilevato un problema nella connessione.

La chiusura della connessione può dare origine alla perdita di dati se la stessa viene chiusa senza assicurarsi che l'altro utente abbia finito di inviarne.

Il protocollo TCP (Transmission Control Protocol):

E' un protocollo di livello trasporto, affidabile e connesso. Affidabile perché garantisce all'utente di ricevere i dati in ordine e di non perderli. Connesso perché ogni pacchetto segue sempre lo stesso percorso come in un circuito virtuale. I dati sono considerati dal TCP come un flusso continuo di bytes, o più precisamente di ottetti.

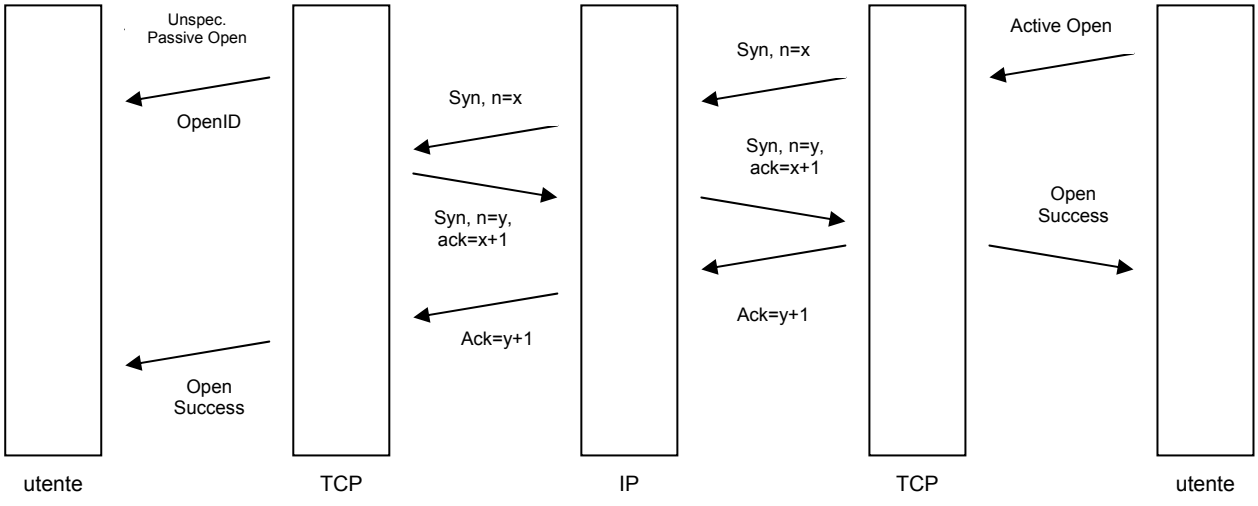
Le primitive:

- UNSPECIFIED-PASSIVE-OPEN.request: consente di porsi in ascolto a qualsiasi utente voglia connettersi.
- FULL-PASSIVE-OPEN.request: consente di porsi in ascolto a un DETERMINATO utente che si vuole connettere.
- ACTIVE-OPEN.request: consente ad un utente di connettersi ad un altro.
- ACTIVE-OPEN-WITH-DATA.request: come l'ACTIVE-OPEN ma in più permette di trasferire i primi dati.
- SEND.request: primitiva dedicata al trasferimento di dati.
- ALLOCATE.request: richiede al ricevitore di allocare memoria in ricezione in previsione dei dati che saranno trasmessi.

Formato delle PDU:

Source port e Destination port: le porte sono gli equivalenti dei TSAP nell'OSI.
Sequence number : indica il numero di byte già trasmessi finora.
Window: indica il credito di trasmissione attuale (in byte).
Urgent pointer: indica il primo byte di dati non urgenti, ha senso se il flag URG è posto ad 1.
Options: l'unica opzione definita serve a trasmettere la dimensione massima della TPDU accettabile all'apertura della connessione.
URG: flag, indica la presenza di dati urgenti.
ACK: indica la presenza di ack in piggybacking.
PSH: indica che i frammenti verranno riassemblati e consegnati all'utente.
RST: indica al destinatario di eseguire un reset della connessione.
SYN: utilizzato all'apertura della connessione per concordare l'inizio della numerazione dei byte.
FIN: indica la fine dei dati da trasmettere in questa direzione.

Apertura di una connessione TCP



Ritrasmissione dei dati:
Il ricevitore può tenere i dati arrivati fuori sequenza, ma invia ack solo per i byte ricevuti correttamente senza gap. Se non si ricevono ack entro il timeout è possibili utilizzare due approcci differenti:

1. ritrasmettere tutte le PDU senza ack.
2. ritrasmettere solo la più vecchia sperando che serva a riempire il gap.

Recupero degli errori nella classe 4:
Si possono utilizzare i checksum per rilevare errori di trasmissione, oppure una serie di timers per rilevare condizioni anomale. Se E è il tempo di attraversamento, A è il tempo di massimo tra la ricezione di una PDU e la generazione dell'ack, allora il tempo T di ritrasmissione è:

$$T = 2E+A+\text{tempo di elab. locale di 1 PDU}$$

Controllo di congestione:
Il controllo di congestione comprende due meccanismi distinti:

1. la finestra di trasmissione: riguarda la congestione del ricevitore.
2. la finestra di congestione: riguarda la congestione generale della rete.

In ogni istante, il numero di bytes che si è abilitati a trasmettere è il MINIMO fra le due finestre.

Dimensione della finestra di congestione:
All'inizio della connessione, la finestra di congestione è pari ad 1 segmento. Esiste una soglia, che al massimo può raggiungere 64 KB. La finestra di congestione non può mai superare la soglia. La procedura di gestione della finestra di congestione è:

1. si attendono gli ack e ogni volta che arrivano si aumenta la finestra di 2^n, in modo esponenziale senza superare la soglia. Se si raggiunge la soglia, l'aumento è soltanto lineare.
2. Se scatta un timeout, la soglia viene portata a metà della finestra di congestione e la finestra inizia dalla dimensione di 1 segmento. L'arrivo di pacchetti ICMP source quench vengono trattati come timeout.

Misura del tempo di andata-ritorno:
Un parametro importante è il RTT, che viene aggiornato ogni volta che si riceve un ack in tempo. Sia M il tempo fra la spedizione di un pacchetto e l'arrivo del corrispondente ack, allora:

$$RTT = \alpha RTT + (1 - \alpha)M$$

Normalmente α vale 7/8. Per determinare il valore del timeout si utilizza un altro parametro che stima la deviazione media dei valori del RTT.

$$D = \beta D + (1 - \beta)|RTT – M|$$

Solitamente, il valore del timer è posto a RTT+4D.

Il protocollo UDP (User Datagram Protocol):

Rispetto ad IP aggiunge un meccanismo di porte come in TCP, non è affidabile e neppure connesso. Le primitive sono due: SEND e DELIVER. Il pacchetto dell'UDP è formato dai campi: Source port, Destination port, Length, UDP Checksum.

Livello Applicazione

Remote Procedure Call (RPC):

L'RPC è un metodo asimmetrico, utilizzato per eseguire procedure remote utilizzando un server che per esempio conserva il codice di un applicazione. Il modello funziona in modo abbastanza semplice: 1. il cliente esegue una richiesta al server, 2. il server esegue la richiesta eseguendo l'operazione, 3. il server invia la risposta al cliente. Questo metodo è molto utile per diminuire la necessità di aprire una sessione per ogni richiesta di operazione. L'RPC tende a far vedere al cliente l'operazione come una chiamata a procedura. Il passaggio dei parametri è uno degli elementi più critici. Infatti per i parametri e le strutture passate by value, non ci sono problemi, ma ce ne sono diversi se i parametri e le strutture sono passati by reference. Utilizzando un parametro puntato nel cliente, l'unica soluzione possibile per passare un valore efficace del parametro alla procedura remota consiste nel ricercare il valore effettivo e passarlo come tale. Tuttavia a volte vi possono essere dei problemi. In generale, ogni parametro del client viene impacchettato dal client stub, spedito con l'entità dedicata al trasporto al server stub che avrà il compito di disimpacchettare il parametro e passarlo al server che genererà la risposta.

