EUCIP IT Administrator
Modulo 4 - Uso Esperto della Rete

Livello Trasporto

Alberto BOSIO

Sommario

- Introduzione
- Protocolli
- Sicurezza del livello 4

Funzione del livello di trasporto

- Il livello di trasporto ha lo scopo di fornire allo strato superiore un servizio di trasferimento dei dati *end to end*, mascherando completamente al livello superiore il fatto che tra i due host terminali esista una rete di qualsiasi tipo, topologia, tecnologia e complessita'
  - per OSI lo strato superiore e' il livello di sessione
  - per TCP/IP lo strato superiore e' il livello di applicazione
- Per assolvere le sue funzioni lo strato di trasporto utilizza i servizi dello strato di rete

Necessita' dello strato di trasporto

- Perche' introdurre un nuovo strato?
  - lo strato di rete opera su tutte le macchine attraversate dai dati, indipendentemente
  - in mancanza di uno strato che operi esclusivamente sui nodi terminali della comunicazione l'utente della rete non puo' tenere sotto controllo cosa accade ai dati una volta che lasciano l'host sorgente
  - lo strato di trasporto rende trasparente allo strato superiore la complessita' (l'esistenza!) della sottorete
- La presenza di uno strato intermedio tra applicazione e rete puo' offrire un meccanismo per rendere affidabile una comunicazione su sottoreti inaffidabili
- Inoltre offre al suo utente una interfaccia indipendente dalle diverse tecnologtie dello strato di rete

Servizi forniti allo strato superiore

- Lo strato di trasporto offre
  - servizio affidabile orientato alla connessione
  - servizio inaffidabile non orientato alla connessione
- In modo analogo ai servizi equivalenti dello strato di rete:
  - il servizio orientato alla connessione realizza un trasferimento dei dati affidabile (controllo della integrita', della completezza, dell'ordine) e permette di gestire controllo di flusso; i dati vengono trasferiti con un procedimento in tre fasi
    - si stabilisce la connessione
    - si inviano i dati attraverso la connessione
    - si chiude la connessione
  - il servizio non orientato alla connessione fornisce un meccanismo di trasferimento dati "best-effort": ogni blocco di dati viene inviato e ci si dimentica di lui; se arrivano, bene, se no l'applicazione dovra' farsi carico di operare azioni correttive (se necessarie)

Primitive di servizio

- Le primitive di servizio costituiscono l'interfaccia che lo strato di trasporto rende disponibile allo strato superiore
- Per il servizio connection oriented si possono elencare in
  - **LISTEN**: lo strato superiore notifica al trasporto che e' pronto a ricevere una connessione
  - **CONNECT**: lo strato superiore chiede allo strato di trasporto di effettuare una connessione (si traduce nell'invio da parte del trasporto di un messaggio "Connection Request" al destinatario)
  - **SEND**: lo strato superiore chiede al trasporto di inviare dati
  - **RECEIVE**: lo strato superiore chiede allo strato di trasporto di trasmettergli i dati in arrivo
  - **DISCONNECT**: lo strato superiore chiede di chiudere la connessione (si traduce nell'invio da parte dello strato di trasporto di un messaggio "Disconnection Request")
- Per il servizio connection less, le due primitive SEND e RECEIVE possono essere sufficienti

Protocollo di trasporto

- Il livello di trasporto realizza le sue funzioni comunicando con il processo paritario secondo un protocollo definito
- Le informazioni vengono scambiate tramite la trasmissione di blocchi di dati
  - in OSI si chiamano TPDU (Transport Protocol Data Unit)
  - in TCP/IP si chiamano segmenti (ma spesso anche pacchetti)
- Il protocollo si dovra' occupare dei meccanismi per gestire i diversi eventi (ove necessari):
  - come si stabilisce la connessione
  - come si chiude una connessione
  - controllo di flusso
  - controllo degli errori, ritrasmissioni e acknowledge
  - ordinamento in sequenza dei dati
- Le problematiche sono simili a quelle del livello di data link

Differenze rispetto al data link

- Il fatto che a livello di trasporto i due terminali siano separati da una rete provoca complicazioni
  - e' necessario un indirizzamento per gestire diverse applicazioni utenti del servizio di trasporto
  - a livello 2 un pacchetto o arriva o non arriva, mentre a livello 4 la sottorete provoca ritardi e riapparizione di pacchetti che si credevano perduti e quindi ritrasmessi, con conseguente duplicazione
  - il numero delle connessioni cui il livello di trasporto deve far fronte e' molto piu' elevato che nel caso del data link layer: non sara' possibile dedicare a tutti i buffer necessari alle comunicazioni

Trasporto in TCP/IP

- TCP/IP utilizza due protocolli di trasporto
  - UDP (User Datagram Protocol): protocollo inaffidabile connection less
  - TCP (Transmission Control Protocol): protocollo connection oriented affidabile
- Entrambi i protocolli forniscono una interfaccia agli applicativi per la trasmissione dei dati, ed utilizzano IP per il trasporto dei dati (e, nel caso di TCP, delle informazioni di controllo del protocollo)

Trasporto in TCP/IP

- Esiste una interfaccia di programmazione, chiamata socket, standardizzata per il linguaggio C
  - esistono implementazioni di interfacce al socket anche per altri linguaggi (ad esempio per il perl)
- Questo rende possibile scrivere applicazioni specifiche (home-made) che debbano far uso della rete di trasmissione dati in aggiunta agli applicativi standardizzati esistenti (generalmente forniti con il sistema operativo)

Indirizzamento del trasporto in TCP/IP

- Le applicazioni che utilizzano il TCP/IP si registrano sullo strato di trasporto ad un indirizzo specifico, detto porta
- La porta e' il meccanismo che ha a disposizione una applicazione per identificare l'applicazione remota a cui inviare i dati
- La porta e' un numero di 16 bit (da 1 a 65535; la porta 0 non e' utilizzata)

Indirizzamento del trasporto in TCP/IP

- TCP/IP permette alla applicazione di registrarsi su una porta definita (nel caso dei server) o su una qualunque porta libera scelta dal livello di trasporto (spesso e' il caso dei client)
- Per rendere funzionali i servizi di utilizzo diffuso, TCP/IP prevede che determinati servizio utilizzino dal lato server delle porte ben definite
  - il valore dei numeri di porta vengono definiti negli RFC che definiscono il protocollo delle applicazioni in questione

Indirizzamento del trasporto in TCP/IP

- Esiste una autorita' centrale, lo IANA (Internet Assigned Numbers Authority), che pubblica la raccolta dei numeri di porta assegnati alle applicazioni negli RFC (http://www.iana.org)
  - non solo: lo IANA centralizza la gestione anche di altro, come le assegnazioni dei numeri di protocollo dei diversi protocolli di trasporto utilizzati nel protocol number di IP o l'assegnazione dei domini di primo livello del DNS

User Datagram Protocol

- UDP implementa un servizio di consegna inaffidabile dei dati a destinazione
- UDP riceve i dati dalla applicazione e vi aggiunge un header di 8 byte, costruendo cosi' il segmento da inviare
- L'applicazione specifica (l'indirizzo di rete e) la porta di destinazione, ed in ricezione UDP recapita il campo dati al destinatario

User Datagram Protocol

- UDP non si preoccupa di sapere nulla sul destino del segmento inviato, ne' comunica alla applicazione qualsiasi informazione
- Di fatto costituisce semplicemente una interfaccia ad IP (che fornisce lo stesso tipo di servizio), con l'aggiunta di fare multiplexing del traffico delle applicazioni su IP
  - tramite il meccanismo delle porte a cui sono associate le applicazioni, di fatto UDP realizza un multiplexing dei dati delle diverse applicazioni su IP

Orientato al datagramma

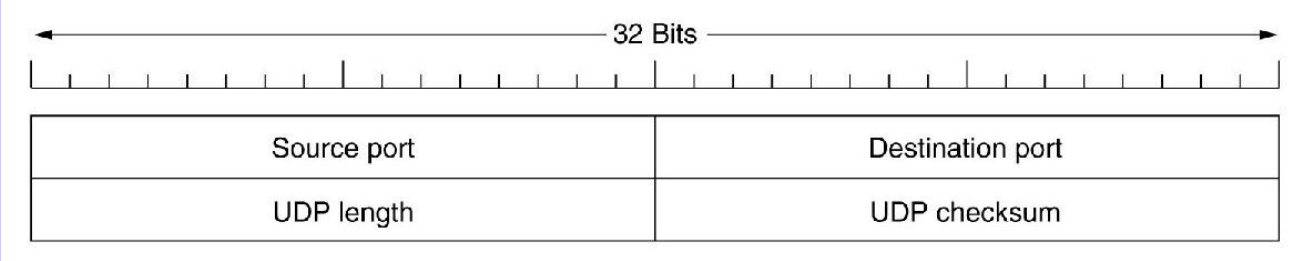
- A differenza di TCP, UDP si occupa di un datagramma per volta
  - quando una applicazione passa dati ad UDP, UDP li maneggia in un unico segmento, senza suddividerlo in pezzi
  - il segmento di massime dimensioni che UDP puo' gestire deve stare interamente nel campo dati del pacchetto IP
  - il segmento viene passato ad IP che eventualmente lo frammenta, ma a destinazione UDP ricevera' il datagramma intero
  - l'applicazione di destinazione ricevera' quindi il blocco completo di dati inviato dalla applicazione che li ha trasmessi

