RETI DI CALCOLATORI

Livelli ISO / OSI:
7) Applicazione : Fornisce i servizi (applicazioni) all'utente(login remoto, file transfer, e-mail)
6)Presentazione: Si occupa dei problemi relativi alla rappresentazione dei dati sintassi dell'informazione, Funzioni : conversione dei dati dal formato di trasmissione ad un formato utile all'applicazione • codifica e decodifica • compressione dei dati • crittografia
Protocolli: definizione del formato dei pacchetti, definizione strutture dati complessi definizione dei messaggi per lo scambio di informazioni, definizione degli algoritmi per codifica/decodifica, compressione, crittografia, ...
5)Sessione: Consente a due applicazioni di sincronizzarsi e gestire lo scambio dei dati
Funzioni: instaurazione e rilascio di una connessione di sessione, scambio di dati normali e di dati con priorità, gestione del dialogo tra entità comunicanti mediante token, sincronizzazione e strutturazione del dialogo, gestione delle eccezioni
Protocolli: definizione del formato dei pacchetti, definizione dei messaggi per lo scambio di informazioni, definizione degli algoritmi per il controllo della sessione
4)Trasporto: Fornisce un canale di trasporto ideale e privo di errori tra due utenti, indipendentemente dalla rete Funzioni: recupero degli errori, multiplazione / demultiplazione, riordino dei pacchetti, controllo della congestione
Protocolli = uguale a sopra (algoritmi congestione)
3)Rete: E' responsabile del trasferimento di informazioni tra nodi, indipendentemente dal tipo di collegamento. Funzioni: instradamento, internetworking Protocolli = uguale a sopra(algoritmi instradamento)
2)Data-link: Fornisce un canale numerico di comunicazione il più possibile affidabile, trasferimento di unità logiche di bit (trame) su un collegamento
Funzioni: gestione collegamento, framing (divisione delle trame) , controllo errori , controllo di flusso. Protocolli: definiscono il formato della trama, definiscono i messaggi di feedback per il controllo di flusso, definiscono gli algoritmi per la gestione trasmissione
1)Fisico: Gestisce la trasmissione del segnale su canale fisico Funzioni: trasferimento di un flusso seriale di bit, attivazione, disattivazione e controllo della connessione fisica
Protocolli: specificano le caratteristiche elettriche, meccaniche e procedurali • ad esempio: trasmissione on-off o antipodale, significato dell'ordine dei bit, formato della flag, …;

Socket: un processo invia/riceve messaggi a/da la sua socket! una socket è analoga a una porta" un processo che vuole inviare un messaggio, lo fa uscire dalla propria "porta" (socket) " il processo presuppone l'esistenza di un'infrastruttura esterna che trasporterà il messaggio del processo di destinazione
Messaggi HTTP: due tipi di messaggi HTTP: richiesta, risposta Messaggio di richiesta HTTP ASCII (formato leggibile dall'utente)
Metodo Post: La pagina web spesso include un form per l'input dell'utente, L'input arriva al server nel corpo dell'entità
Metodo URL: Usa il metodo GET, L'input arriva al server nel campo URL della riga di richiesta:

GET condizionale: Obiettivo: non inviare un oggetto se la cache ha una copia aggiornata dell'oggetto cache: specifica la data della copia dell'oggetto nella richiesta HTTP server: la risposta non contiene l'oggetto se la copia nella cache è aggiornata: HTTP/1.0 304 Not Modified

FTP:1)Il client FTP contatta il server FTP alla porta 21, specificando TCP come protocollo di trasporto 2) Il client ottiene l'autorizzazione sulla connessione di controllo Il client cambia la directory remota inviando i comandi sulla connessione di controllo 3) Quando il server riceve un comando per trasferire un file, apre una connessione dati TCP con il client4) Dopo il trasferimento di un file, il server chiude la connessione Client FTP. Server FTP
1)Porta 21 per la connessione di controllo TCP Porta 20 per la connessione dati TCP 2)Il server apre una seconda connessione dati TCP per trasferire un altro file.3)Connessione di controllo: "fuori banda" (out of band) 4) Il server FTP mantiene lo stato": directory corrente, autenticazione precedente

DNS: Query iterativa: Il server contattato risponde con il nome del server da contattare
Query ricorsiva: Affida il compito di tradurre il nome al server DNS contattato
Messaggi DNS Protocollo DNS: domande (query) e messaggi di risposta, entrambi con lo stesso formato Intestazione del messaggio, Identificazione: numero di 16 bit per la domanda; la risposta alla domanda usa lo stesso numero
Query flooding: Completamente distribuito nessun server centrale, Protocollo di pubblico dominio usato da Gnutella, Ciascun peer indicizza i file che rende disponibili per la condivisione (e nessun altro), Rete di copertura: grafo Arco tra i peer X e Y se c'è una connessione TCP Tutti i peer attivi e gli archi formano la rete di copertura, Un arco è un collegamento virtuale e non fisico
Trasferimento file:HTTP Il messaggio di richiesta è trasmesso sulle connessioni TCP esistenti Il peer inoltra il messaggio di richiesta Il messaggio di successo è trasmesso sul percorso inverso S

Socket = (Interfaccia di un host locale, creata dalle applicazioni, controllata dal SO (una "porta") in cui il processo di un'applicazione può inviare e ricevere messaggi al/dal processo di un'altra applicazione
Programmazione delle socket : Obiettivo: imparare a costruire un'applicazione client/server che comunica utilizzando le socket