<u>Localizzazione dei server:</u>

Quando un server viene attivato, deve segnalare la sua presenza e dare un identificatore di 32 bit scelto da una base dati centrale. Ogni client, si rivolgerà al client stub che richiederà il server per svolgere quella procedura.

<u>La gestione delle eccezioni:</u>

Nelle procedure remote, è possibile che l'esecuzione possa avere dei problemi, quindi è necessario che esista un trattamento delle eccezioni che non restituisca il controllo al chiamante ma alla routine del trattamento delle eccezioni.

<u>Crash del servitore:</u>

In caso di crash del servitore esistono tre alternative:

1. Aspettare un tempo infinito la risposta.
2. Usare un timeout che scadendo segnala il problema all'utente che deve trattare l'eccezione relativa.
3. Usare un timeout che scandendo genera una ritrasmissione.

Quando il server ritorna attivo dopo il crash deve compiere di nuovo la registrazione presso la base dati, così gli sarà assegnato un altro identificativo. In questo modo anche il cliente dovrà ricontattare la base dati per conoscere il nuovo identificativo del server.
Dopo un crash, la soluzione migliore non è sempre ritrasmettere, anzi può non esserlo affatto. La ritrasmissione si effettua solamente con operazioni IDEMPOTENT. In generale si possono distinguere tre casi per l'esecuzione:

1. UNA SOLA VOLTA: in caso di problemi la procedura passa al trattamento delle eccezioni.
2. UNA VOLTA AL MASSIMO: la RPC ritorna al chiamante che deve controllare che non vi siano stati problemi, perché in questo caso, l'operazione non è stata eseguita.
3. ALMENO UNA VOLTA: in caso di crash viene effettuata la ritrasmissione.

Crash del cliente:
Se il cliente va in crash mentre attende una risposta, la sua richiesta diventa orfana. Se il cliente riprende la RPC prima che sia scaduta la richiesta orfana, questa può generare confusione. Per risolvere questi problemi vi sono 3 tipi di soluzione:

1. LO STERMINIO DEGLI ORFANI: lo stub cliente ha un file di log dove sono registrate tutte le richieste pendenti, che vengono cancellate non appena giunge una risposta ad una determinata richiesta. Alla ripresa del cliente, si analizza il log e si chiede al servitore di uccidere ricorsivamente tutti gli orfani. Ovviamente per utilizzare questa soluzione il file di log deve assolutamente sopravvivere ai crash.
2. CANCELLAZIONE A TEMPO: quando viene inviata una richiesta, le viene assegnato un tempo massimo di completamento, il server se non riesce a rispettare la scadenza, può chiedere delle proroghe e possono essere concesse. Se la proroga non viene concessa, l'esecuzione viene bloccata. Alla ripresa da un crash, il cliente aspetta un tempo almeno pari all'entità di una singola proroga per essere sicuro che gli orfani siano già morti.
3. REINCARNAZIONE: a volte non si riesce a sterminare tutti gli orfani perché magari si trovano in zone temporaneamente non raggiungibili. Per ovviare a questa situazione si utilizza la suddivisione in epoche numerate. Quando il cliente riparte, invia a tutti il suo nuovo numero d'epoche, provocando lo sterminio degli orfani.

In certi casi, può essere problematico eliminare degli orfani se hanno effettuato un lock.

RPC SUN (RFC 1057):
Definisce il protocollo RPC usato soprattutto nelle macchine UNIX. Non definisce meccanismi di recupero, che deve essere realizzato nello stub cliente o servitore. Funziona su vari tipi di trasporto, in particolare su TCP ed UDP.

Funzioni svolte dall'RPC:
Le funzioni svolte sono:

1. Definire una specifica unica per la procedura da chiamare.
2. Associare le richieste con le risposte.
3. Fornire un meccanismo per autenticare il cliente ed il servitore.
4. Fornire meccanismi per la segnalazione di errori: errori nel protocollo RPC, errore nel numero di programma remoto, errori nella chiamata della procedura, errore nell'autenticazione, altri tipi di errore.

Chiamata RPC:
La chiamata RPC contiene:

- N. DEL PROGRAMMA: corrisponde ad una libreria RPC, questo numero deve essere unico e viene assegnato da un'entità centrale.
- N. VERSIONE: serve per identificare la versione in modo che il server possa rispondere correttamente secondo le esigenze dei clienti.
- N. PROCEDURA: identifica l'operazione che si vuole svolgere.

Questi parametri identificano univocamente la procedura da chiamare.

Autenticazione:
La chiamata contiene due campi: CREDENTIALS e VERIFIER, mentre la risposta contiene solo VERIFIER. I meccanismi di autenticazione sono 4:

1. NULLO: non è prevista autenticazione.

2. UNIX: il cliente deve fornire nome macchina e UID e GID (questi ultimi non sono controllati).
3. DES: usa una chiave DES che il cliente ha passato nella prima chiamata (usando meccanismo a chiave pubblica) per cifrare il timestamp.
4. CHIAVE PUBBLICA: si una un meccanismo simile a RSA per definire la chiave di colloquio.

Altri meccanismi RPC:
Esistono altri due meccanismi RPC:

1. BATCHING:
   - permette di spedire varie richieste senza avere risposte.
   - Il server accetta le richieste e non risponde mai.
   - Il flusso di richieste viene solitamente terminato da una chiamata ad una procedura vuota che fornisce una risposta vuota.
   - Il client non viene bloccato dalla chiamata a procedura.
2. BROADCAST:
   - Il server broadcast risponde solo in caso di esecuzione corretta.
   - Il cliente può ricevere più di una risposta.
   - Usato di solito su UDP.

Il World Wide Web (WWW):

Per localizzare le risorse del WWW si utilizza URI (Uniform Resource Identifier) che comprende URL ed URN. URI serve ad identificare in modo univoco una risorsa sulla rete di Internet, secondo la collocazione URL o secondo il nome URN.

Sintassi degli URL HTTP:

<**schema**> ":" "//" <**host**> [":"<**porta**>] [<**path_assoluto**>]

dove **schema** identifica il tipo di protocollo utilizzato, in questo caso http, **host** identifica il nome della macchina come registrata al DNS, **porta** dove si trova il server che gestisce la risorsa, normalmente 80 e il **path_assoluto** è il direttorio dove si trova la risorsa, specificato utilizzando il formato UNIX.

Operazioni HTTP:
Le operazioni sono basate sul modello client-server, quindi le interazioni sono di tipo richiesta-risposta. Le interazioni avvengono utilizzando una connessione TCP aperta dal client prima di effettuare la richiesta.
I contenuti della richiesta sono: METODO, URI, VERSIONE HTTP, MIME (modificatori della richiesta, informazioni cliente, corpo).
I contenuti delle risposte sono: (riga di stato) VERSIONE HTTP, CODICE ERRORE/SUCCESSO, MIME (informazioni sul server, metainformazioni sulle entità, corpo).
Sia le richieste che le risposte sono in formato ASCII ad eccezione di quelle parti MIME codificate in modo diverso.

Formato di una richiesta HTTP:
Il formato di una richiesta HTTP è il seguente:

<metodo> SP <URI> SP "HTTP/" <num.vers.> CRLF

dove **metodo** identifica il metodo usato e CRLF è il carattere di a capo, SP è lo spazio. URI deve essere un path assoluto.
Le intestazioni sono di tipo generale, di richiesta, dell'entità. Ciascuna intestazione è costituita da vari campi, il formato è:

<nome campo> ":" <valore> CRLF

Il corpo dell'entità, costituito da un'insieme di byte la cui lunghezza e codifica è specificata nei parametri della intestazione, il corpo dell'entità è presente solo quando richiesto dal metodo invocato.