Il segmento UDP

- Il segmento UDP e' costituito da un header di lunghezza fissata (8 byte) piu' il campo dati, che deve avere dimensione massima tale da stare dentro il campo dati di IP
  - poiche' il pacchetto IP puo' essere lungo 65535 byte, il campo dati UDP puo' essere lungo al massimo (65535 – 8 – lunghezza header IP) byte

UDP header

- L'header e' costituito da quattro campi di due byte:
  - source e destination port: le porte di associazione alle applicazioni mittente e destinataria dei dati
  - length: lunghezza del segmento in byte (compreso l'header)
  - checksum: questo campo contiene una checksum del segmento completo (anzi: viene aggiunto uno pseudo-header con le informazioni degli indirizzi IP di sorgente e destinazione)
    - l'utilizzo del campo checksum e' opzionale, e l'applicativo puo' decidere di non utilizzarlo (in questo caso il campo e' riempito con zeri)
    - molti applicativi non lo utilizzano per motivi di efficienza
    - se viene utilizzato, un errore provoca la rimozione del segmento senza che vengano prese altre iniziative



Caratteristiche di UDP

- Benche' inaffidabile, UDP ha caratteristiche che per molte applicazioni sono appetibili
  - puo' utilizzare trasmissione multicast o broadcast
    - TCP e' un protocollo orientato alla connessione, quindi per definizione non puo' gestire una comunicazione tra piu' di due entita'
  - e' molto leggero, quindi efficiente
    - la dimensione ridotta dell'header impone un overhead minimo, ed una rapidita' di elaborazione elevata
    - la mancanza di meccanismi di controllo rende ancora piu' rapida l'elaborazione del segmento ed il recapito dei dati

Applicativi che utilizzano UDP

- Applicativi che necessitano di trasmissioni broadcast
- Applicativi per i quali la perdita di una porzione di dati non e' essenziale, ma richiedono un protocollo rapido e leggero
  - stream voce/video
- Applicativi che si scambiano messaggi (e non flussi di byte) di piccole dimensioni, e che non risentono della perdita di un messaggio
  - interrogazione di database
  - sincronizzazione oraria
  - in questi casi la perdita della richiesta e della risposta provoca un nuovo tentativo di interrogazione

Applicativi che utilizzano UDP

- Applicativi che necessitano di risparmiare l'overhead temporale provocato dalla connessione, ed implementano a livello di applicazione il controllo della correttezza dei dati
  - ad esempio applicativi che scambiano dati con molti host, rapidamente, per i quali dover stabilire ogni volta una connessione e' peggio che ritentare se qualcosa va storto

Applicativi standard su UDP

- Sono molti, ed in aumento
- Gli applicativi che storicamente utilizzano UDP sono
  - **DNS**, sulla porta 53
  - **TFTP** (Trivial File Transfer Protocol), sulla porta 69
  - **NetBIOS** Name Service (anche WINS) sulla porta 137
  - **SNMP** (Simple Network Management Protocol) sulla porta 161
  - **NTP** (Network Time Protocol) sulla porta 123
  - **NFS** (Network File System) via portmap

Transmission Control Protocol

- TCP e' stato progettato per offrire un flusso di byte affidabile orientato alla connessione
- TCP offre i seguenti servizi allo strato applicativo:
  - protocollo orientato alla connessione
  - affidabilita' dei dati (controllo, ritrasmissione, ordinamento)
  - gestione del controllo di flusso
  - gestione della congestione

Trasmissione dei dati in TCP

- TCP trasmette dati in segmenti, ciascuno costituito da un header ed un campo dati
- TCP considera i dati da trasmettere come flusso di byte (a differenza di UDP che opera in termini di messaggi)
- TCP utilizza buffer in trasmissione e ricezione per la gestione dei dati
  - TCP non invia necessariamente i dati appena li riceve dalla applicazione: per motivi di efficienza puo' tenere nei buffer i dati da inviare fino a che non ce ne siano abbastanza per evitare messaggi troppo piccoli
- L'informazione sul numero di sequenza e' quindi riferito al byte trasmesso, ed e' utilizzato sia per l'acknowledge che per il riordinamento e la ritrasmissione

Dimensione del segmento TCP

- Il segmento TCP e' costituito da un header di 20 byte (piu' campi opzionali, come in IP) seguito dal campo dati
- La dimensione massima del segmento TCP deve stare nel campo dati di un pacchetto IP
  - poiche' il pacchetto IP ha lunghezza massima 65535 byte, con un header di 20 byte, il campo dati di TCP avra' valore massimo 65495 byte (ma in caso di utilizzo di intestazione estesa sara' meno)

Connessione TCP

- TCP utilizza per la connessione il meccanismo di handshake a 3 vie
  - un segmento (SYN) viene inviato dal client al server; questo trasporta il sequence number iniziale del client, e le informazioni di porta sorgente e destinazione
  - un segmento (SYN+ACK) viene inviato in risposta dal server; questo trasporta l'acknowledge del SYN precedente, ed il sequence number iniziale el server, per le comunicazioni in verso opposto
    - se nessuno ascolta sulla porta di destinazione, il server inviera' un segmento RST (Reset) per rifiutare la connessione
  - un segmento di ACK viene inviato dal client al server; questo riporta lo stesso sequence number iniziale (non sono ancora stati trasmessi dati) e l'acknowledge del secondo segmento SYN

Connessione TCP

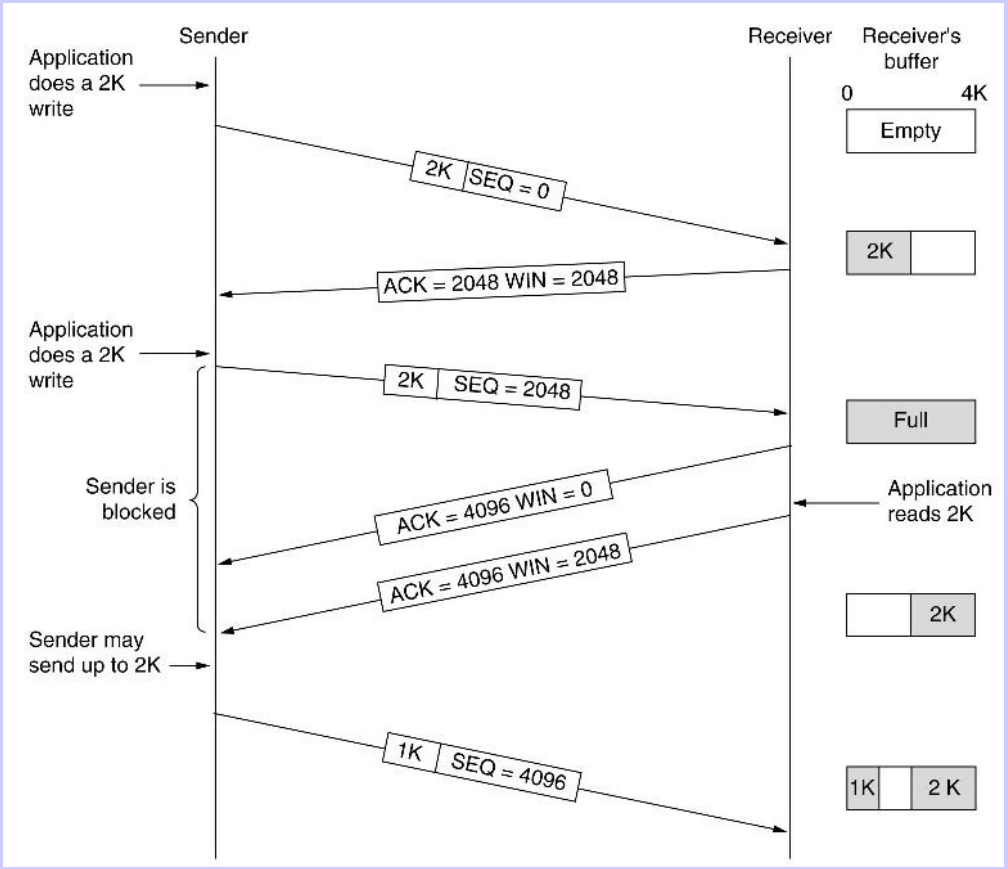
- A questo punto la connessione viene considerata stabilita (la connessione e' definita dalla quaterna host1-port1-host2-port2)
- I messaggi di SYN possono opzionalmente trasportare le informazioni di MTU/MRU per determinare il MSS della connessione