Programmazione delle socket con TCP Il client deve contattare il server Il processo server deve essere in corso di esecuzione Il server deve avere creato una socket (porta) che dà il benvenuto al contatto con il client Il client contatta il server:1) Creando una socket TCP 2)Specificando l'indirizzo IP, il numero di porta del processo server 3) Quando il client crea la socket: il client TCP stabilisce una connessione con il server TCP 4) Quando viene contattato dal client, il server TCP crea una nuova socket per il processo server per comunicare con il client consente al server di comunicare con più client numeri di porta origine usati per distinguere i client

Programmazione delle socket con UDP: UDP: non c'è "connessione" tra client e server, Non c'è handshaking, Il mittente allega esplicitamente a ogni pacchetto l'indirizzo IP e la porta di destinazione, Il server deve estrarre l'indirizzo IP e la porta del mittente dal pacchetto ricevuto UDP: i dati trasmessi possono perdersi o arrivare a destinazione in un ordine diverso da quello d'invio Punto di vista dell'applicazione UDP fornisce un trasferimento inaffidabile di gruppi di byte ("datagrammi") tra client e server

HEADER TCP:Significato dei campi 1)Source port – Destination port [16 bit]: indirizzi della porta sorgente e della porta destinazione
2) Sequence Number [32 bit]: numero di sequenza del primo byte del payload. contiene il numero necessario per sapere quale sia l'ordine dei segmenti e per sapere se qualcuno è andato perduto. Diversamente da come si potrebbe pensare, non viene usato il numero di segmento, ma il numero del primo byte di quel segmento. Quindi se ogni segmento contiene 1500 byte, il sequence number sarà 0…1500…3000… , e non 0…1…2… .
3)Acknowledge Number [32 bit]: numero di sequenza del prossimo byte che si intende ricevere (ha validità se il segmento è un ACK).
-viene usato per segnalare che avete ricevuto tutti i dati fino al numero di byte specificato meno uno, e dovrebbe essere uguale al valore del prossimo Sequence number che sarà ricevuto.
4) Offset [4 bit]: lunghezza dell'header TCP, in multipli di 32 bit
5) Reserved [6 bit]: riservato per usi futuri
6) Window [16 bit]: ampiezza della finestra di ricezione (comunicato dalla destinazione alla sorgente)
nei segmenti con il flag ACK, indica in byte l'ampiezza della finestra che il computer è in grado di ricevere. Per il funzionamento vedi il paragrafo Il meccanismo delle finestre scorrevoli.
7) Checksum [16 bit]: risultato di un calcolo che serve per sapere se il segmento corrente contiene errori nel campo dati.
8) Urgent pointer [16 bit]: indica che il ricevente deve iniziare a leggere il campo dati a partire dal numero di byte specificato. Viene usato se si inviano comandi che danno inizio ad eventi asincroni "urgenti"
Significato dei campi
Flag [ogni flag è lunga 1 bit]:
– URG: vale uno se vi sono dati urgenti; in questo caso il campo urgent pointer ha senso
– ACK: vale uno se il segmento è un ACK valido; in questo caso l'acknowledge number contiene un numero valido
– PSH: vale uno quando il trasmettitore intende usare il comando di PUSH;
– RST: reset; resetta la connessione senza un tear down esplicito
– SYN: synchronize; usato durante il setup per comunicare i numeri di sequenza iniziale
– FIN: usato per la chiusura esplicita di una connessione
Options and Padding [lunghezza variabile]: riempimento (fino a multipli di 32 bit) e campi opzionali come ad esempio durante il setup per comunicare il MSS Data: i dati provenienti dall'applicazione.

TCP/APERTURA CONNESSIONE: Il TCP è un protocollo connection oriented prima di iniziare a trasferire i dati ci deve essere una connessione tra i due end-system
L'instaurazione della connessione avviene secondo la procedura detta di "three-way handshake"
1)a stazione che richiede la connessione (A) invia
un segmento di SYN • parametri specificati: numero di porta dell'applicazione cui si intende accedere e Initial Sequence Number (ISNA)
2) la stazione che riceve la richiesta (B) risponde con
un segmento SYN • parametri specificati: ISNB e riscontro (ACK) ISNA
3)la stazione A riscontra il segmento SYN della stazione B (invia un ACK alla stazione B)
L'instaurazione della connessione serve per scambiarsi i dati relativi alla comunicazione numero di porta (applicazione), numero iniziale di sequenza, dimensione della finestra, MSS, varie opzioni,Il canale di comunicazione aperto è bidirezionale entrambe le stazioni possono trasferire dati

TCP/CHIUSURA CONNESSIONE: Poiché la connessione è bidirezionale, la terminazione deve avvenire in entrambe le direzioni Procedura di terminazione
1)la stazione che non ha più dati da trasmettere e decide di chiudere la connessione invia un segmento FIN (segmento con il campo FIN posto a 1 e il campo dati vuoto)
2)la stazione che riceve il segmento FIN invia un ACK e indica all'applicazione che la comunicazione è stata chiusa nella direzione entrante Se questa procedura avviene solo in una direzione (half close), nell'altra il trasferimento dati può continuare (gli ACK non sono considerati come traffico originato, ma come risposta al traffico)
3) per chiudere completamente la connessione, la procedura di half close deve avvenire anche nell'altra direzione

HEADER UDP: Source Port e Destination Port [16 bit]: identificano i processi sorgente e destinazione dei dati
Length [16 bit]: lunghezza totale (espressa in byte) del datagramma, compreso l'header UDP
Checksum [16 bit]: campo di controllo che serve per sapere se il datagramma corrente contiene errori nel campo dati è' utilizzato per il supporto di transazioni semplici tra applicativi e per le applicazioni multimediali