Metodi ammessi:
GET: produce il trasferimento all'UA della risorsa specificata nell'URI, se la risorsa è un oggetto che fa una elaborazione, all'UA verrà spedito il risultato, può essere condizionato da alcuni parametri.
POST: invia nell'entità dei dati in ingresso per la risorsa indicata nell'URI, le risorse non debbono finire nella cache.
HEAD: come il GET, tranne che la risposta non contiene entità ma solo le intestazioni
PUT: richiede la creazione dell'URI specificata presso il server, con contenuto preso dall'entità inviata. Se l'URI esiste già, ne viene rimpiazzato il contenuto.
DELETE: richiede la cancellazione della risorsa specificata.
OPTIONS: è una richiesta per conoscere le opzioni di comunicazione disponibili lungo la catena di trasmissione.
TRACE: serve a verificare cosa viene ricevuto alla fine della catena delle richieste/risposte.
CONNECT: metodo riservato ai proxy che possono diventare tunnel.

I campi delle intestazioni:
Possono contenere diversi parametri come DATE, USER-AGENT, MIME-VERSION, IF-MODIFIED-SINCE, PRAGMA, FROM …

Formato della risposta HTTP:
Il formato di una risposta HTTP è il seguente:

"HTTP/" <num.versione> SP <status code> SP <frase espl.> CRLF

dove lo **status code** è un numero decimale a tre cifre:

- 1xx: codici riservati
- 2xx: codici di successo
- 3xx: bisogna eseguire ulteriori azioni per completare la richiesta
- 4xx: errori nella richiesta
- 5xx: errore nell'esecuzione di una richiesta

la **frase** è una stringa di caratteri che spiega l'errore.
Le intestazioni sono di tipo generale, di risposta e di entità.
L'entità è opzionale.

I campi delle intestazioni:
Anche nelle risposte esistono diversi parametri per le intestazioni, alcuni sono: LOCATION, SERVER, RETRY-AFTER, WWW-AUTHENTICATE …

Proxy, Gateway e Tunnel:
Un **proxy** è un package di software che permette ad un User Agent, per esempio un browser web, di contattare un server di cui non possiede internamente il client adeguato. (per esempio un server FTP). Lo UA si rende conto di non parlare con un server HTTP. Funzione di caching.
Un **gateway** è un package di software che permette di utilizzare un server non HTTP di essere usato mediante HTTP. Il cliente può non accorgersi che dall'altra parte non c'è un server HTTP. (per esempio un server RDBMS). Funzione di caching.
Un **tunnel** è un sistema intermedio che non interpreta, né modifica il formato delle richieste e risposte HTTP. Non esiste la funzione di caching.

Utilizzo della cache:
Ogni sistema intermedio come proxy e gateway possono tenere nella cache alcune informazioni ricevute in precedenza dal server, allo scopo di servire più velocemente il cliente che ne facesse richiesta. Il problema delle cache è la correttezza dei dati. In generale una risposta può essere generata a partire dal contenuto della cache se:

- Il contenuto della cache è stato controllato come equivalente a quello del server.
- Il contenuto è sufficientemente "fresco".
- La risposta è **not modified**, **proxy redirect**, oppure è un errore.

La cache NON DEVE MAI tentare di invalidare delle risposte non valide che normalmente invierebbe al cliente. Una cache si definisce privata se è accessibile ad un solo cliente, altrimenti è condivisa. Il parametro che serve a controllare l'uso delle cache è il *cache-control*.

Trasferimento frammentato:
Il corpo di una risorsa può anche essere trasferito diviso in frammenti, utilizzando il parametro *transfer-coding* = *chunked*. Lo scopo è di poter trasferire il contenuto di una risorsa creata dinamicamente mano a mano che viene creata. La risorsa può avere delle intestazioni al fondo.

Trasferimento parziale:
Il cliente può richiedere solo una parte della risorsa, a patto di sapere quale. Si utilizza il parametro *range* per specificare il range di bytes voluti.

Il Directory Service:

Per contattare i diversi host esiste un sistema che semplifica le cose, permettendo di associare gli indirizzi numerici con nomi simbolici. Questa associazione è compito del Directory Service.

Domain Name System:
Il Domain Name System è il Directory Service di Internet, segue uno schema gerarchico di assegnazione dei nomi, per rendere i nomi simbolici unici si utilizza il suffisso del dominio. All'interno di ciascun dominio è possibile assegnare liberamente i nomi simbolici.

Domain Name Server o DNS:
Spesso per ogni dominio esistono uno o più Domain Name Server, ciascuno di questi contiene al proprio interno una tabella di associazione tra indirizzo IP e nome simbolico dell'host. Questa base dati si definisce Domain Information Base o DIB. Ogni DNS contiene un DIB in cui sono conservate tutte le informazioni sugli host o DNS figli e sui DNS padri. (considerando la gerarchia).

Ottimizzazione del sistema e meccanismi di accelerazione:
E' spesso opportuno utilizzare una gerarchia collassata, che permetta cioè di aumentare la dimensione dei DIB e diminuire la quantità di DNS in modo da diminuire le richieste di risoluzione. I meccanismi più opportuni per accelerare la risoluzione dei nomi è l'introduzione dei **name caches** nei DNS e dei **name resolver** negli host. Entrambi questi meccanismi possono contenere sia informazioni statiche che dinamiche. Tuttavia l'utilizzo di cache con persistenza elevata possono dare problemi in caso di riconfigurazione della rete.

La Posta elettronica:

I protocolli di posta elettronica più utilizzati sono l'**SMTP** ed il **POP3**. L'SMTP permette il trasferimento dei messaggi da un MTA (Message Transfer Agent) ad un altro. Mentre il POP3 permette il trasferimento dall'MS (Message Store) del server di posta elettronica all'UA (User Agent) ovvero il cliente di posta elettronica. Quindi l'SMTP non si occupa di consegnare la posta ai destinatari finali, è un protocollo atto solo al trasferimento dei messaggi. Molto spesso il cliente SMTP ed il server SMTP sono separati.

Formato dei messaggi:
Gli indirizzi di posta elettronica sono composti da due parti, quella **locale** e quella **globale**. Ovvero il nome della mailbox all'interno del server di posta e il nome simbolico del server registrato nei DNS.
Il messaggio contiene due parti:

- **Intestazione:** in formato ASCII, con una serie di parametri che servono all'instradamento, all'interpretazione del contenuto (all'arrivo) e alla gestione locale dei messaggi.
- **Corpo:** vero e proprio testo del messaggio perlopiù in ASCII con righe abbastanza corte (meno di 80 caratt.) perché i sistemi intermedi attraversati potrebbero tagliarle.

Campi dell'intestazione:
I campi dell'intestazione sono:
To: mailbox del destinatario.
From: nome autore del messaggio.
Cc: destinatari in chiaro a cui inviare una copia.
Bcc: destinatari a cui è oscurato il nome degli altri destinatari a cui inviare una copia.
Date:
Sender: indirizzo e-mail di chi ha spedito il messaggio.
Subject: oggetto del messaggio.
Recived: riga aggiunta dai sistemi intermedi.
Return-Path: può servire ad identificare la strada per la risposta.
Reply-to: indirizzo a cui inviare la risposta.
Message ID: codice identificativo del messaggio.
In-reply-to: message-id del messaggio di cui quello attuale è la risposta.
References: message-id dei messaggi a cui si fa riferimento.
Keyword: parole chiave definite dall'utente.

Gateways postali:
Non tutti i server di posta utilizzano il protocollo SMTP, con i gateway, è possibile modificare le intestazioni dei messaggi per renderli compatibili, o modificare i livelli sottostanti.

MIME (Multipurpose Internet Mail Extension) :
Questo formato permette di includere informazioni diverse da un semplice testo ASCII, utilizzando proprie intestazioni all'interno del messaggio vengono utilizzate dal ricevitore per interpretare correttamente il contenuto. Le righe di intestazione MIME sono sempre in ASCII, come anche le righe del contenuto.