Disconnessione TCP

- La connessione TCP e' full duplex
- Per la disconnessione, si utilizza un handshake a due vie per ogni direzione:
  - chi vuole disconnettere invia un segmento FIN
  - l'altro invia un ACK del FIN: il primo considera chiusa la comunicazione in quel verso, ma non nel verso opposto
  - la stessa cosa fa il secondo quando non ha piu' dati da trasmettere, ed aspetta il relativo ACK
  - il tutto spesso viene fatto con tre segmenti, inviando il secondo FIN assieme all'ACK del primo
- Vengono utilizzati dei timer per aggirare il problema dei due eserciti

Controllo di flusso

- Il controllo di flusso viene realizzato tramite la comunicazione dello spazio nei buffer disponibile in ricezione
  - lo spazio viene comunicato sempre in termini di byte liberi nei buffer
- Come gia' visto, il meccanismo permette al ricevente di regolare la trasmissione del trasmittente, in modo disgiunto dagli acknowledge
- La "dimensione della finestra" (quanti segmenti si possono inviare) e' data dal rapporto tra i byte a disposizione ed il valore del MSS



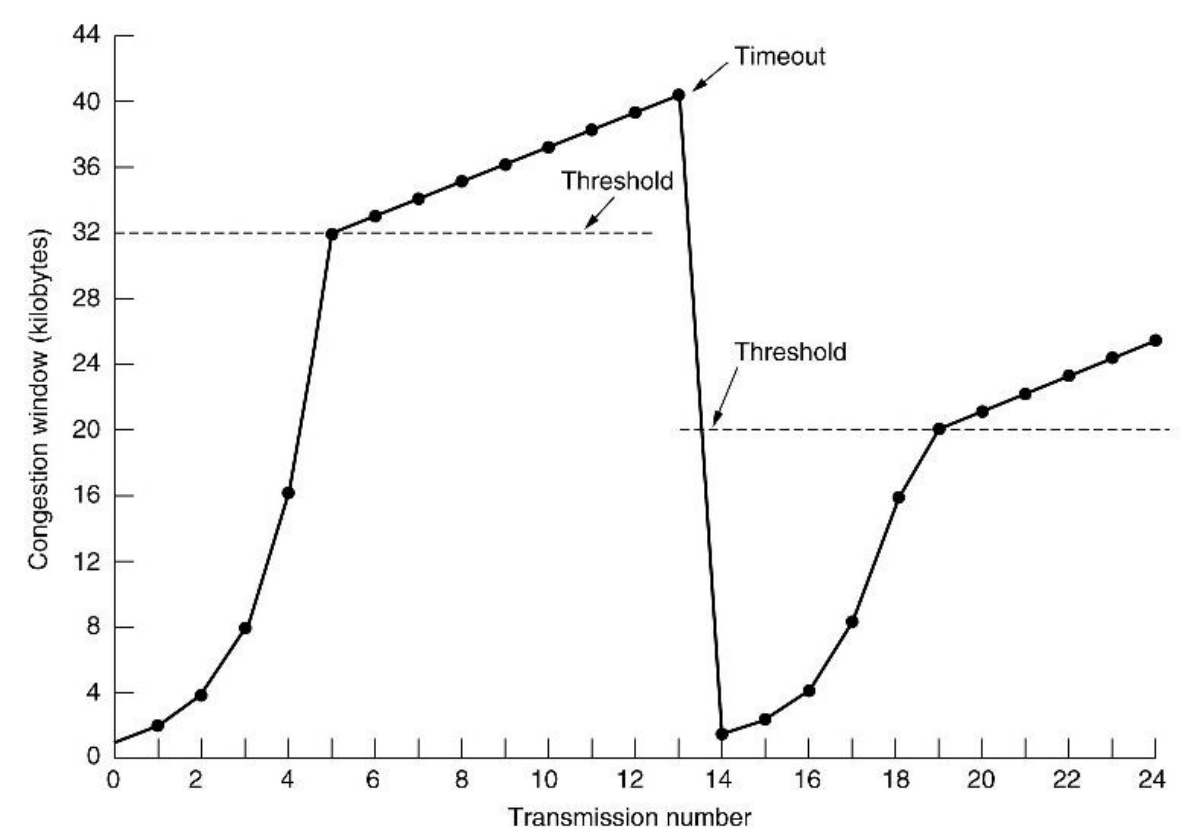
Controllo della congestione

- Accanto alla finestra di ricezione (legata ai buffer) viene utilizzata una finestra di congestione: il limite a cui si puo' trasmettere e' il minimo tra la finestra di ricezione e quella di congestione
- Per prevenire le congestioni, il TCP utlizza una tecnica detta "slow start":
  - inizialmente il trasmittente inizializza la finestra di congestione al valore del MSS, ed invia un segmento
  - se il segmento riceve l'ACK, raddoppia la finestra di congestione, e via cosi' fino al massimo valore determinato dalla finestra di ricezione, o fino a che si incontra un timeout; questo valore viene quindi mantenuto per la comunicazione
  - in base alla insorgenza di timeout o di ack duplicati (o di ICMP source quench), il trasmittente puo' valutare l'insorgere di una congestione
  - quando questo avviene, la finestra di congestione viene ridotta ad un MSS, ed una soglia viene impostata alla meta' del valore precedente (il limite raggiunto dallo startup lento)
  - riprende la progressione iniziale, ma solo fino al valore di soglia, oltre il quale si incrementa la dimensione di un MSS per volta