DATAGRAMMA IP = Nello stack TCP/IP si utilizza il termine datagramma IP per riferirsi ad un pacchetto, Ciascun datagramma è formato da un header lungo da 20 a 60 bytes,

contenente informazioni essenziali per instradamento e la consegna del datagramma stesso seguito dai dati (payload) – La dimensione dei dati non e fissa, ma determinata dall applicazione che invia i dati – Un datagramma può contenere un solo byte o fino a 64K byte

HEADER IP: Contiene informazioni utili per trasferire il datagramma stesso
Le informazioni dell'header includono:
– l'indirizzo della sorgente (chi ha inviato inizialmente il datagramma)
– l'indirizzo della destinazione (a chi va consegnato)
– un campo che specifica il tipo di dati trasportato nel payload
Gli indirizzo negli header sono indirizzi IP formato standard che vedremo successivamente
Ciascun campo dell'header ha una dimensione fissa in tal modo il processing dell'header può essere fatto in maniera efficiente:
VERS : 4 bit che specificano la versione del protocollo
H.LEN (header length) 4-bit utilizzati per specificare la dimensione dell'header (numero totale di byte / 4)
SERVICE TYPE: 8-bit che identificano la classe di servizio del datagramma (usato raramente)
TOTAL LENGTH: intero a 16-bit che specifica il numero totale di byte del datagramma intero (header + payload)
IDENTIFICATION numero di 16-bit (di solito sequenziale) assegnato al datagramma
• utilizzato per ricomporre un datagramma nel caso in cui venga frammentato
• Utilizzato, come da specifiche iniziali, per identificare in modo univoco i vari frammenti in cui può essere stato "spezzato" un pacchetto IP. Alcune sperimentazioni successive hanno però suggerito di utilizzare questo campo per altri scopi, come aggiungere la funzionalità di tracciamento dei pacchetti;
FLAGS: 3-bit, dove ciascun bit specifica se il datagramma e un frammento o meno, ed eventualmente se è l'ultimo frammento
FRAGMENT OFFSET: 13-bit che specificano l'offset del frammento rispetto al datagramma originale, il valore del campo deve essere moltiplicato per 8 per ottenere il vero offset
- Indica l'offset (misurato in blocchi di 8 byte) di un particolare frammento relativamente all'inizio del pacchetto IP originale: il primo frammento ha offset 0
TIME TO LIVE intero a 8-bit inizializzato dalla sorgente, viene decrementato da ciascun router attraversato dal datagramma se raggiunge il valore 0, il datagramma viene scartato e un messaggio di errore viene inviato alla sorgente
TYPE: 8-bit che specificano il tipo di dati trasportato nel payload
HEADER CHECKSUM: 16-bit checksum dell'header
SOURCE IP ADDRESS: indirizzo Internet di 32 bit della sorgente
DESTINATION IP ADDRESS: indirizzo Internet di 32 bit della destinazione
IP OPTIONS: Campi opzionali (non necessariamente presenti) con informazioni addizionali

PADDING Se il campo Option è presente e la sua dimensione non è un multiplo di 32 bit, vengono messi degli 0 per raggiungere il multiplo di 32 bit

TIME TO LIVE intero a 8-bit inizializzato dalla sorgente, viene decrementato da ciascun router attraversato dal datagramma se raggiunge il valore 0, il datagramma viene scartato e un messaggio di errore viene inviato alla sorgente. Il time to live (TTL) è un meccanismo che determina il tempo di vita di un dato in un host di una rete.
Nel caso di un pacchetto IP, determina il numero di volte alla quale un router può accedervi prima che venga distrutto. L'utilità di questa funzione diventa evidente se si immagina una situazione in cui per errore o guasto si viene a creare un loop in una catena di router. Se non esistesse il TTL i pacchetti viaggerebbero indefinitamente nel circolo vizioso sovraccaricando inutilmente i router.

MESSAGGI DHCP: DHCP rilascia un indirizzo per un periodo limitato (lease) – In questo modo il server DHCP può tornare in possesso degli indirizzi. Quando il periodo di lease scade – il server considera l'indirizzo come disponibile per un'eventuale nuova assegnazione – un host può liberare l'indirizzo o rinegoziare con il server DHCP. Di solito, il server DHCP approva le richieste di estensione – L'host continua a lavorare senza interruzioni – In ogni caso, un server DHCP può essere configurato per negare l'estensione per ragioni tecniche o amministrative – Se il server nega l'estensione, l'host deve smettere di usare l'indirizzo • il server DHCP ha il controllo degli indirizzi

CAMPI DHCP: DHCP adotta un versione leggermente modificata del formato
dei messaggi BOOTP
– OP indica se si tratta di una Request o una Response – i campi HTYPE and HLEN il tipo di hardware della rete e la lunghezza dell'indirizzo hardware – FLAGS indica se l'host puo' ricevere messaggi broadcast o risposte
dirette – HOPS indica a quanti server rigirare la richiesta – TRANSACTION IDENTIFIER contiene un valore usato da un host per
capire se la risposta di riferisce ad una sua richiesta – SECONDS ELAPSED indica quanti secondi sono passati dall avvi dell host
DHCP: Formato dei messaggi: I campi finali sono usati per trasportare nelle risposte informazioni verso la sorgente – se un host non conosce il proprio indirizzo IP, il server usa il campo YOUR IP ADDRESS per fornire il valore - il server usa i campi SERVER IP ADDRESS e SERVER HOST NAME per fornire all'host informazioni sulla posizione del server – il campo ROUTER IP ADDRESS contiene l'indirizzo IP del router di
default