Campi MIME:
MIME version: versione.
Content type: descrizione del tipo di informazione contenuta, viene specificato come <tipo>/<sottotipo>.
Content transfer econding: indica come le informazioni specificate da conten-type vengono rappresentate nel messaggio. BASE64, 7BIT, 8BIT BINARY …

Esempi di tipi sono Text, Application, Image, Audio, Video. Esempi di sottotipi sono Plain, Richtext, Gif, Jpeg …

Codice Mobile:
Il sistema di posta elettronica può essere utilizzato come mezzo di trasporto per il codice mobile. Codice mobile indica un programma che si muove attraverso la rete ed esegue parte della sua elaborazione su ogni macchina su cui arriva. Per potersi spostare si fanno spedire codice e variabili in messaggi di posta elettronica sfruttando un tipo di contenuto application/x- … All'arrivo l'interprete MIME riconosce il tipo e lancia un apposito interprete dandogli in input il file del messaggio ricostruito. Tutte le ricerche vengono fatte in locale, non si guadagna nulla nella velocità di interrogazione, risparmio però i trasferimenti sulla rete.

USENET:
E' una rete logica per la diffusione delle informazioni attraverso delle "bacheche" elettroniche. Ogni utente può leggere ed affiggere messaggi. Tutte le informazioni presenti sul server sono organizzate in gruppi di discussione a seconda dell'argomento trattato e alla specificità. Questo sistema di propagazione può funzionare con diversi tipi di rete, tuttavia è necessario un nome o indirizzo Internet, il protocollo usato è NNTP ovvero News Network Transfer Protocol.
Alcuni gruppi non sono controllati, ovvero tutti possono affiggere messaggi, invece altri sono moderati e solo certe persone possono affiggere messaggi.

Propagazione delle news:
Il sistema è visto come un grafo orientato dove i nodi rappresentano i server e agli archi sono associate liste di newsgroups e rappresentano i percorsi di propagazione delle news. Si utilizza il flooding secondo i labels degli archi. Ogni messaggio, conserva il percorso effettuato per evitare di compiere loop. La propagazione può avvenire appena la news è stata affissa, o dopo qualche tempo. E' possibile trasferire una news alla volta o tutte insieme. Il trasferimento può essere anche fatto in differita se un server in un determinato momento è irraggiungibile. Periodicamente secondo le regole locali i servers cancellano i messaggi troppo vecchi. In alcuni casi, però esistono server storici che tengono traccia delle discussioni più significative. Un messaggio diretto ad un newsgroup può anche essere trasferito via email.

Formato delle news:
Le news sono costituite da testo ASCII e sono composte da intestazioni e da un corpo. Le intestazioni obbligatorie sono:

Date:
From:
Path: elenco news server attraversati, il primo è in fondo.
Message-id:
Newsgroup: lista di newsgroup in cui affiggere la news
Subject:
Altre intestazioni opzionali sono: REPLYTO, SENDER, EXPIRES, REFERENCES, CONTROL, KEYWORDS, SUMMARY, APPROVED…

I comandi:
I comandi sono messaggi di controllo che possono essere inviati tra server all'interno di messaggi di news nel campo CONTROL o attraverso un terminale virtuale sulla porta 119 del server di news. I comandi principali sono:
IHAVE <message-id>: si indica che il messaggio specificato è disponibile.
SENDME <message-id>: risposta al comando IHAVE, si richiede il trasferimento del messaggio.
NEWGROUP <groupname>: si vuole creare un nuovo gruppo di discussione.
RMGROUP <groupname>:
LIST
… e altri.

MHS (Message Handling System):

E' un protocollo di posta elettronica dell'ISO, offre le funzioni tipiche di un sistema di posta elettronica che non sono fornite dal trasferimento file: gestione liste di distribuzione, possibilità di essere avvisati dell'arrivo di nuovi messaggi, gestione destinatari alternativi, strutturazione del testo in mittente, destinatario, oggetto, priorità di spedizione dei messaggi.

Struttura del MHS:
E' suddiviso in sottolivelli, lo USER AGENT ENTITY (UAE), MESSAGE TRANSFER AGENT ENTITY (MTAE) e SUBMISSION AND DELIVERY ENTITY (SDE).
User Agent Entity: fornisce un'interfaccia all'utente, effettuando eventuali conversioni locali.
Message Transfer Agent Entity: esegue il routing dei messaggi, fornisce memoria per conservarli, esegue delle operazioni di gestione.
Submission and Delivery Entity: fornisce meccanismi di comunicazione con MTAE, senza routing.

Funzioni svolte dal MSH:
Questo sistema permette di **comporre** messaggi, effettuare **reporting**, **trasferirli**, **convertirli**, **formattarli**, **disposizione** per gestire i messaggi.

Sistema di gestione:
Il sistema di posta coinvolge vaste aree geografiche e numerose organizzazioni sia pubbliche che private. Ogni organizzazione controlla un Management Domain (MD) composto da almeno un MTAE e da zero o più UAE.
Le organizzazioni si distinguono in pubbliche e private. Quelle pubbliche gestiscono gli Administration Management Domain (ADMD), quelle private i Private Management Domain (PRMD). Ciascuna organizzazione decide il livello di interfaccia permesso all'utente.

Limitazioni dei domini privati:
Vi sono alcune limitazioni per i domini privati. Un dominio privato può esistere solo all'interno di un paese, può avere accesso a più amministrazioni, non può svolgere operazioni di routing fra amministrazioni ed infine gli ADMD svolgono funzioni di controllo sui PRDM per verificare che le operazioni di accounting, logging, qualità del servizio, siano svolte correttamente.

Modello dei messaggi:
Il modello dei messaggi è del tutto simile a quello dell'SMTP. Consta di un'intestazione, di un corpo e di una busta o envelop.
L'intestazione contiene **nome del mittente**, **utenti che hanno autorizzato la spedizione**, **destinatari**, **indirizzo per risposte**, **riferimenti ad altri messaggi**, **oggetto**, **importanza**, **sensitività (personale, privato, riservato)**.
Il corpo contiene la rappresentazione dell'informazione che si vuole trasferire col messaggio, può essere composto da varie parti (ASCII, fax, teletext, videotel) accompagnate dall'indicazione del formato utilizzato.

Conversioni:
La conversione fra il formato originario e quello del dispositivo destinatario può essere disabilitata, implicita, esplicita. Per un UAE è possibile informare un MTAE dei formati che è in grado di trattare.

Servizio di Probe:
Il servizio di PROBE serve a verificare il funzionamento corretto del sistema, verifica anche la compatibilità dei formati, della lunghezza e dei contenuti.

Gestione di Rete, SNMP (Simple Network Management Protocol):

Le operazioni di gestione della rete includono la gestione della configurazione (attivazione e disattivazione di componenti di rete), gestione dei guasti (rilevazione e riconfigurazione), ottimizzazione delle prestazioni (misura dei parametri di prestazione e modifica dei parametri di configurazione), gestione degli accessi (permessi per poter effettuare determinate operazioni nella rete). Queste operazioni devono poter essere effettuate da nodi remoti autorizzati.
Il protocollo SNMP è per reti basate su UDP, TCP, IP, attualmente il più usato in reti commerciali, basato sul modello cliente/servitore, si posiziona a livello applicazione. Il cliente è detto NETWORK MANAGEMENT STATION (NMS) è l'elemento che permette di effettuare agli utenti le operazioni di gestione. Il servitore, detto AGENTE, riceve richieste ed esegue operazioni impartitegli dai NMS. In generale vi sono molti agenti e pochi NMS.

Le basi dati per la gestione (MIB):
L'agente, deve conservare delle tabelle con i valori dei parametri utili per la gestione che devono poter essere letti e modificati dagli NMS che ne possiedono l'autorizzazione. La base dati, ha un formato standard, formata da 11 gruppi. Esiste la possibilità per un NMS di ricevere informazioni di gestione senza effettuare una richiesta utilizzando la TRAP. Ogni agente deve mantenere aggiornata la propria base dati con le informazioni sul proprio dominio. Se arrivano richieste da un NMS, è necessario verificare le autorizzazioni del mittente, eseguire l'operazione richiesta ed inviare il risultato al NMS richiedente. Ciascun NMS ha il compito di interrogare periodicamente gli agenti per mantenere uno stato aggiornato della rete, inviare i comandi ricevuti dall'utente e riportare eventuali problemi.