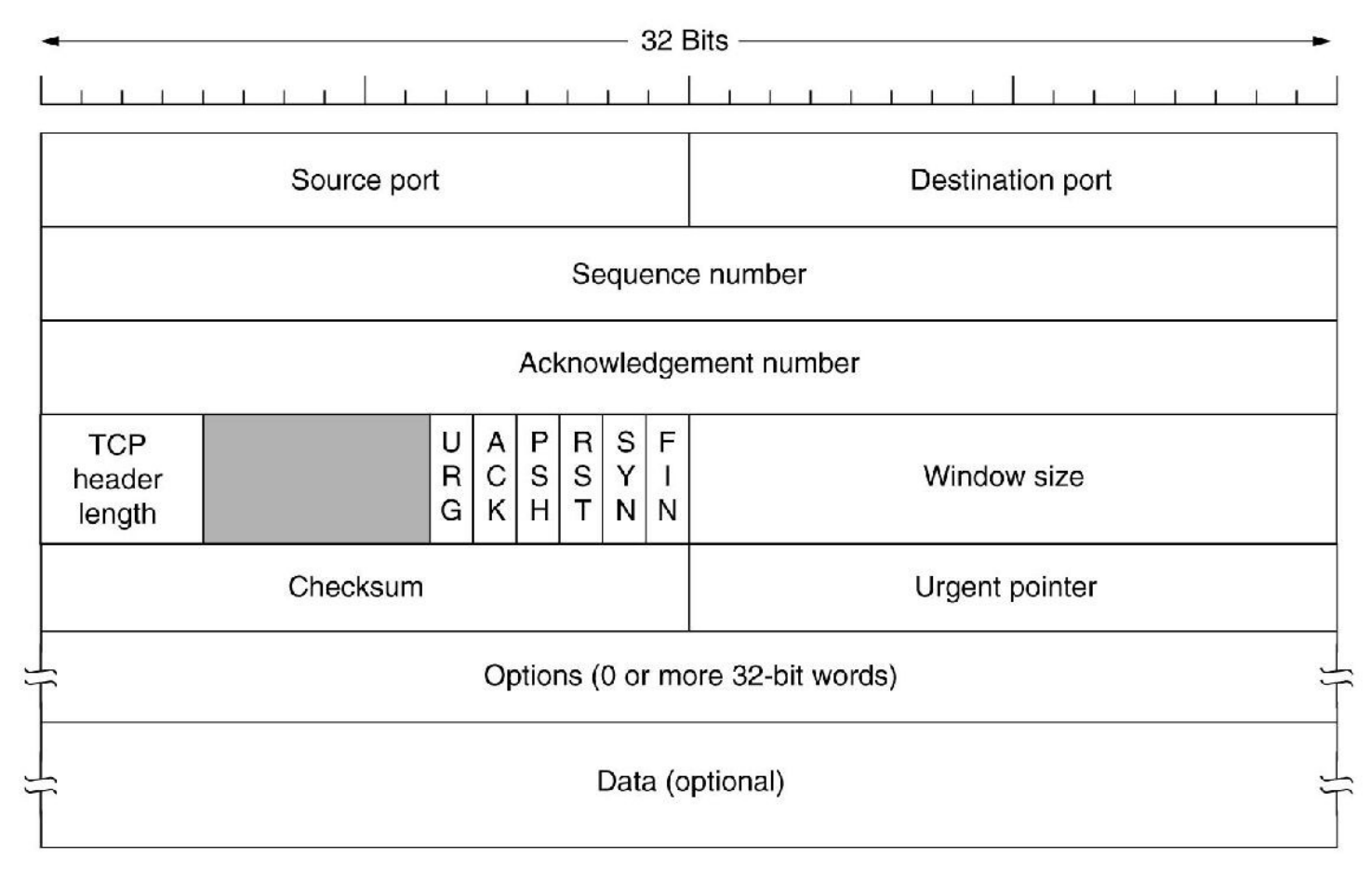
Controllo della congestione

- In questo modo il TCP tenta di prevenire la congestione (con l'avvio lento), ed abbatte immediatamente la trasmissione di segmenti quando la congestione inizia a presentarsi, risale rapidamente fino ad un certo valore per non perdere in efficienza e poi piu' lentamente per non ricreare le condizioni di congestione

Controllo della congestione



Header TCP


32 Bits
Source port
Destination port
Sequence number
Acknowledgement number
TCP header length
URG
ACK
PSH
RST
SYN
FIN
Window size
Checksum
Urgent pointer
Options (0 or more 32-bit words)
Data (optional)

Header TCP (cont.)

- source e destination port
  - le porte del sorgente e del destinatario, che permettono di identificare le applicazioni a cui sono destinati i dati (16 bit ciascuna)
- sequence number (32 bit)
  - il valore del primo byte trasmesso nel segmento; all'atto della connessione viene stabilito il valore iniziale, basato sul clock del trasmittente
- acknowledge number (32 bit)
  - il valore dell'ultimo byte riscontrato piu' uno (cioe' del successivo atteso)
- TCP header length (4 bit)
  - il numero di gruppi di 32 bit contenuti nella intestazione; necessario perche' sono previsti campi opzionali (non piu' di 60 byte)
- flag URG (urgent)
  - il campo dati contiene dati urgenti, che devono essere passati alla applicazione prima degli altri ancora in attesa nei buffer (ad esempio: il CTRL^C in applicazioni di terminale remoto)

Header TCP (cont.)

- flag ACK
  - il segmento trasporta un riscontro; tutti i segmenti tranne il primo dovrebbero averlo settato
- flag PSH (push)
  - indica che l'applicativo ha richiesto l'invio dei dati senza ulteriore attesa (ed in ricezione deve essere fatto lo stesso)
- flag RST (reset)
  - utilizzato per comunicare che la connessione deve essere abortita, o quando viene rifiutata una nuova connessione
- flag SYN (synchronize)
  - utilizzato per stabilire una connessione; questi segmenti definiscono il sequence number iniziale per i due versi
- flag FIN (finish)
  - utilizzato per comunicare alla controparte che non si hanno piu' dati da inviare e che si desidera chiudere la connessione; il doppio FIN con relativo riscontro genera il rilascio della connessione

Header TCP (cont.)

- window size (16 bit)
  - la dimensione in byte dello spazio disponibile dei buffer in ricezione: il valore massimo e' di 64 KB
  - le reti moderne molto veloci rendono questo limite inefficiente: e' possibile utilizzare un header opzionale per accordarsi su una window size a 30 bit (buffer fino ad 1 GB)
- checksum (16 bit)
  - obbligatoria per TCP (al contrario di UDP); anche in TCP la checksum viene calcolata su tutto il segmento piu' uno pseudo header che riporta gli indirizzi IP di sorgente e destinazione
- urgent pointer (16 bit)
  - definisce l'offset dell'ultimo byte facente parte dei dati urgenti quando la flag URG e' settata

Header opzionali

- Le opzioni sono definite da una lunghezza, un tipo, ed i dati relativi; sono definite diverse opzioni, tra cui:
  - padding: necessario in presenza di opzioni per rendere il campo header nel suo complesso un multiplo di 32 bit
  - MSS: utilizzato con i segmenti SYN per determinare il MSS scambiandosi i valori di MTU ed MRU
  - window scale: utilizzata per definire la dimensione della finestra fino a 30 bit
  - selective acknowledge: TCP utilizza normalmente il go-back-N; questa opzione permette di utilizzare il selective reject
  - timestamp: utilizzata per valutare (a livello di trasporto) il round trip time e poter definire valori opportuni per i timer interni

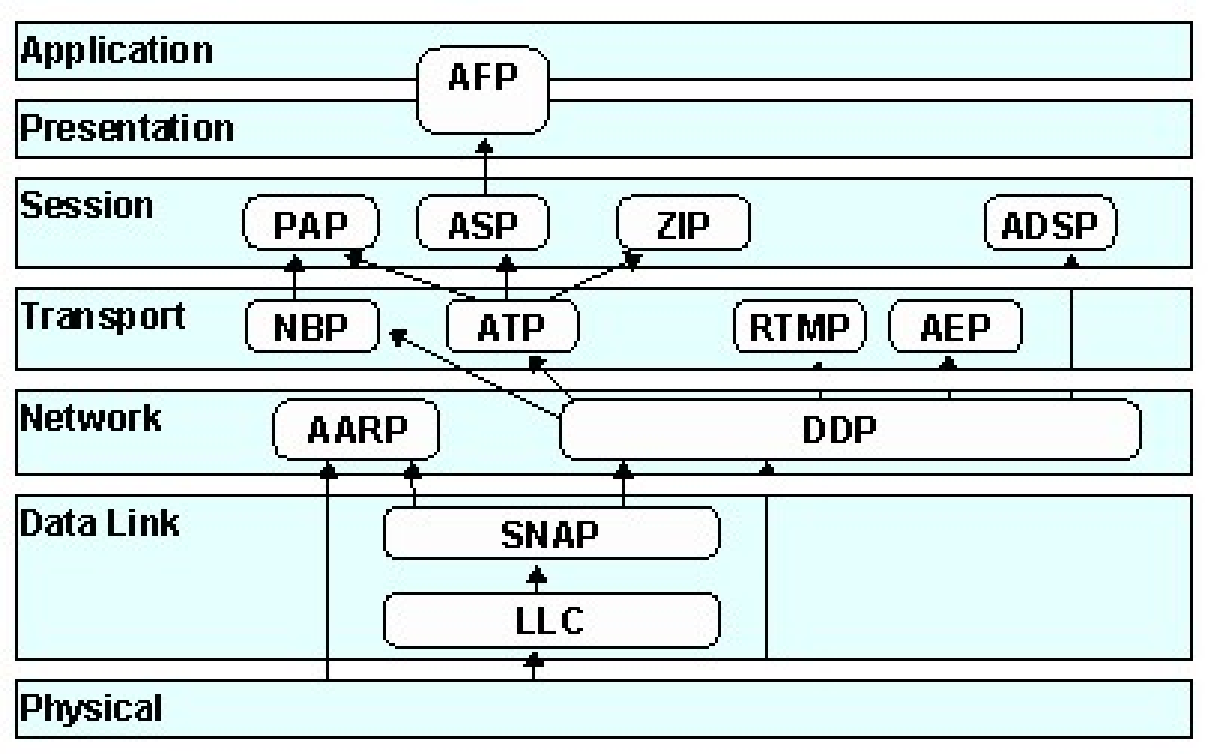
Applicazioni che usano TCP

- Tutte quelle che richiedono affidabilita' dei dati, e che non hanno bisogno della comunicazione multicast/broadcast
  - la comunicazione in TCP e' orientata alla connessione tra due punti terminali; non puo' quindi supportare comunicazione multicast
- Esistono tantissime applicazioni; tra le piu' diffuse:
  - **file transfer** (ftp, port 21)
  - **login remoto criptato** (ssh, port 22)
  - **login remoto** (telnet, port 23)
  - **posta elettronica** (smtp, port 25)
  - **TFTP** (port 69) (esiste anche su UDP)
  - **HTTP** (port 80) (il protocollo del World Wide Web)
  - …

Servizi di Condivisione Apple

- Apple Talk Suite Protocol
  - Insieme di protocolli per differenti funzioni
    - AFP : trasferimento file
    - PAP : condivisione stampanti
    - ZIP : definzione delle zone (usato nei router)
    - ASP : gestore delle sessioni
    - ATP : gestione delle transazioni
    - Il livello rete utilizza datagram DDP (in maniera innaffidabile)
    - AARP : traduzione indirzzi fisici in logici

Apple Talk Suite Protocol



AARP

- AARP (AppleTalk Address Resolution Protocol) gestione della mappatura tra indirizzi fisici e logici.
- In grado di supportare qualunque implementazione del livello 2

DDP

- The Datagram Delivery Protocol (DDP) : invio di datagram e servizio di routing verso I protcolli di lato livello

RTMP

- The Routing Table Maintenance Protocol (RTMP) : gestisce le informazioni per eseguire l'indirizzamento in una rete apple talk.