RIP = Routing Information Protocol : Protocollo intra-dominio semplice, Implementazione diretta del routing basato su Distance, Vector… – Versione distribuita dell'algoritmo di Bellman-Ford (DBF) …con i problemi noti di tali algoritmi – convergenza lenta (in caso di guasto) – funziona con reti di dimensione limitata. Punti di forza – semplice da implementare, semplice da gestire, uso diffuso.
Metrica basata su conteggio degli hop : – valore massimo e 15, considerato come ∞
• imposto per limitare il tempo di convergenza – l'amministratore di rete può assegnare valori maggiori di 1 al singolo hop, Ciascun router invia i vettori delle distanze ogni 30 secondi (qualora le tabelle di routing cambino per motivi esterni) a tutti i vicini
– RIP usa UDP, porta 520, per l'invio dei messaggi, I cambiamenti si propagano sulla rete, Le entry hanno un timeout di 3 minuti – se scade, la distanza viene posta a 15
RIP formato dei messaggi : Command: 1=request 2=response, Gli aggiornamenti sono considerati, response sia che ci sia stata una richiesta esplicita che non – Un nodo appena connesso invia in broadcast le richieste – Alle richieste si risponde immediatamente Version: 1, Address family: 2 per IP IP address: la parte di Host ID è sempre posta a zero, Metric: – Distanza dal router alla rete specificata nell indirizzo IP -- Tipicamente = 1, ovvero la metrica rappresenta il numero di hop
RIP mantiene 3 timer per le proprie operazioni
– Aggiornamento periodico (25-30 sec)
• usato per inviare i messaggi di aggiornamento
– Timer di invalidazione (180 sec)
• Se un entry non è stata aggiornata per 180 secondi, essa non viene ritenuta più valida
– Timer per il garbage collection (120 sec)
• Un entry non valida viene marcata, ma non rimossa
• Per 120 sec il router include la destinazione ma con distanza infinita

RIP: input processing: Messaggi di Request: – generati da router appena avviati – azione: il router risponde direttamente a chi ha fatto la richiesta
Messaggi di Response: – possono arrivare da router che inviano messaggi di aggiornamento, o in risposta ad una query specifica – azione: il router aggiorna la sua tabella di routing
RIP: output processing: Un output viene generato – quando un router viene avviato – se richiesto dalla procedura di processing degli input – dall aggiornamento regolare. Azione: il router genera il messaggio a seconda del comando ricevuto

RIPv2: Formato dei messaggi Version: 2, Route Tag: usato per trasportare informazioni di altri protocolli di routing, Maschera di subnet della rete identificata dall'indirizzo IP, Next hop – identifica un indirizzo di nexthop migliore rispetto a quello pubblicizzato dal router (se esiste altrimenti impostato a zero). Uso esplicito delle subnet Interoperabilità
– RIPv1 e RIPv2 possono essere usati sulla stessa rete perchè RIPv1 ignora i campi sconosciuti• RIPv2 risponde alle richieste di RIPv1 con risposte RIPv1. Multicast – invece di inviare i messaggi di RIP in broadcast, RIPv2 usa l indirizzo dimulticast 224.0.0.9

RIP: limitazioni (il costo della semplicità)Uso esplicito delle subnet, Interoperabilità – RIPv1 e RIPv2 possono essere usati sulla stessa rete perché RIPv1 ignora i campi sconosciuti • RIPv2 risponde alle richieste di RIPv1 con risposte RIPv Multicast – invece di inviare i messaggi di RIP in broadcast, RIPv2 usa l indirizzo di multicast 224.0.0.9

LINK STATE VS DISTANCE VECTOR : Link State: Le informazioni sulla topologia sono inviate su tutta la rete (flooding), Il miglior cammino viene calcolato da ciascun router localmente, Il miglior cammino determina il next-hop, Funziona solo se la metrica e condivisa e uniforme Esempio: OSPF. Distance Vector: Ciascun router ha una visione limitata della topologia della rete , Data una destinazione è possibili individuare il miglior next-hop, Il cammino end-to-end è il risultato della composizione di tutte le scelte di next-hop, Non richiede metriche uniformi tra tutti i router, Esempio: RIP

IPV6: IPv6 mantiene molte delle caratteristiche di IPv4, tra cui – Come IPv4, IPv6 è connectionless – Come IPv4, l header del datagramma contiene un numero massimo di hop che il datagramma può fare prima di essere scartato
Dimensione degli indirizzi: – invece di 32 bit, gli indirizzi di IPv6 sono formati da 128 bit – lo spazio di indirizzamento dovrebbe essere sufficiente per contenere eventuali crescite future • ci sono circa NA (= numero di Avogadro = 6*1023) indirizzi per metro quadroq
Formato dell'header:– Completamente differente rispetto a IPv4 Introduzione del concetto di Extension Header: – IPv6 raggruppa le informazioni in header separati – Un datagramma consiste in un header IPv6 di base, seguito da nessuno o
più extension header, seguiti dai dati, Supporto del traffico Real-Time– introdotto un meccanismo che permette di creare un cammino tra una sorgente e una destinazione, e di associare i datagrammi a tale cammino – utilizzato da applicazioni audio e video

Protocollo estensibile: – IPv6 permette alla sorgente di aggiungere informazioni addizionali al datagramma Un datagramma IPv6 contiene una serie di header – ciascun datagramma inizia con un header di base – seguito da nessuno o piu extension header – seguito dal payload
Sebbene l'header IPv6 sia grande il doppio dell header IPv4,
contiene meno campi: l'header base ha una lunghezza fissa (40 byte)