Formato dei MIB:
Sia MIB che MIB-2 sono suddivisi in 11 gruppi di variabili, le richieste raggiungono l'agente utilizzando l'indirizzo IP e la porta fornita dall'UDP. Per individuare sull'agente la risorsa necessaria si utilizza una serie di prefissi che distinguono anche il gruppo. Ogni gruppo del MIB ha un suo spazio di nomi separato. L'identificazione è possibile utilizzando la sintassi: <n.gruppo><n. oggetto> oppure: <nome gruppo><nome oggetto>. Ad ogni oggetto sono associate le informazioni sullo STATO, ACCESSO, VALORE.

Trasporto per SNMP:
Il trasporto delle informazioni viaggia utilizzando il protocollo UDP sulla porta 161 si ricevono tutti i messaggi tranne i trap che vengono ricevuti sulla porta 162. I messaggi sono di 484 bytes, 512-28 per header UDP e IP.

Comandi:
I comandi disponibili sono:

- GETREQUEST: legge una specifica variabile.
- GETNEXTREQUEST: legge una variabile non specificata.
- SETREQUEST: imposta un valore per una variabile.
- GETRESPONSE: invio di una risposta.
- TRAP: invio delle informazioni non sollecitate.

Richieste e risposte:
Poiché è possibile inviare varie richieste dallo stesso NMS allo stesso agente, è necessario distinguere all'arrivo le risposte per accoppiarle alle richieste pendenti. Sia le richieste che le risposte possono essere perse o duplicate, o arrivare fuori ordine. Il campo **request-id** contiene in partenza un numero unico fra tutte le richieste pendenti. Nella risposta, l'agente copia il **request-id** della richiesta nella PDU di risposta.

TELNET il terminale virtuale:

Fornisce un collegamento fra terminali a caratteri e "processi terminal-oriented", anche fra processi e processi. Si basa su una connessione TCP ed il protocollo è la realizzazione di un dispositivo virtuale chiamato Network Virtual Terminal, NVT.
Ciascuna implementazione mappa il NVT locale sul dispositivo reale e viceversa, per supportare terminali di tipo diverso, lo standard definisce una serie di opzioni negoziabili. La porta del processo o terminale server è 23.

Caratteristiche del NVT:
L'output da NVT è line-buffered, le informazioni di controllo (comandi) sono trasmesse con la sequenza del byte stuffing e sono indipendenti dalle caratteristiche del device reale.

Informazioni di controllo o comandi:
Vi è una realizzazione di alcuni segnali tipici dei terminali:

- **IP (Interrupt Process)**: richiesta da parte del terminale di interrompere il processo utente sull'host remoto.
- **AO (Abort Output)**: richiesta da parte del terminale di disabilitare l'output del processo utente sull'host remoto.
- **EC (Erase Character), EL (Erase Line)**: tipiche funzioni di editing della linea di commando.

In più vi è la negoziazione delle opzioni del NVT.

Segnalazioni urgenti:
Per segnalazioni urgenti, un NVT può inviare il segnale speciale SYNCH. Questo corrisponde alla trasmissione di un messaggio TCP urgente con il campo dati avente come ultimo carattere DM (Data Mark). Il ricevitore, scandisce l'input, scartando tutti i messaggi che non sono comandi, fino al raggiungimento del DM.

Opzioni TELNET:
Inizialmente, ogni NVT ha caratteristiche standard, tuttavia però, alcune di esse possono essere modificate tramite negoziazione. Alcuni esempi di opzioni sono ECHO, DIMENSIONE DELLA FINESTRA (numero di righe e colonne del display del NVT), LINEMODE (abilita a svolgere localmente l'editing della linea di comando, riducendo il traffico sulla linea).

Protocollo di negoziazione:
Se un NVT richiede di attivare un'opzione riferita a se stesso o all'altro, e se l'altro è d'accordo, invierà un ACK, così che l'opzione avrà effetto immediatamente dopo. Se l'altro NVT non è d'accordo manderà un ACK NEGATIVO, l'opzione non avrà effetto e si continuerà ad utilizzare il profilo standard. Invece la richiesta di disattivazione di un'opzione non può essere rifiutata. Non è possibile richiedere opzioni già aventi effetto e richiedere opzioni per cui è arrivato un ACK NEGATIVO.

Comandi per la negoziazione di opzioni:
I comandi per la negoziazione di opzioni sono:

- **WILL XXX**: si richiede di poter operare secondo l'opzione XXX (poter attivare l'opzione localmente).
- **DO XXX**: si richiede che l'altro operi secondo l'opzione XXX.
- **WONT XXX**: è l'ack negativo in risposta al DO.
- **DONT XXX**: è l'ack negativo in risposta ad un WILL.

La risposta positiva al WILL è DO, la risposta positiva al DO è WILL.
Per la negoziazione dei parametri delle opzioni si utilizza SB per l'inizio dell'elenco dei parametri e SE per la fine dell'elenco dei parametri. La sotto-negoziazione segue un protocollo specifico per ciascuna opzione.

FTP File Transfer Protocol:

L'FTP è un protocollo di trasferimento files che consente ad un utente di eseguire le seguenti operazioni:

- LOGIN (su un sistema remoto con identificazione ed autorizzazione)
- LISTING DEI DIRETTORI REMOTI
- COPIA AFFIDABILE DA FILE SYSTEM REMOTO A LOCALE E VICEVERSA
- ESECUZIONE DI ALCUNI SEMPLICI COMANDI SULLA MACCHINA REMOTA (cambio di direttorio corrente, rinomina file …)

Il modello client/server:
Gli elementi sono lo User Interface che si interfaccia con l'utente, il Client PI (Protocol Interpreter) che attraverso una control connection si scambia informazioni col Server PI su connessione telnet. Esistono poi le entità Server DTP (Data transfer Protocol) e Client DTP collegate da una data connection e basati sul file system remoto e locale.
La CONTROL CONNECTION si basa sul protocollo TELNET sulla porta 21 del server e la DATA CONNECTION è una connessione TCP che viene instaurata ogni volta che si deve trasferire un file.

I comandi e le risposte:
Sono delle sequenze di caratteri TELNET terminate da end-of-line TELNET. I comandi vengono emanati dallo user e hanno la seguente struttura:

codice (keyword) spazio **parametri**

Ad ogni comando segue la risposta con la struttura:

codice decimale spazio **testo corrispondente al codice**

I comandi per il controllo dell'accesso e della control connection:
- USER: specifica l'identità dell'utente
- PASS: specifica la pwd
- ACCT: specifica l'account
- SMNT: monta in remoto un file system
- QUIT: termina i trasferimenti e chiude la control connection
- REIN: come quit ma non chiude la connessione

I comandi per controllare i parametri di trasferimento:
- PORT: specifica la porta su cui il client farà open passiva per instaurare la data connection
- PASV: richiede che il server svolga un ruolo passivo
- TYPE: specifica tipo di dato
- STRU: specifica struttura file
- MODE: specifica il modo di trasferimento

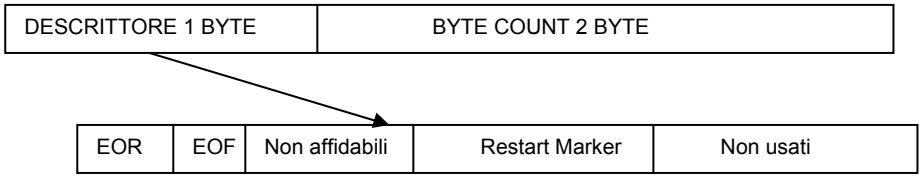
I comandi per il trasferimento file:
- RETR: trasferimento da server al client
- STOR: trasferimento da client a server
- STOU: come store ma crea un nome file unico sul server
- APPE: come store ma se il file esiste viene concatenato con quello nuovo
- REST: chiede al server di ripartire nel trasferimento dal marker

Convenzioni:
Per la CONTROL CONNECTION, il server attiva una open passiva sulla porta 21, il client attiverà una open attiva su quella porta del server.
Per la DATA CONNECTION per default, il server esegue una open attiva usando la porta 20 ed il client effettua una open passiva sulla stessa porta della control connection. Per modificare questi valori, che sono quelli di default, si utilizzando i seguenti comandi: PORT e PASV. La data connection viene aperta quando lo user invia un comando che implica il trasferimento dati. La data connection è chiusa normalmente dal server DTP ma può essere scritto alla fine di un file, o semplicemente se riceve un comando di ABORT, la modifica della porta o la chiusura della control connection.