AEP

- The AppleTalk Echo Protocol (AEP) : servizio di echo

ATP

- The AppleTalk Transaction Protocol (ATP) : assicura un servizio affidabile per *transaction-oriented operations*.
- Utilizza token per gestire acknowledgement e il flusso di controllo gestiti dai protocolli di livello superiore

NBP

- The AppleTalk Name Binding Protocol (NBP) : gestisci I nomi all'interno di una rete AppleTalk.
- NBP mantiene *names directory* che include I nomi delle macchine e il loro indirizzo.
- Dopo che un nome è registrato si può effettuare una ricerca per nome ed ottenere l'indirizzo
- Analogo al DNS

ZIP

- The AppleTalk Zone Information Protocol (ZIP) : gestisce le relazioni tra segmenti di reti e zone.

ASP

- The AppleTalk Session Protocol (ASP) : gestione delle sessioni per livelli superiori.
- Unico identificatore di sessione per le connessioni logiche

PAP

- The Printer Access Protocol (PAP) : condivisione stampanti.

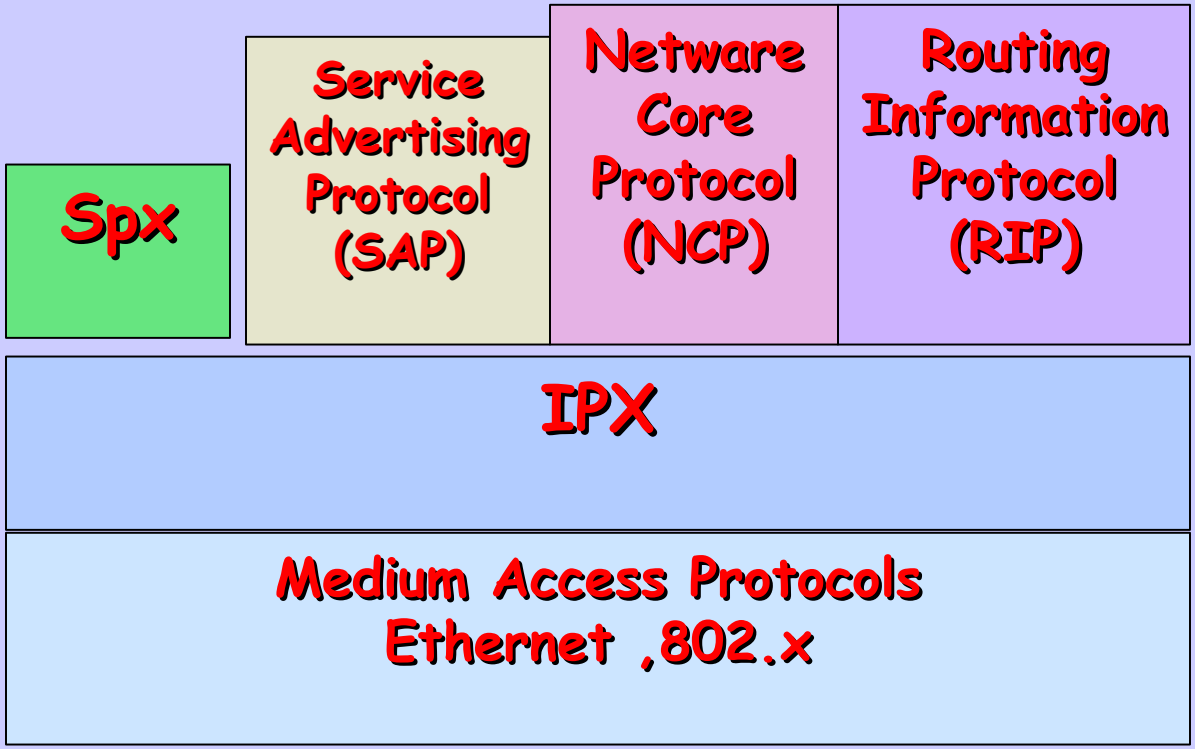
ADSP

- The AppleTalk Data Stream Protocol (ADSP) : protocollo connection.oriented che garantisce l'invio in sequenza dei dati con controllo di flusso.

AFP

- The AppleTalk Filing Protocol (AFP) : servizio di condiviosne dei files.

Architettura Novell IPX/SPX



La suite IBM-Microsoft: NetBeui-NetBios

- **NetBios: Network Basic Input Output System:**
  - **Un API application programming interface**
  - **Migliora il BIOS del DOS sommando speciali funzioni legate alle risorse disponibili sulla LAN.**
  - **Alcune aziende hanno migliorato il protocollo sommando**
  - **funzioni di network**
  - **Gestisce formato di messaggio definito:**
  - **Server Message Block (SMB).**

La suite IBM-Microsoft: NetBeui-NetBios

- **Netbeui: NetBios Enhanced User Interface:**
  - **E` un protocollo di trasporto.**
  - **una versione avanzata del protocollo NetBIOS usato nei sistemi operativi di rete come LAN Manager, LAN Server, Windows for Workgroups, Windows 95 e Windows NT.**
  - **Sviluppato inizialmente da IBM per il suo Lan Manager e` utilizzato poi da Microsoft e Novell.**

Identificazione degli Hosts su una rete NetBeui:

- Gli hosts sono definiti da due campi alfanumerici

- (ASCII) di 16 caratteri max ciascuno:
  - Nome proprio (unico).
  - Nome di un gruppo di appartenenza (dominio).

SMB Stack

OSI

Application Presentation	SMB				Application
Session	NetBIOS	NetBEUI	NetBIOS	NetBIOS	
Transport	IPX[1]		DECnet	TCP&UDP	TCP/UDP
Network				IP	IP
Link	802.2, 802.3,802.5	802.2 802.3,802.5	Ethernet V2	Ethernet V2	Ethernet or others
Physical					

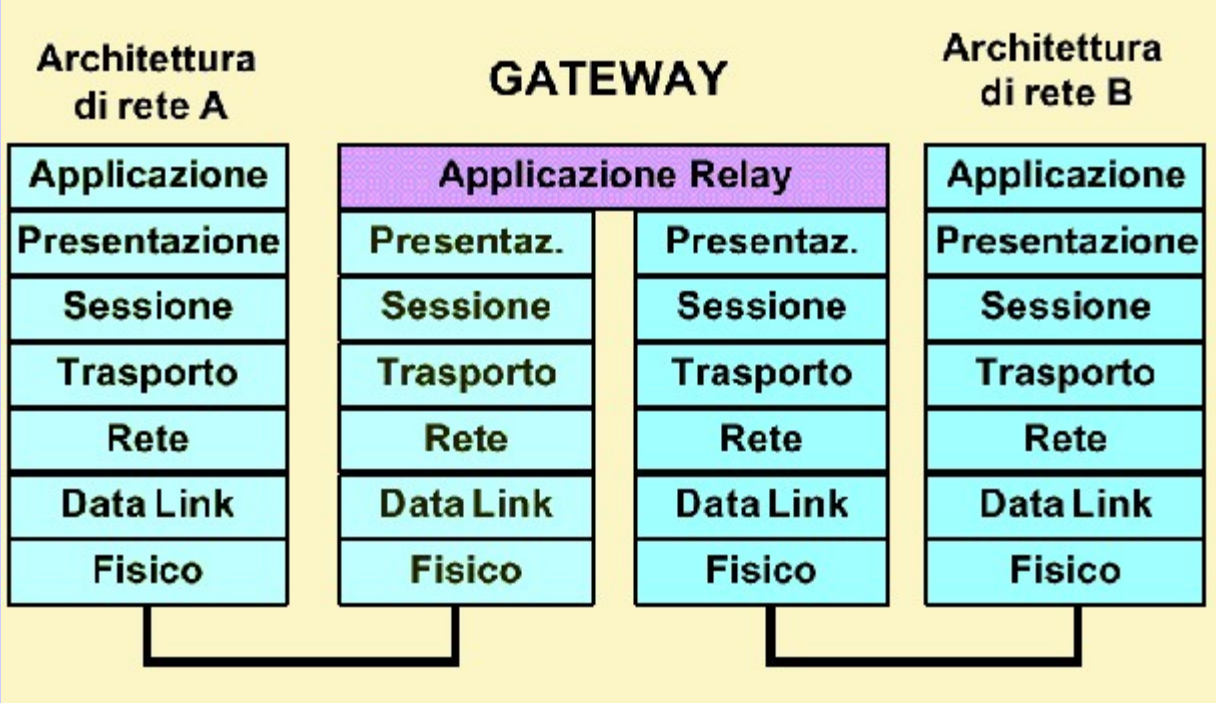
Servizi Internet LAN-WAN:
- **Telnet**
- **FTP**
- **Http**
- **Smtp**
- **NFS**
- **…**