IPv6: Formato dell'header di base: VERS (Versione: 6), TRAFFIC CLASS: – specifica la classe di traffico in base al tipo di traffico – rientra nel framework differentiated services per specificare i requisiti che la rete dovrebbe soddisfare, PAYLOAD LENGTH – specifica la dimensione del payload (dati trasportati dopo l header) – in IPv4 c era un campo total length che invece includeva l'header, HOP LIMIT – corrisponde al campo TIME-TO-LIVE di IPv4, FLOW LABEL: – usato per associare un datagramma con un cammino specifico, NEXT HEADER: – campo con un doppio significato: specifica il tipo di informazione che segue l'header corrente – Se il datagramma include un extension header

• il campo NEXT HEADER indica il tipo di extension header – Se non ci sono extension header
• il campo NEXT HEADER specifica il tipo di dati trasportato nel payload

Frammentazione, Riassemblaggio e Path MTU: Frammentazione in IPv6 è simile alla frammentazione in IPv4 ma ci sono differenze, Come in IPv4 – il prefisso in ciascun datagramma viene copiato in ciascun frammento – la dimensione del payload viene modificato in base alla dalla Maximum Transmission Unit (MTU) della rete da attraversare Diversamente da IPv4: – non esistono campi predeterminati nell header di base per la frammentazione – bisogna aggiungere un extension header con le informazioni sulla frammentazione • la presenza stessa di un extension header di tipo frammentazione indica che si tratta di un frammento

IPv6 introduce il concetto di gerarchia multi-livello: – Sebbene l'assegnazione degli indirizzi non e fissa, si può assumere che • il livello più alto corrisponde agli ISP • il livello successivo corrisponde ad un organizzazione (ad es., azienda)a

IPV6 Indirizzi Speciali: unicast: L'indirizzo corrisponde ad un singolo host. Un datagramma inviato a tale indirizzo viene instradato sul cammino minimo
multicast: L'indirizzo corrisponde ad un insieme di host e i membri dell'insieme possono cambiare in qualsiasi momento. Viene consegnata una copia del datagramma a ciascun membro dell insieme. anycast: l'indirizzo corrisponde ad un insieme di host che condividono un prefisso. Il datagramma viene consegnato ad uno qualsiasi dei membri dell'insieme.

Ethernet e Standard IEEE 802.3:Ambito di utilizzo – reti locali (LAN) • uffici, campus universitari,, Tecnologia economica, facilità di installazione e manutenzione
Si interfaccia direttamente e gestisce il livello fisico, Sopporta un carico medio del 30% (3 Mb/s) con picchi del 60% (6 Mb/s) Sotto carico medio – Il 2-3% dei pacchetti ha una sola collisione Qualche pacchetto su 10,000 ha più di una collisione

Principale differenza tra Ethernet e 802.3: – 802.3 definisce un'intera famiglia di sistemi CSMA/CD con velocità 1-10Mbps – Ethernet è solamente a 10Mbps
Gli standard Ethernet e 802.3 implementano un livello MAC di tipo CSMA/CD 1-persistent In caso di collisione, l'istante in cui ritrasmettere viene calcolato utilizzando un algoritmo di

binary exponential backoff – dopo i collisioni, l'host attende prima di ri-iniziare la procedura di
trasmissione un tempo casuale nell'intervallo [0, 1, ..., 2i-1] – vincoli • dopo 10 collisioni il tempo di attesa è limitato all'intervallo [0, 1, ..., 1023] • dopo 16 collisioni viene riportata una failure al sistema operativo

Formato della trama
PREAMBOLO (7 byte) – sequenza di byte "10101010" utilizzata per sincronizzare il ricevitore, START OF FRAME(1 byte) – flag di inizio della trama ,ADRESSES(6 byte) – indirizzi destinazione e sorgente della trama, LENGHT (2 byte) lunghezza in byte della trama (0-1500), PAYLOAD: informazione trasmessa, CHECKSUM:codice per rilevazione di errore

PPP per framing = E' un protocollo di livello 2 utilizzato sia nell'accesso e che nel backbone, Caratteristiche principali sono: character oriented, character stuffing per il framing, identificazione degli errori, supporta vari protocolli di livello superiore (rete) negoziazione dinamica degli indirizzi IP, autenticazione del "chiamante",
Il formato della trama è composto da:
FLAG (1 byte): identifica inizio e fine della trama.
ADDRESS(1 byte): utilizzato in configurazione "tutti gli host"
CONTROL (1 byte): valore finito unnumbered di default non fornisce un servizio affidabile: richiesta di ritrasmissione e rimozione replicazioni sono lasciate ai livelli superiori
PROTOCOL (1 o 2 byte) identifica il tipo di livello di frame ,
PAYLOAD(>0 byte): informazione trasmessa,
CHECKSUM(2 o 4 byte): identificazione dell'errore.
Ad esempio nell ADSL dove abbiamo: – Sistema asimmetrico su singola coppia – Rate adaptive: » 640 – 8200 kb/s downstream » Fino a 512 kb/s upstream – Strato di trasporto di livello 2: PPP su ATM – Distanze: a seconda del bit-rate

Risoluzione degli indirizzi: L'associazione tra un indirizzo IP di un host e il suo corrispondente indirizzo hardware è nota come risoluzione degli indirizzi, La risoluzione degli indirizzi avviene localmente – semplice nel caso di connessioni Point-to-Point, -- più complicata nel caso di mezzi condivisi (ad es., Ethernet) = serve un protocollo specifico