Tipi di dato supportati da FTP:
I tipi supportati dal FTP sono ASCII, EBCDIC, IMAGE (a 1 byte per volta), LOCAL n (a n byte per volta).

Strutture di file supportate:
Le strutture di file supportate da FTP sono FILE SEQUENZIALE, FILE STRUTTURATO A RECORD, FILE STRUTTURATO A PAGINE (ad ogni pagina è associato un header standard).

Modi di trasmissione supportati:
I modi di trasmissione supportati sono STREAM, BLOCK, COMPRESSED.
Lo STREAM MODE è applicabile a file contenenti qualsiasi tipo di dato, con struttura sequenziale o a record, per i file strutturati a record gli EOR e EOF sono codificati usando 11111111.
Il BLOCK MODE:



Il COMPRESSED MODE :
Il flusso di dati viene suddiviso in sequenze, alcune delle quali trasmesse in forma compressa. La compressione viene effettuata inserendo informazioni di controllo analoghe a quelle dell'header del block mode, in testa alle sequenze.

Error recovery / restart:
E' previsto solo per i modi BLOCK e COMPRESSED, non riguarda perdita o alterazione dei dati trasferiti a cura del livello trasporto, solo a situazione di failure all'interno del sistema. E' basato sui RESTART MARKER, hanno significato solo per chi invia i dati che decide formati e scadenza, normalmente sono dei byte count. Non appena il marker viene ricevuto deve essere comunicato, in caso di failure lo user potrà emettere un comando di restart.

NAT – Network Address Translation

Tipologie, NAT e NAPT:

Nel NAT tradizionale o basic NAT, si traduce un gruppo di indirizzi in un altro senza che l'utente se ne accorga, nel NAPT si traduce un gruppo di indirizzi IP e porte TCP/UDP in un altro gruppo di indirizzi IP e porte TCP/UDP senza che l'utente se ne accorga.

Traduzione degli indirizzi nel Basic NAT:
I router dotati di NAT associano un indirizzo IP di una macchina interna ad uno degli indirizzi esterni disponibili. Se il numero degli indirizzi interni è uguale o inferiore a quelli esterni, esiste sempre la possibilità di accedere all'esterno. L'associazione tra i numeri può essere STATICA e DINAMICA. La stessa macchina può iniziare più sessioni simultaneamente usando lo stesso indirizzo.

Traduzione degli indirizzi nel NAPT:
Nel NAPT, l'associazione e la traduzione degli indirizzi avviene associando un indirizzo IP interno e una porta TCP/UDP con una stessa coppia esterna. Una volta stabilita questa associazione di coppie si ha la traduzione in entrambi i sensi.

Caratteristiche di Basic NAT e NAPT:
Nel Basic NAT non è possibile iniziare una sessione dall'esterno, è invece possibile con il NAPT, una volta conosciuto l'IP e la porta assegnata ad una macchina interna. Alcuni servizi come ICMP e DNS richiedono l'apertura di queste sessioni dall'esterno.

Routing e switch-over:
Il NAT non pubblicizza all'esterno la configurazione della rete, ma effettua il contrario, tanto che dall'interno, se un IP è disponibile, qualsiasi macchina collegata alla rete può navigare liberamente.
Lo SWITCH-OVER è un meccanismo che consente una volta finiti gli indirizzi IP con il Basic NAT di passare automaticamente al NAPT per collegare una o più macchine all'esterno, non è previsto dallo standard ma è realizzabile.

Possibili problemi:
SICUREZZA:
Il NAT garantisce una forma di protezione alle macchine interne contro attacchi esterni, tuttavia impedisce anche il monitoraggio di macchine che dall'esterno cercano di effettuare azioni dannose. Può anche rendere difficile il debugging.
FRAMMENTAZIONE:
L'invio di datagram frammentati su NAPT può fallire, perché solo il primo frammento contiene informazioni sulla porta.

NAT-PT (Protocol Translation):
E' la versione del NAT che effettua la traduzione dell'indirizzo da IPv.4 a IPv.6. Il NAT-TP tradizionale prevede sessioni iniziate solo dall'interno, mentre quello NAT-TP da entrambi i lati.

Reti con servizi integrati e VoIP

Il protocollo RSVP (Resource reSerVation Protocol):
RSVP è un protocollo utile a realizzare una prenotazione di risorse in servizi che richiedono che alcuni parametri del QoS rimangano entro certi limiti di qualità. Questi parametri sono la banda, il tempo di attraversamento, il jitter (variazione del ritardo), probabilità di perdita. Questo protocollo si interessa a comunicare i bisogni dei clienti alle varie apparecchiature di rete, quali routers … Permette l'utilizzo di ricevitori eterogenei, la costituzione di un gruppo, l'utilizzo di applicazioni con esigenze differenti, IP versione 4 o 6, è modulare da adattarsi alle reti sottostanti e l'overhead del protocollo cresce al massimo linearmente con l'aumento dei nodi.

Le operazioni effettuate da RSVP:
Inizialmente, prima che RSVP entri in gioco, i trasmettitori ed i ricevitori aderiscono ad un multicast group. Le operazioni che compie RSVP sono:

1. Segnalazione TRASMETTITORE-RETE: messaggi **path** che rendono il trasmettitore conosciuto al router, con il teardown si ha la distruzione del path nei router.
2. Segnalazione RICEVITORE-RETE: messaggio **reserve** che prenota le risorse dal trasmettitore al ricevitore, la reservation teardown si rimuove.
3. Segnalazione RETE-END SYSTEM: messaggi d'errore nel path o reserve.

I messaggi PATH:
I messaggi PATH contengono:

- **Indirizzo**: unicast o multicast.
- **Flowspec**: indicazioni sulle richieste circa banda, ritardi, jitter.
- **Filter flag**: se abilitato permette di ricevere l'identità dei vari riceventi per fare filtraggio alla sorgente.

- **Previous Hop**: indirizzo del router upstream (il primo in direzione del mittente).
- **Refresh Time**: tempo per cui queste informazioni sono valide.

Lo scopo dei messaggi PATH è quello di comunicare informazioni circa la sorgente e ricevere informazioni sui percorsi utilizzati. Permette ai routers di ricevere informazioni su come instradare le prenotazioni dei ricevitori.

I messaggi RESERVE:
Il contenuto dei messaggi RESERVE è:

- **Banda desiderata**
- **Filter type**: può essere: NO FILTER (tutti i pacchetti del gruppo possono usare questa prenotazione, FIXED FILTER (solo i mittenti specificati possono usare questa prenotazione), DYNAMIC FILTER (i mittenti abilitati cambiano nel tempo).
- **Filterspec**: indica i mittenti autorizzati.

Servono per prenotare le risorse per i messaggi che viaggiano in upstream verso i mittenti, aggiungono altre informazioni di stato nei router.

Nel caso che diversi mittenti vogliano prenotare una banda *b*, ed il loro traffico totale supera la banda si avrà un'alternanza casuale dei pacchetti e quando la banda non basterà più, i pacchetti verranno scartati.

Usi del rinfresco per PATH/RESERVE:
Possono servire a:

1. recuperare un path/riserve precedentemente perso, in questo caso il timeout che fa scomparire lo stato deve essere più lungo di quello fra 2 rinfreschi.
2. trattare il caso di mittenti o destinatari che scompaiono senza teardown, il timeout fa quindi scomparire lo stato nei routers e libera risorse non più usate.
3. causa l'aggiunta di nuove prenotazioni per un mittente che è apparso dopo l'ultimo refresh delle prenotazioni.