Condivisione risorse LAN:
- **Stampanti**
- **FileSystems**

Confronto tra i diversi protocolli

- I protocolli visti non sono compatibili tra loro
- Possono coesistere su tecnologia ethernet
  - Alla medesima rete fisica possono essere collegati macchine diverse
- Possono scmbiarsi dati attraverso gateway che effettua uan conversione di protocolli
- IP tunneling
  - I diversi pacchetti sono incapsulati in pacchetti IP
  - I router inidirzzano i pacchetti IP nel giusto tratto di rete
  - Il router locale reimmetterà il pacchetto originale sulla rete

Gateway

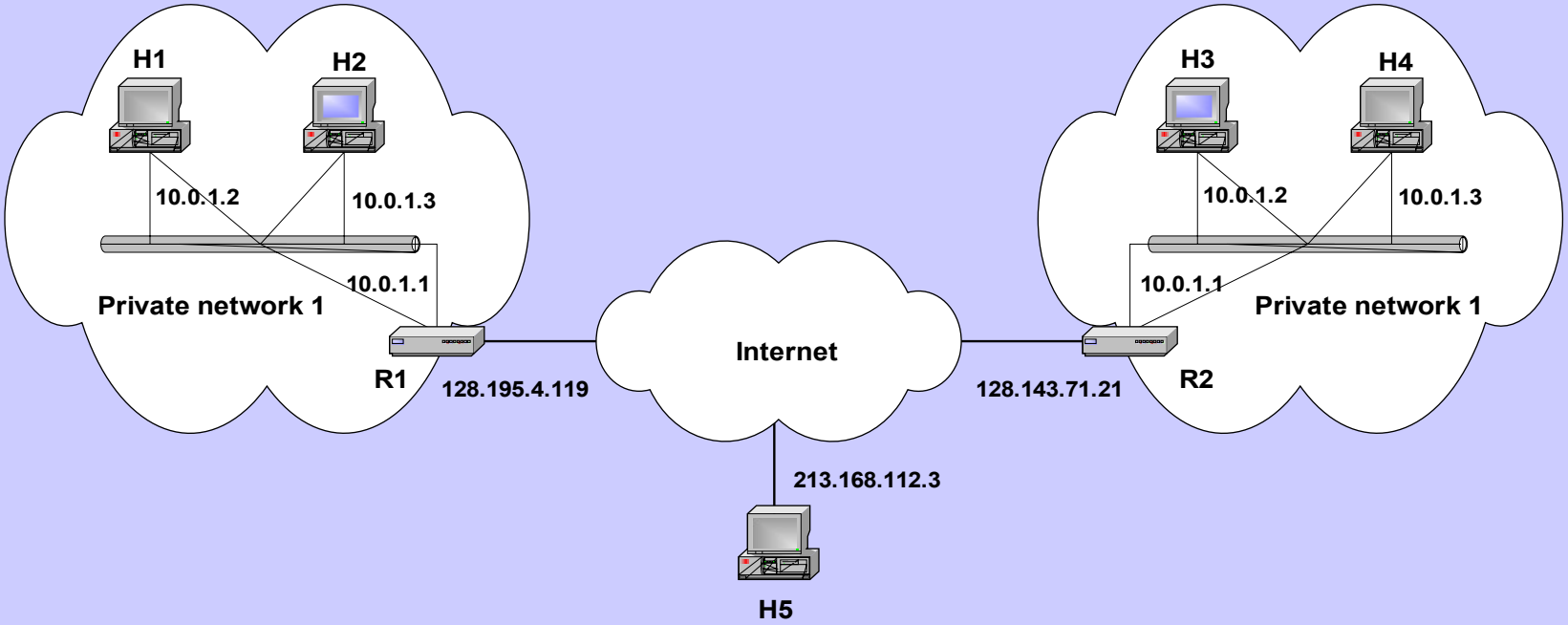

Architettura di rete A
GATEWAY
Architettura di rete B
Applicazione
Presentazione
Sessione
Trasporto
Rete
Data Link
Fisico
Applicazione Relay
Presentaz.
Sessione
Trasporto
Rete
Data Link
Fisico
Presentaz.
Sessione
Trasporto
Rete
Data Link
Fisico
Applicazione
Presentazione
Sessione
Trasporto
Rete
Data Link
Fisico

NAT (Network Address Translation)

- *ndirizzo IP privato:* fa parte di una rete che non è direttamente connessa a internet

- Un indirizzo IPall'interno di una rete può essere assegnato arbitrariamente.
  - Un indirizzo non registrato non garantisce l'unicità dell'indirizzo stesso
- In genere le reti private usano I seguenti indirizzi (*non-routable addresses*):
  - 10.0.0.0 – 10.255.255.255 (Classe A)
  - 172.16.0.0 – 172.31.255.255 (Classe B)
  - 192.168.0.0 – 192.168.255.255 (Classe C)

Indirizzi Privati



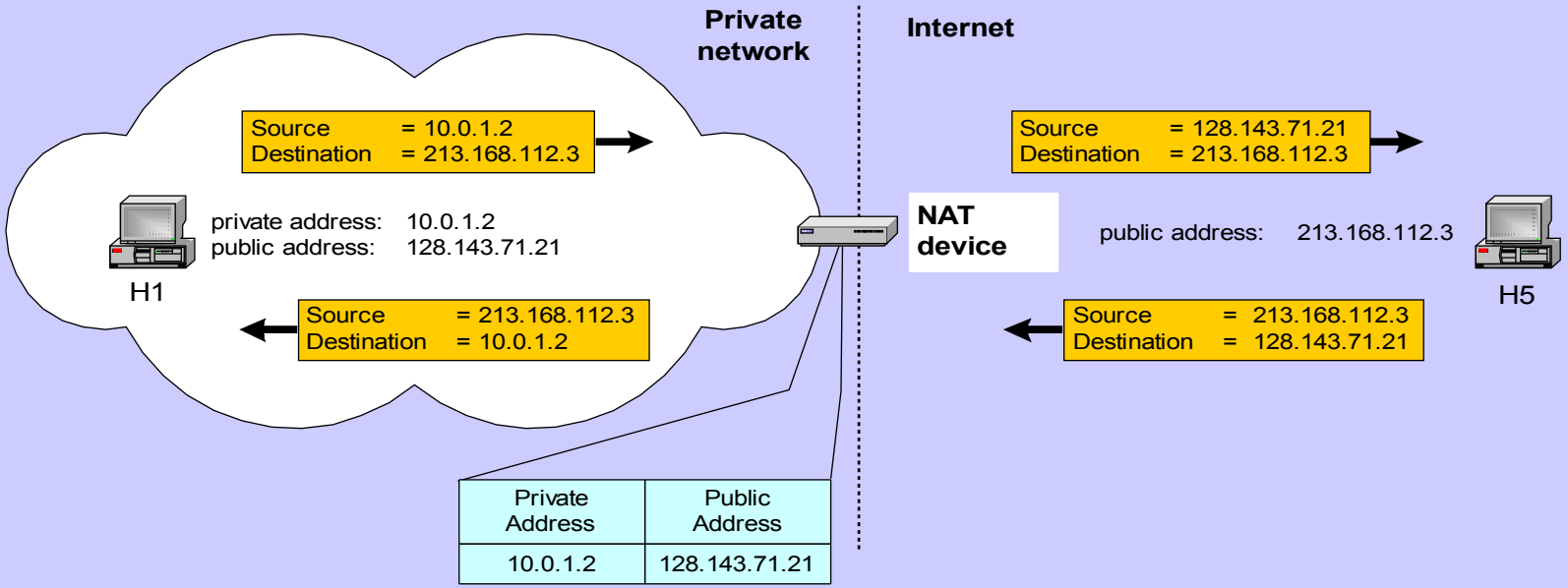
Network Address Translation (NAT)

- NAT è una funzione in cui l'indirizzo IP (ed eventualmente la porta) di un datagram IP sono sostituiti al *confine* di una rete privata

- NAT è un metodo che consente le macchine di una rete privata di comuincare con le altre macchine su internet

- NAT è eseguito sul router che collega la rete privata alla rete pubblica (internet), rimpiazza la coppia *indirizzo IP , porta* in un altra coppia *indirizzo IP , porta* .