ARP/Address Resolution Protocol = Arp è nato per risolvere il problema che un host non poteva sapere se l'indirizzo da risolvere appartenesse alla stessa rete fisica (in quanto lindirizzo IP da risolvere (di destinazione) deve avere lo stesso prefisso (NetID) dell'host sorgente). Arp non risolve solo gli indirizzi IP e MAC ma è uno standard generale e specifica i diversi messaggi a seconda dei protocolli coinvolti, Per questo non è possibile avere una dimensione prefissata per contenere l'indirizzo hardware di un host
– La soluzione sta nell'avere un campo iniziale (di dimensione fissa) che indica la dimensione dell'indirizzo hardware utilizzato. Ad esempio, se ARP viene usato con Ethernet Il protocollo ARP può essere dunque usato per la risoluzione di un indirizzo di rete arbitrario (non solo IP) con un indirizzo hardware arbitrario.
Nella pratica, ARP viene utilizzato principalmente per assegnare indirizzi IP con indirizzi Ethernet (IEEE 802.3) o wireless LAN (IEEE 802.11)

ARP: formato dei messaggi
HARDWARE ADDRESS TYPE: campo da 16-bit che specifica il tipo di indirizzo hardware utilizzato, n caso di Ethernet, tale valore è pari a 1
PROTOCOL ADDRESS TYPE: campo da 16-bit che specifica il tipo di indirizzo del protocollo utilizzato, in caso di IP (versione 4) il valore e 0x0800
HADDR LEN: intero a 8-bit che specifica la dimensione in byte dell'indirizzo hardware, in caso di Ethernet, tale valore è pari a 6, PADDR LEN: intero a 8-bit che specifica la dimensione in byte dell'indirizzo del protocollo, in caso di IP (versione 4) il valore è 4
OPERATION: campo a 16-bit che specifica se il messaggio è una Request (valore pari a 1) o una Response (valore pari a 2), SENDER HADDR = indirizzo hardware della sorgente (lunghezza pari a HADDR LEN), SENDER PADDR= indirizzo del protocollo della sorgente (lunghezza pari a PADDR LEN), TARGET HADDR = indirizzo hardware del target (lunghezza pari a HADDR LEN), TARGET PADDR = indirizzo del protocollo del target (lunghezza pari a PADDR LEN)

ARP: formato dei messaggi: Un messaggio ARP contiene i campi per due associazioni di indirizzo, una per la sorgente, l'altro per la destinazione, denominata target. Quando viene inviata una richiesta: la sorgente non conosce l'indirizzo hardware della destinazione
• il campo TARGET HADDR in una richiesta ARP è formato da zeri, Quando viene inviata una risposta, --il target si riferisce all'host che aveva originato la richiesta, e quindi non serve a nulla, -- l'inclusione del campo target deriva da versioni precedenti di ARP ed è sopravvissuta

ARP/Trasporto dei messaggi: Quando viaggiano su una rete fisica, i messaggi ARP vengono racchiusi in una trama di livello data link (es., Ethernet), Il messaggio ARP viene quindi considerato come dei dati trasportati dal livello 2, il livello di rete non fa il processing dei messaggi di ARP
Nell'header della trama esiste un campo type che indica il tipo di trama trasportata, Per Ethernet il valore 0x806 denota i messaggi ARP, La sorgente deve assegnare il valore opportuno a tale campo prima di inviare la trama, Un host deve esaminare sempre il camp type di ciascuna trama ricevuta, Il campo type assume lo stesso valore sia che si tratti di richieste ARP che di risposte ARP – La destinazione, una volta determinato che si tratta di un messaggio ARP, andrà a vedere il campo OPERATION del messaggio per determinare se si tratta di una richiesta o di una risposta

ARP Caching e Processing dei Messaggi: Inviare una richiesta ARP per ciascun datagramma è inefficiente, Tre trame attraversano la rete per ciascun datagramma • richiesta ARP, risposta ARP, e la trama con i dati, Nella maggior parte dei casi, la comunicazione tra host avviene usando una sequenza di pacchetti, Per ridurre il traffico di rete Il software ARP estrae e salva le informazioni delle risposte ARP • in modo da poterle utilizzare anche in futuro – Il software non mantiene tali informazioni per sempre
• le mantiene in memoria in una tabella, ARP gestisce la tabella come una cache – un associazione (tra indirizzo IP e MAC) viene aggiornata quando si riceve una risposta

– se la tabella ha raggiunto la sua dimensione massima e arriva una nuova informazione, si procede alla rimozione delle informazioni più vecchie– se un'informazione non e stata aggiornata per molto tempo, viene rimossa

Architettura di riferimento delle WLAN: Station (STA) = Terminale con capacità di accesso al mezzo wireless, Basic Service Set (BSS): Insieme di terminali che usano le stesse frequenze, Access Point: Stazione integrata sia nella WLAN che nel Distribution System, Portal: Bridge verso altre reti (wired), Distribution System: Rete di interconnessione per formare un unica rete logica (ESS: Extended Service Set) partendo da diverse BSS

RTS/CTS: Scopo: risolvere il problema del terminale nascosto, La sorgente invia una trama RTS (Request To Send) dopo aver percepito il canale libero per un intervallo pari a DIFS, Il ricevente risponde con una trama CTS (Clear To Send) dopo un intervallo SIFS, I dati possono essere trasmessi, RTS/CTS vengono usati per riservare il canale per la trasmissione dei dati, in modo tale che le eventuali collisioni, possano avvenire solo tra i messaggi di controllo