Il protocollo RTP (Real-Time transfer Protocol):

E' un procotollo che si occupa di trasferire i dati end to end con caratteristiche di real-time. Raccoglie una serie di meccanismi di base che possono essere utili a varie applicazioni. E' un protocollo flessibile perché fornisce meccanismi ma non algoritmi, neutro rispetto ai protocolli di livello inferiore, scalabile da unicast a O(10^7) utenti, controllo e dati sono separati e permette la cifratura dei messaggi.
Normalmente RTP è usato utilizzando UDP, per convenzione, il numero della porta del RTCP è maggiore di 1 rispetto a quella del RTP, di solito si trasmette solo un tipo di dato o audio o video fra ogni coppia di porte.

Le funzioni del RTP:
Il protocollo RTP possiede sia funzioni per i dati che per il controllo. Per i dati si hanno a disposizione funzioni per gestire il timing, rilevazione e perdita dati, etichettatura contenuti, Talkspurts e cifratura. Per il controllo si effettuano controlli periodici sul QoS, dei partecipanti alla sessione e alla rilevazione dei loop. Altre funzioni sono la risincronizzazione ed identificazione delle sorgenti.

Sistemi intermedi:
Esistono due tipologie di sistemi intermedi, i MIXER ed i TRANSLATOR. I MIXER fondono più flussi in ingresso in un unico flusso d'uscita, cambiano il formato dei dati, possono essere utilizzati per gestire la trasmissione su tratte con banda limitata, appaiono come

delle nuove sorgenti di dati con un proprio identificatore. Il TRANSLATOR agisce su un unico flusso dati, può anche convertirli, è utile quando bisogna effettuare una traduzione di protocolli (es ATM → IP).

Formato dei pacchetti di dato:
SSRC: sorgente di sincronizzazione, numero scelto a caso.
CSRC: contributing source, individua le sorgenti che contribuiscono ad una parte del flusso.
Sequence number: rivela la perdita dei dati
TIMESTAMP: istante di tempo in cui è stato raccolto il campione, incrementato di 1 per ogni nuovo campione, in caso di gap si assume che sia silenzio.

I pacchetti di controllo:
Simili a quelli di dato e spediti in gruppo, i campi più significativi sono:
SR: sender report, contiene informazioni sul n. di byte e pacchetti spediti, timestamp e tempo NTP.
RR: reception report, n. di pacchetti ricevuti ed attesi, jitter, ritardi.
I pacchetti RTCP servono a sincronizzare dati di flussi differenti, gli SR servono a correlare i tempi di diversi flussi, perché i timestamps sono differenti.

Banda riservata:
E' necessario prenotare la banda necessaria prima di usare RTP, per esempio con RSVP. La banda deve comprendere quella per i dati RTP e per l'RTCP, normalmente RTCP occupa il 5% e ripartita ¼ ai trasmettitori, e ¾ ai ricevitori. Il problema sussiste quando il numero di ricevitori e trasmettitori varia di molto nel corso della sessione.

Controllo del numero di partecipanti:
Ogni volta che si ricevono pacchetti da un nuovo trasmettitore, lo si aggiunge alla lista dei partecipanti in funzione del SSRC o CSRC visto nel pacchetto. Un trasmettitore viene eliminato dalla tabella quando invia un BYE, se per un certo numero di intervalli non si vedono arrivare pacchetti RTP o RTCP, il partecipante diventa inattivo, comunque contato nel calcolo della banda. Se un partecipante rimane disattivo per un periodo lungo, viene cancellato dalla tabella dei partecipanti.

Rilevazione e controllo delle collisioni:
Se due nuovi trasmettitori scelgono lo stesso SSRC si ha collisione, in questo caso è necessario inviare un BYE, uscire dalla sessione e rientrare scegliendo un nuovo SSRC. Ascoltando il traffico della sessione prima di entrare è possibile non collidere.

Per la rilevazione dei loop è necessario utilizzare un algoritmo che tenga presente dell'indirizzo di trasporto associato al SSRC.

Il protocollo SIP (Session Initiated Protocol):

E' il protocollo per la gestione di chiamate telefoniche su IP, esegue operazioni per l'inizio di una connessione, permette l'aggiunta di nuovi partecipanti ad una teleconferenza, si interfaccia con PSTN, gestisce la mobilità degli utenti, gestisce dinamicamente le preferenze degli utenti ed esegue altre funzioni PSTN come il follow-me.
Si appoggia solitamente ad UDP ma può anche utilizzare TCP.

Elementi dell'architettura (fisica):
SIP utilizza un'architettura client/server e i vari elementi sono:

- SIP User Agent, i telefoni
- SIP Servers, sono i proxy o redirect e vengono usati per localizzare i clienti, possono essere stateful o stateless
- SIP Gateways, interfacce verso PSTN e verso H.323

In generale, come di consueto nell'architettura client/server, i clienti generano le richieste ed i server rispondono o inoltrano i messaggi.

Le entità logiche:
Ciascuna entità fisica può essere composta da diverse entità logiche.
Gli User Agent si compongono di client e di server. I Network Servers si suddividono in REGISTRAR: accettano le segnalazioni dagli utenti circa il loro indirizzo effettivo o nuovo indirizzo, PROXY: decide verso quale altro dispositivo instradare la richiesta, REDIRECT: invia l'indirizzo del prossimo dispositivo al cliente.

Indirizzamento:
Per l'indirizzamento in SIP si utilizza lo stesso formato degli URI www, supporta sia indirizzi internet che PSTN (numeri telefonici). La forma generale è del tipo nome@dominio, e per completare la chiamata bisogna risolvere l'indirizzo fino ad ottenere utente@host.

Richieste:
Sono del tipo:

metodo spazio **uri** spazio **SIP-Version** CRLF

I metodi più diffusi sono INVITE, ACK (risposta positiva all'invite), OPTIONS, BYE, CANCEL, REGISTER.

Risposte SIP
Sono del tipo:

n. versione SIP spazio **codice stato** spazio **frase esplicita** CRLF

Corpo dei messaggi SIP:
Il più delle volte il corpo del messaggio può essere SDP, o serve a specificare informazioni riguardanti sessioni multi-mediali.

Autenticazione e cifratura:
SIP supporta una serie di approcci: cifratura end-to-end e cifratura tratto per tratto. I proxy possono richiedere autenticazione e anche gli utenti SIP possono richiederla.

Funzioni PSTN in SIP:
E' possibile ritrovare le funzioni disponibili presso un qualsiasi apparecchio telefonico anche nel SIP, per esempio risposta a chiamata, occupato, rifiuto collegamento, identificativo chiamata, messa in attesa, accodamento chiamate. Anche i Server hanno delle funzioni molto simili, per esempio la deviazione di chiamata, il follow-me, voicemail…

Il protocollo H.323:

E' un protocollo per teleconferenze multimediali su reti a pacchetto, inizialmente definito per le reti locali e successivamente esteso, si appoggia a TCP come livello di trasporto.

Architettura:

I componenti sono gli END POINTS che sono i terminali ed i Multipoint Control Unit (MCU). I GATEWAYS che servono all'interfacciamento con gli altri sistemi, i GATEKEEPER utilizzati per la gestione di una zona ed infine i BORDER ELEMENTS per la comunicazione fra più zone.

Gli End points contengono due categorie, i terminali e gli MCU. I terminali sono telefoni, videotelefoni, voicemail, soft-phone. Mentre gli MCU, ovvero i Multipoint Control Unit, sono utilizzati per gestire una conferenza con più di due terminali. Contengono un Multipoint Controller (MC) che garantisce tutta la segnalazione relativa alla conferenza. Può contenere un Multipoint Processor (MP), che effettua la commutazione dei vari media.

I Gateways interfacciano l'H.323 con gli altri sistemi come PSTN. Può essere utilizzato fra due sistemi H.323 (proxy), per nascondere i dettagli di indirizzamento, attraversare un firewall o altro.