NAT



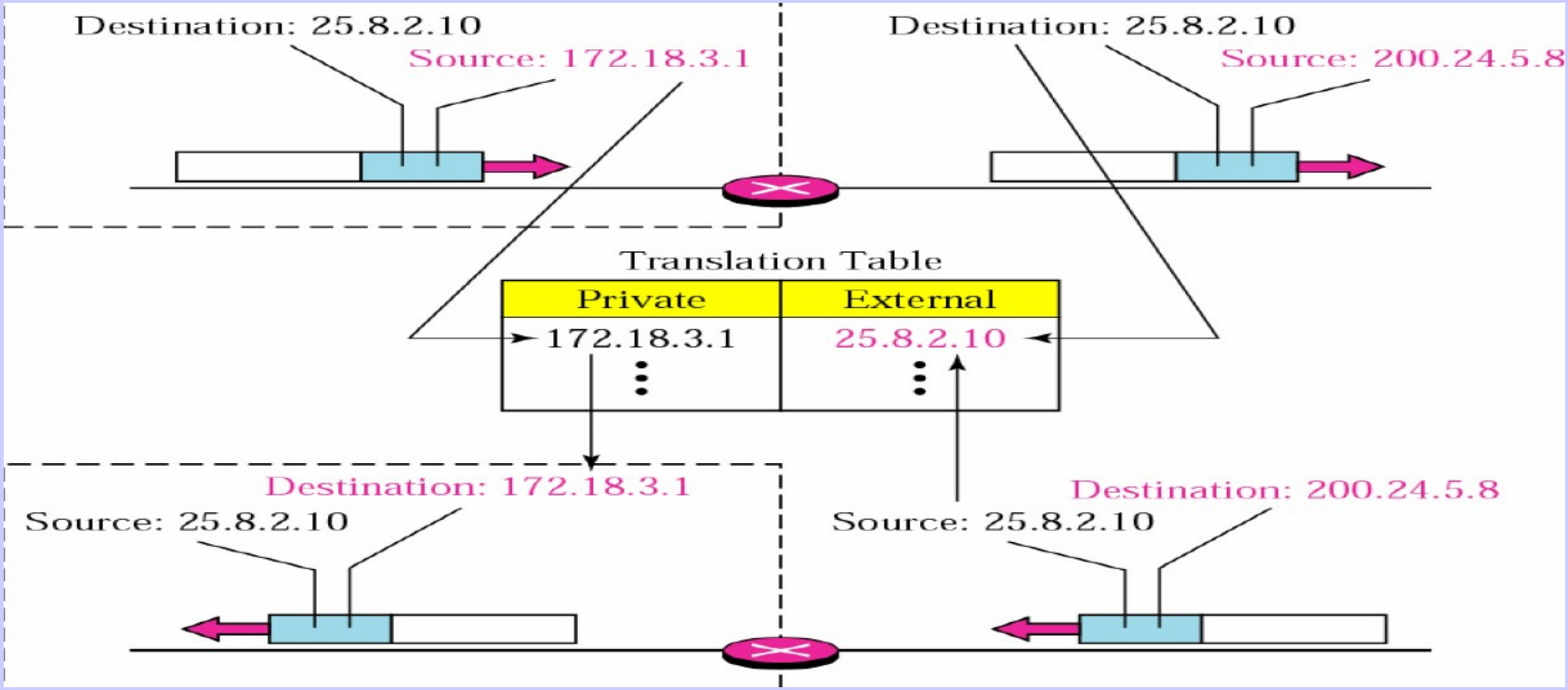
- NAT deve avere una *address translation table*

Network Address Translation (NAT)

- La tecnica del NAT è pricipalmente utilizzata per tre scopi:
  - realizza un tipo di firewall che nasconde gli indirizzi IP interni
  - consente di utilizzare più indirizzi privati senza avere problemi di coinflitti con indirizzi IP.
  - Permette di condividere una linea tra diverse connessioni.

Cosa fa il NAT

- NAT utilizza una tabella di tralsazione



Cosa fa il NAT

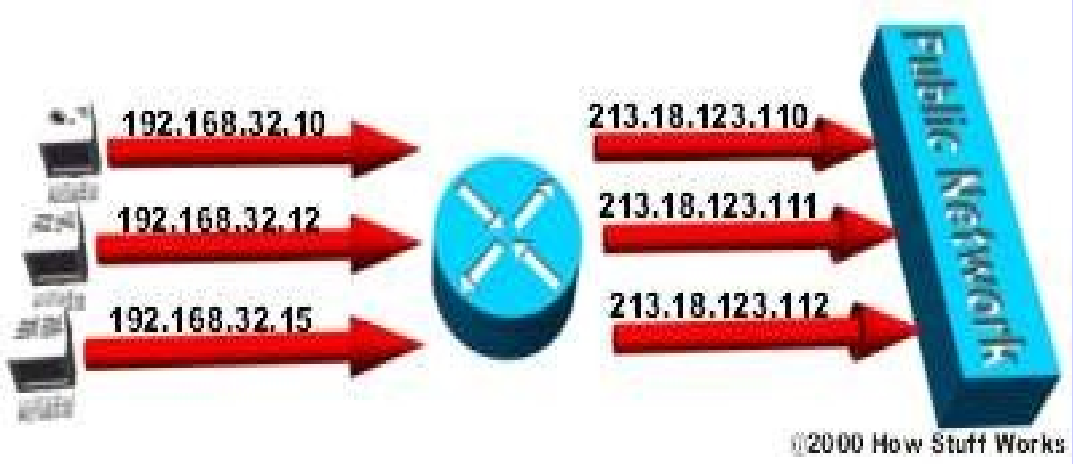
- Quando un host della rete privata invia un datagram ad un host della rete pubblica, il processo di NAT preleva un indirizzo IP pubblico (dall'insieme di IP pubblici) e lo associa a quello privato dell'host che ha iniziato la comunicazione

Cosa fa il NAT

- In generale il NAT traduce *m* indirizzi IP in *n* indirizzi IP
- Considerandi *m*, *n* ≥ 1. Esistono tre casi possibili:
  - **Static NAT (SNAT):** m=n
  - **Dynamic NAT (DNAT):** m>n
  - **Overloading:** m>n =1

Static NAT

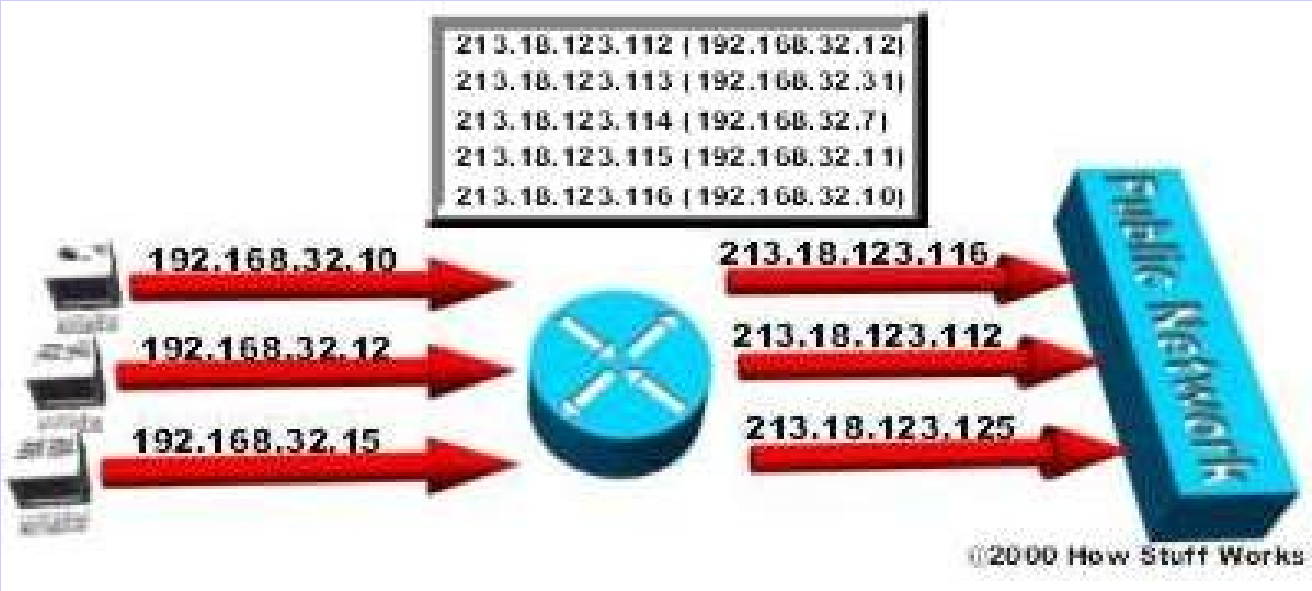
- crea in modo statico la tabella di mapping tra indirizzo IP privato e indirizzo IP pubblico



static NAT: il computer con indirizzo IP pari a 192.168.32.10 sarà sempre convertito con 213.18.123.110.