Terminale nascosto: Il segnale generato dalle stazioni (o dall'access point) è percepibile solo fino ad una certa distanza. La distanza dipende dalla potenza di emissione del segnale e quando il segnale è troppo debole non è possibile ricostruirlo. Ci sono particolari disposizioni spaziali per cui il segnale emesso da una stazione può essere percepito solo da un sottoinsieme di altre stazioni.
Nasce così il problema del terminale nascosto.
Ad esempio date 3 stazioni (a-b-c) come può C sapere quanto tempo deve aspettare prima di poter tentare una trasmissione? La stazione A include la lunghezza dei dati da trasmettere nella trama RTS, La stazione B include tale informazione nella trama CTS, La stazione C, quando ascolta il canale e riceve la trama CTS, riceve anche la durata della trasmissione e calcola per quanto tempo inibire la trasmissione.
Per risolvere il problema è stato adottato l'RTS/CTS. Gli RTS/CTS vengono usati per riservare il canale per la trasmissione dei dati, in modo tale che le eventuali collisioni possano avvenire solo tra i messaggi di controllo. Nell' RTS/CTS la sorgente invia una trama RTS (Request To Send) dopo aver percepito il canale libero per un intervallo pari a DIFS,
Il ricevente risponde con una trama CTS (Clear To Send) dopo un intervallo SIFS permettendo ora ai dati di essere trasmessi.

Network Allocation Vector (NAV): Nello standard 802.11, l'ascolto del canale è sia fisico che virtuale. Se una delle due funzionalità indica che il mezzo è occupato, allora 802.11 considera il canale occupato, l'ascolto virtuale del canale è fornito dal NAV (Network Allocation Vector)
La maggior parte delle trame 802.11 includono il campo di lunghezza della trama, I nodi che percepiscono le trame, impostano il NAV al tempo in cui si aspettano che il mezzo sia libero
Se il NAV > 0, il mezzo è considerato occupato.

Indirizzi privati (vs pubblici): IETF ha definito alcuni range di indirizzi all'interno dello spazio di indirizzamento IP da utilizzare solamente in ambito privato

– private addresses o non-routable addresses
– ogni volta che un router pubblico riceve un pacchetto destinato ad un indirizzo IP privato, viene segnalato un errore

Indirizzi privati: ambito di impiego
La carenza di indirizzi IP ed il costo degli archi di indirizzamento sono alla base dell'utilizzo degli indirizzi privati.
– le reti con un solo punto di connessione alla Big Internet possono utilizzare l'indirizzamento privato.
Indirizzi privati: instradamento (1)
E necessario introdurre un'ulteriore funzionalità sul bordo tra privato/pubblico per permettere di ricevere i pacchetti all'interno della rete privata
L'indirizzo IP pubblico è un indirizzo IP visibile e raggiungibile da tutti gli host della rete Internet. Viene utilizzato per la comunicazione tra dispositivi non appartenenti alla stessa rete locale e per accedere a Internet.

Le porte note: Le porte note sono le porte TCP e UDP nell'intervallo 0-1023 e sono assegnate a specifici servizi dalla IANA. Sono porte associate ad applicazioni largamente utilizzate (posta elettronica, web, ftp, ...), ad esempio la porta 80 identifica l'applicazione web.

RTO e RTT: Il timeout (RTO " Retransimission Time Out) indica il tempo entro il quale la sorgente si aspetta di ricevere il riscontro (ack) – nel caso in cui il riscontro non arrivi, la sorgente procede alla ritrasmissione. l' RTO non può essere un valore statico predefinito – il tempo di percorrenza sperimentato dai segmenti è variabile e dipende
• dalla distanza tra sorgente e destinazione
• dalle condizioni della rete
• dalla disponibilità della destinazione
–ad esempio: un fattorino che deve consegnare un pacco in città !
l' RTO deve dunque essere calcolato dinamicamente di volta in volta
– durante la fase di instaurazione della connessione
– durante la trasmissione dei dati !
Il calcolo dell' RTO si basa sulla misura del RTT (Round Trip Time)
– RTT: intervallo di tempo tra l'invio di un segmento e la ricezione del riscontro di quel segmento

STIMA RTT----------
SRTTattuale = (α*SRTTprecedente) + ((1-α)*RTTistantaneo)
dove:
RTTistantane= misura di RTT sull' ultimo segmento;
SRTTprecedente=stima precedente del valore medio di RTT
SRTTattuale=stima attuale del valore medio di RTT
α=coefficiente di peso (0 1)
- Poiché RTT può variare anche molto in base alle condizioni della rete, il valore di RTT (SRTT, Smoothed RTT) utilizzato per il calcolo di RTO risulta una stima del valor medio di RTT sperimentato dai diversi segmenti

STIMA RTO----------
RTO= β * SRTT
dove:
β = delay variance factor (tipicamente 2)
- La sorgente, dunque, attende fino a 2 volte il RTT medio (SRTT) prima di considerare il segmento perso e ritrasmetterlo

CSMA (Carrier Sense Multiple Access) : Ambito LAN: le stazioni possono monitorare lo stato del canale di trasmissione (ritardi bassi)
Le stazioni sono in grado di “ascoltare” il canale prima di iniziare a trasmettere per verificare se c’è una trasmissione in corso
Algoritmo – se il canale è libero, si trasmette – se è occupato, sono possibili diverse varianti
• non-persistent – rimanda la trasmissione ad un nuovo istante, scelto in modo casuale
• persistent – nel momento in cui si libera il canale, la stazione inizia a trasmettere
– se c’è collisione, come in ALOHA, si attende un tempo casuale e poi si cerca di ritrasmettere .