I Gatekeeper, sono necessari nel caso si debba attraversare un gateway. Hanno funzioni di traduzione indirizzi ed ammissione e autorizzazione di nuove chiamate. Oltre alle chiamate normali, può realizzare funzioni tipo follow-me o deviazione se occupato. Può effettuare un semplice controllo sulla banda occupata. Un gatekeeper è associato ad una zona composta da tutti quegli elementi che si sono registrati con il gatekeeper.

Il Border element si trova molto spesso nella stessa macchina del gatekeeper e partecipa allo scambio di informazioni sugli indirizzi e alle procedure di ammissione per le chiamate fra diversi domini. Possono aggregare informazioni sugli indirizzi, in modo da semplificare il routing. Possono effettuare le operazioni relative alle chiamate fra domini o direttamente oppure con l'uso di una **clearinghouse**.

Registration Admission and Status (RAS):

Permette ad un endpoint di richiedere l'autorizzazione a iniziare o a rispondere ad una chiamata. Permette ad un gatekeeper di controllare le chiamate in ingresso e in uscita dalla propria zona. Permette ad un gatekeeper di comunicare gli indirizzi di altri endpoint.

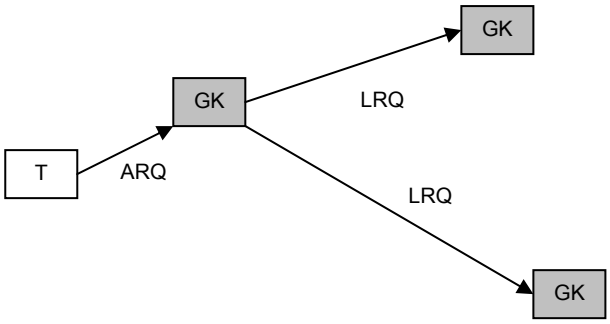
Risoluzione degli indirizzi:

Un gatekeeper può risolvere un indirizzo di endpoint in vari modi:

- Inviando Location Request (LRQ) ad altri gatekeeper
- Chiedendo ad un BE
- Interrogando un database

I border element possono chiedere ad altri border elements e scambiare indirizzi, anche indipendentemente dalle richieste di chiamata.

Risoluzione con Location Request (LRQ):
Il gatekeeper può inviare dei LRQ ad uno o più gatekeeper, può accetare una risposta LCF ed utilizzarla per rispondere alla richiesta ARQ.



Uso dei Gatekeeper gerarchici:
Il gatekeeper può girare un LRQ ricevuto ad un altro GK, allo scopo di risolvere l'indirizzo. La risposta può essere inviata direttamente al GK originario o a quello da cui si è ricevuto la richiesta.

Uso di un border element:
I Border element possono inviare la richiesta al di fuori del dominio amministrativo con AccessRequest, indicando dove inviare la risposta. Un border element può anche rispondere ad una richiesta anche utilizzando informazioni finite nella cache e relative a scambi precedenti.

Complementi su TCP:

Il controllo di flusso, ovvero la finestra di trasmissione:

Oltre al controllo di congestione, esiste il controllo di flusso determinato esclusivamente dal ricevitore. Il ricevitore, infatti con il meccanismo dei crediti impone al trasmettitore una finestra di trasmissione. Il meccanismo è gestito con gli ACK, che insieme alla conferma dell'avvenuta ricezione invia anche la dimensione della finestra di trasmissione.

Complementi sui socket in ambito UNIX:

```
struct in_addr {
unsigned long s_addr;
}
struct sockaddr_in {
short sin_family;
u_short sin_port;
struct in_addr sin_addr;
char sin_zero[8];
};
struct sockaddr_in saddr;
saddr.sin_family = AF_INET;
saddr.sin_port = 80;
saddr.sin_addr.s_addr = INADDR_ANY;
result = bind(s, (struct sockaddr *) &saddr, sizeof(saddr));
```

Creazione di un socket tipo:

```
int socket (int family, int type, int protocol)
• family dominio del socket
• type tipo di socket
• protocol protocollo utilizzato dal socket
```

Assegnazione di un indirizzo locale di rete ad un socket:

```
int bind (int socket, struct sockaddr *addr,
int addrlen)
• socket socket cui assegnare l'indirizzo
• addr puntatore all'indirizzo che si vuole assegnare
• addrlen lunghezza della struttura sockaddr utilizzata
(sockaddr varia a seconda della address family)
```

Socket di tipo SOCK_STREAM

Predisposizione del socket per ricevere richieste di connessione (server):

```
int listen (int socket, int backlog)
• socket socket al quale mettersi in attesa
• backlog lunghezza massima della coda delle
richieste pendenti
```

Richiesta di connessione (client):

```
int connect (int socket, struct sockaddr
*destaddr, int addrlen)
• socket socket che si vuole connettere
• destaddr puntatore all'indirizzo del server remoto
cui si vuole indirizzare la richiesta
• addrlen lunghezza dell'indirizzo
```

Accettazione di una richiesta di connessione (server):

```
int accept (int socket, struct sockaddr
*srcaddr, int *addrlen)
• socket socket dove si riceve la richiesta
• srcname puntatore all'indirizzo del socket remoto
con cui viene stabilita la connessione
```

• namelen puntatore alla lunghezza dell'indirizzo

Invio dati su una connessione:

```
int send (int socket, char *data, int datalen,
int flags)
• socket socket connesso attraverso cui si inviano
i dati
• data buffer contenente i dati da inviare
• datalen lunghezza del blocco di dati da inviare
• flags specifica eventuali opzioni, come per
esempio "out-of-band data"
```

Ricezione dati su una connessione:

```
int recv (int socket, char *buffer, int
buflen, int flags)
• socket socket connesso attraverso cui si ricevono
i dati
• buffer buffer di ricezione
• buflen lunghezza del buffer di ricezione
• flags specifica eventuali opzioni, come per
esempio "out-of-band data"
```

Invio/ricezione di datagram su socket di tipo SOCK_DGRAM:

Invio di datagram:

```
int sendto (int socket, char *data,
int datalen, int flags,
struct sockaddr *addr, int addrlen)
• socket socket attraverso cui inviare il datagram
•data buffer contenente il datagram da inviare
•datalen lunghezza datagram da inviare
•flags eventuali opzioni
•addr puntatore all'indirizzo del destinatario
•addrlen lunghezza dell'indirizzo
```

Ricezione di datagram:

```
int recvfrom (int socket, char *buffer,
int buflen, int flags,
struct sockaddr *addr, int *addrlen)
• socket socket attraverso cui ricevere il datagram
•buffer buffer in cui verrà depositato il datagram ricevuto
•datalen lunghezza del buffer
•flags eventuali opzioni
•addr puntatore indirizzo del mittente
•addrlen puntatore lunghezza dell'indirizzo del mittente
```

ESEMPI DI DOMANDE D'ESAME

- Descrivere il controllo di flusso in TCP
- Descrivere sequenza di pacchetti scambiati in TCP (riportare anche disegni o schemi esplicativi)
- Elencare metodi http più usati, ossia Get, PostHead, Options…etc e specificare cosa fanno
- Quali sono le informazioni necessarie per chiamare una procedura remota, ossia n° versione, n° porta, IP…etc
- Come funziona il DNS? (Traduzione indirizzo)
- Come funziona ARP
- Come funziona RIP
- Come funziona OSPF
- Pezzo di codice in C e spiegare cosa fa (socket e simili).Può anche essere richiesto di individuare un eventuale errore. Non è richiesta la programmazione in C ma solo un commento a un pezzo di programma già svolto.
- Come funzionano e si propagano i Newsgroups
- Elencare i modi di trasferimento in FTP
- Descrivere il protocollo FTP in generale (Control connection, Data connection…etc)
- Descrivere l'FTP Start Marker
- Descrivere la struttura del file FTP
- Descrivere Telnet e il suo funzionamento
- Descrivere come funziona il protocollo "x" (dove x è uno fra tutti i protocolli affrontati)