Dynamic NAT

- Mappa un indirizzo IP privato in uno pubblico selezionato tra un insieme di indirizzi IP pubblici



dynamic NAT: il computer con indirizzo IP pari a 192.168.32.10 sarà convertito con il primo indirizzo IP pubblico disponibile tra l'insieme 213.18.123.100 fino a 213.18.123.150.

Overloading

• un titpo di DNAT in cui più indirizzi IP privati sono mappati in un unico indirizzo IP pubblico utilizzando differenti porte.
•Tecnica conosciuta come PAT (Port Address Translation), single address NAT or port-level multiplexed NAT.

In PAT, ogni computer della rete privata sarà cobevrtito con il medesimo indirizzo IP (213.18.123.100), ma con differenti porte

Sicurezza con tecniche NAT

- Firewall:
  - Ricorre alla tecnica di DNAT crea automaticamente un **firewall** tra la rete privata e il resto del mondo
  - Il NAT consente accessi a internet solo alle macchine interne alla rete.
    - Le macchine delle reti esterne non possono connettersi ad una macchina all'interno della rete privata
    - È possibile collegarsi ad internet, scaricare files, ecc… senza lasciare traccia del proprio indirizzo IP.

Sicurezza con tecniche NAT

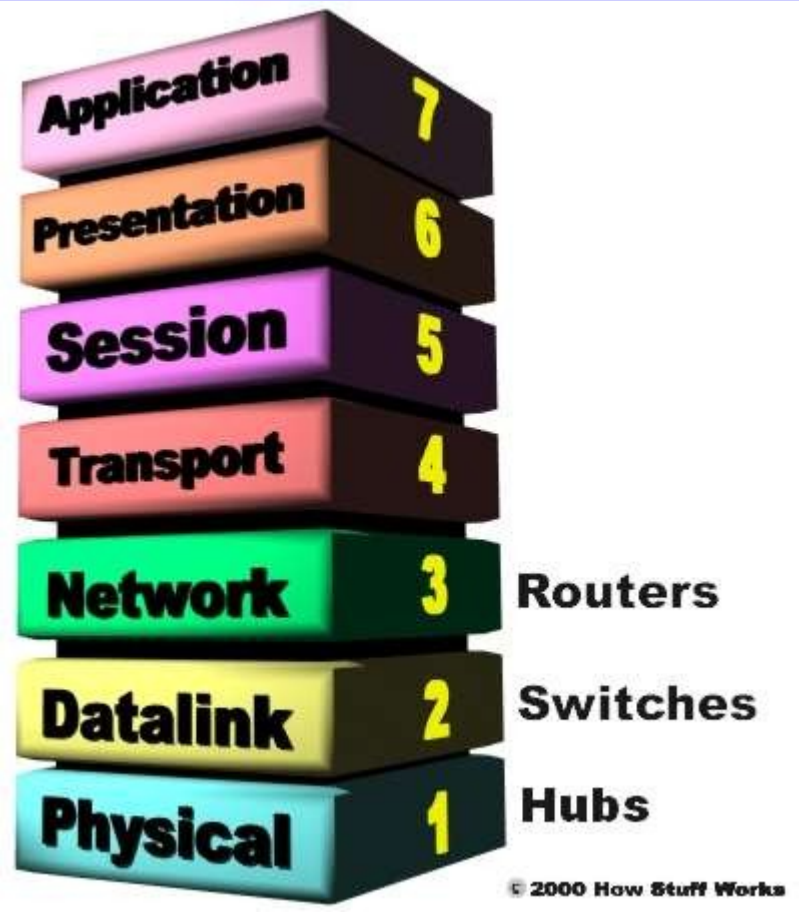
- Sicurezza:
  - Non è possibile capire la topologia della rete privata (numero di host)
  - Il NAT può escludere destinazioni sospette

Sicurezza con tecniche NAT

- Filtraggio e log del traffico:
  - Il filtraggio consente di controllare quali siti l'tente visiti
  - Blocco di materiale non desiderato.

- Generare un report con I siti visitati
- E la Privacy?

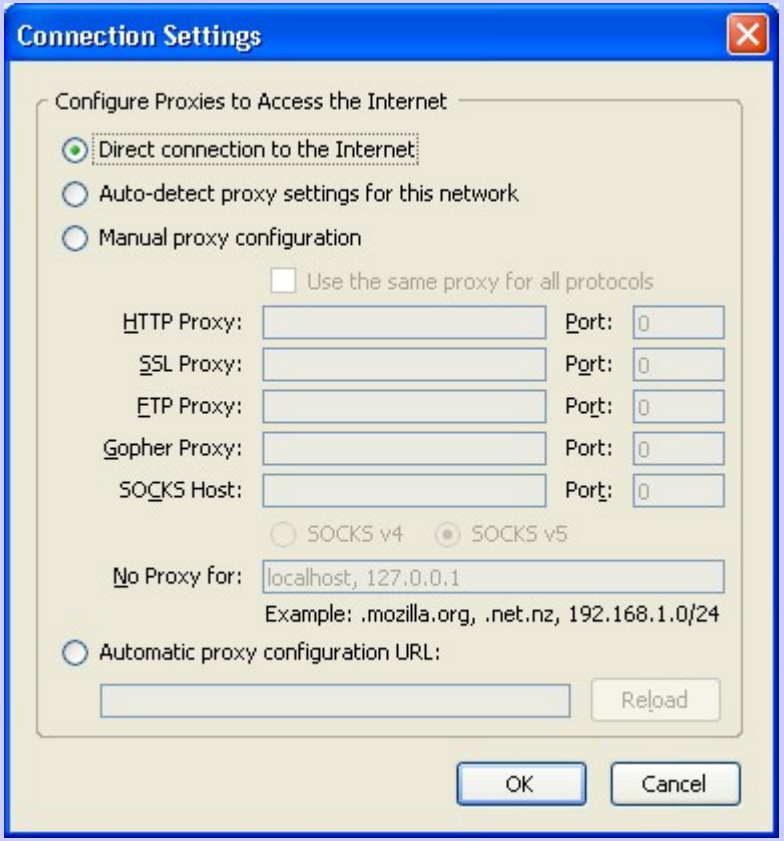
Sicurezza con tecniche NAT



Il NAT opera a livello 3 (Network layer)del modello OSI

Address Proxy

- Un proxy server implementa una (o più) delle tecniche di NAT viste prima
- È possibile abilitare il proxy per diversi protocolli (ftp, http, ecc)
- Deve essere impostato il browser dal lato client



Cache Proxy

- Quando viene richiesta una pagina web, il proxy prima di connetersi al sito scaricarla verifica che non esista una copia locale (sul proxy stesso)
- Se si restituisce al richiedente la copia locale della pagina
- Se non esiste una copia o se non è aggiornata si connette al sito web e scarica la pagina in questione
- Permette di ridurre il numero di connessioni
- Es: squid

FireWall

- Un firewall (muro di fuoco) è un insieme di programmi correlati che protegge le risorse di una rete privata da possibili accessi indesiderati
  - Impedire che dall'esterno si tenti di accedere a dati privati
- Generlamente installato su una macchina dedicata all'interno della rete
- Esamina ogni pacchetto per decidere se inoltrarlo o meno
- Puo utilizzare tecniche di NAT
- Lavora con Proxy Server

FireWall

- Ricorre a due metodi
  - Proxy services : realizzano il NAT
  - Packet filtering: ispezione di ogni singolo pacchetto
    - Necessario ipostare una serie di regole per valutare ogni pacchetto