CSMA: modalità p-persistent
Il tempo viene suddiviso in intervalli – la lunghezza degli intervalli è uguale alal periodo di vulnerabilità • round trip propagation delay 2τ
Algoritmo 1. ascolta il canale • se il canale è libero – si trasmette con probabilità p; – se si è deciso di trasmettere, si passa al punto 2 – se non si è deciso di trasmettere, si attende un intervallo di tempo e si torna al punto 1 • se è occupato, si attende un intervallo di tempo e si torna al punto 1 2. se c’è collisione • si attende un tempo casuale e poi si torna al punto 1

CSMA con Collision Detection (CSMA-CD)
è un Miglioramento rispetto al csma – se la stazione che sta trasmettendo rileva la collisione, interrompe immediatamente
In questo modo, una volta rilevata collisione, non si spreca tempo a trasmettere trame già corrotte ed Inoltre, per far sentire a tutte le stazioni che vi è stata collisione, si trasmette una particolare sequenza, detta di jamming.
L'algoritmo è dunque il seguente:
-L'adattatore sistema il frame da trasmettere in un buffer;
-Se il canale è inattivo (idle), cioè non si rilevano altri pacchetti trasmessi da altre stazioni, si attende un tempo di 96 bit-time e si procede alla trasmissione, se invece è occupato (busy) si attende che il canale torni libero prima di ritrasmettere;
-Durante l'intera trasmissione l'adattatore monitora la rete (è questo il vero e proprio Collision Detection): se non riceve segnali da altri adattatori considera il frame spedito. Il segnale che valuta l'eventuale collisione o meno si ricava confrontando il segnale ricevuto con quello trasmesso: se i due differiscono è avvenuta una collisione;
-Se l'adattatore riceve, durante una trasmissione, un segnale da un altro adattatore, arresta la trasmissione e trasmette un segnale di disturbo (jamming signal) di 32 bit che avverte le altre stazioni dell'avvenuta collisione bloccandone la contemporanea trasmissione. Questo passaggio è necessario perché sulle lunghe distanze il segnale potrebbe essere attenuato a

tal punto da non permettere alle altre stazioni di rilevarlo, generando quindi un inconsapevole collisione;
-Dopo aver abortito la trasmissione le stazioni trasmittenti applicano ciascuna un algoritmo di subentro attendendo in maniera esponenziale randomizzata il tempo per la ritrasmissione (algoritmo di backoff esponenziale binario).

NAT (Network Address Translation): funzionalità introdotta per risolvere i problemi di instradamento tra una rete ad indirizzamento privato ed una rete ad indirizzamento pubblico
-Al router di confine tra privato e pubblico viene assegnato un indirizzo pubblico sull'interfaccia verso la rete esterna.
Al router di bordo (privato/pubblico) viene assegnata la funzionalità di Network Address Translation
– Il NAT traduce l'indirizzo IP dei datagrammi uscenti ed entranti sostituendo
• l'indirizzo sorgente di ogni pacchetto uscente con il proprio indirizzo pubblico
• l'indirizzo destinazione di ogni pacchetto entrante con l'indirizzo privato dell'host corretto
• l'indirizzo sorgente di ogni pacchetto uscente con il proprio indirizzo pubblico

NATT (Network Address Translation Table): Il router NAT mantiene al suo interno una tabella di record con il mapping tra indirizzo privato sorgente della comunicazione ed indirizzo pubblico destinazione della comunicazione.
Metodi di aggiornamento della NAT Table:
– Configurazione manuale
• il gestore della rete configura in modo statico i record della NAT Table
– Datagrammi uscenti
• i record vengono creati in modo dinamico ogni volta che un pacchetto verso una data destinazione attraversa il NAT
• cancellati con meccanismo di timeout
Configurazione manuale--- Vantaggi: Possibilità permanente di pacchetti in ingresso ed in uscita, Svantaggi: Record statici
Datagrammi uscenti--- Vantaggi: Record dinamici Svantaggi: Non permettono l'attivazione di una comunicazione dall'esterno.
Il NAT basato unicamente sull'indirizzo non permette a differenti host privati di connettersi contemporaneamente allo stesso host pubblico.

Dominio di collisione/broadcast:
Dominio di collisione – parte di rete per cui, se due stazioni trasmettono dati contemporaneamente, il segnale ricevuto dalle stazioni risulta danneggiato
Dominio di broadcast (detto anche Segmento data-link) – parte di rete raggiunta da una trama con indirizzo broadcast (a livello 2)
Stazioni appartenenti alla medesima rete di livello 2 condividono lo stesso dominio di broadcast – gli apparati che estendo le LAN possono solo influire sul dominio di collisione

Bit Stuffing:
-Ogni trama può includere un numero arbitrario di bit
-Ogni trama inizia e termina con uno speciale pattern di bit, 01111110, chiamato byte di flag

-Problema: come comportarsi se la trama contiene al suo interno il pattern di bit usato per il byte di flag?
Soluzione:
– Se la sorgente incontra 5 bit “1” consecutivi, aggiunge uno “0”
• **bit stuffing**
es. la sequenza “011111x” è trasmessa come “0111110x”, dove “x” e’ il bit successivo, puo’ essere sia “0” che “1”
– Se la destinazione incontra 5 bit “1” consecutivi, toglie uno “0”
es. la sequenza “0111110x” è modificata in “011111x”