Caratteristiche generali delle RETI

- DISTANZE
- SCAMBIO INFORMAZIONI
- M. TRASMISSIVO
- COMMUTAZIONE
- VELOCITÀ RETE
- GESTIONE RISORSE
- RESISTENZA GUASTI
- STANDARD

DISTANZA:

<5 cm => chip hardware => NO RETE DI CALC.
↳ non mi interessa come si comunica nel chip

<50 cm => motherboard PC => NO RETE

RETE 1°
<5 Km => interconnessione tra risorse di calcolo in uno stesso edificio o edificio vicino
RETE LAN - LOCAL AREA NETWORK

RETE 2°
>5 Km => interconnessione tra risorse di calcolo in una vasta area geografica
RETE WAN - WIDE AREA NETWORK

SCAMBIO INFORMAZIONI:

- tramite memoria condivisa e in parallelo
  ↳ non ci interessa questo tipo di trasmissione
- tramite scambio di messaggi in serie
  ↳ CASO DI STUDIO: costituisce una rete di computer

MEZZO TRASMISSIVO

- NON CI INTERESSA LA PROPAGAZIONE E LE CARATTERISTICHE DEI MEZZI TRASM.
- CAVO COASSIALE
- CAVO TELEFONICO
- FIBRE OTTICHE
- ONDE RADIO
- [illegible]
- RFC: request for commerce, standard numerato

COMMUTAZIONE DI CIRCUITO

- « tecniche elettroniche e metodologie utilizzate per far dialogare tra loro calcolatori connessi in reti »

Importato dalle reti telefoniche da cui derivano anche le commutazione:
- di circuito
- di pacchetto
  - ↳ di datagramma
  - ↳ circuito virtuale

→ n^2 fili per connettere n. computer

→ si usano centraline intermedie con la possibilità di chiudere n interruttori per arrivare al pc. destinazione. Durante l'intera operazione di scambio dati la linea è dedicata ai soli 2 pc. ⇒ NON si usa per internet, perché terrei impegnata una linea solo per 2 pc, quando potrebbe servire per molte altre connessione

COMMUTAZIONE DI PACCHETTO A DATAGRAMMA

- NON ATTRIBUISCE LINEA PER 2 SOLI PC
- divido il messaggio da scambiare in più pacchetti che poi saranno uniti insieme nel punto di destinazione. I pacchetti sono distribuiti sui fili con in base al livello di carico delle linee. Fra un pacchetto e il successivo di uno stesso messaggio ci si può trovare un pacchetto di un messaggio diverso. ⇒ linee condivise

- PROBLEMA: RIORDINAMENTO LENTO NEL PUNTO DI ARRIVO

COMMUTAZIONE DI PACCHETTO A CIRCUITO VIRTUALE

I pacchetti di uno stesso messaggio passano su una stessa linea in serie.

-> Fra i pacchetti di uno stesso messaggio possono essere inviati pacchetti di altri messaggi (di altri utenti)

- STABILISCO a priori la strada che tutti i pacchetti devono seguire.

- Scelta strada da far percorrere ai pacchetti per via della moltitudine dei gestori.

-> Ora si usa quello a datagramma
↳ per via dei diversi gestori

Velocità: (BANDA DI COMUNICAZIONE)

LAN: 10 Mb/s -> 100 Gb/sec

WAN: 64 Kb/sec - 10 Gb/sec - 50 Gb/s - 100 Gb/s - 200/400 Gb/s

VELOCITÀ ROMA-TOKYO DIPENDE DA:

- DISTANZA
- VELOCITÀ LUCE

DELAY o RITARDO: cambia il ritmo con cui si caricano dati sul canale (BANDA)

Si compra + banda molto facilmente ma non si compra meno delay ph dipende dalla velocità della luce e sotto una soglia non si può andare, e anche la distanza tra server e client

GESTIONE RISORSE

- Insieme di risorse interconnesse => Rete di calcolatori

- GESTIONE RISORSA:
  - VERIFICA ACCESSO
  - SEQUENZIAMENTO
  - ESECUZIONE DELLE OPERAZIONI DISP.

- Per ogni risorsa posso avere:

- 1 gestore "autocrate"
- + gestori per la stessa risorsa "multilaterali"
  - -> Partizionate: 1 solo processo gestisce 1 attività
  - -> Successive: tutte le attività sono effettuate a turno da tutti i processi
  - -> REPLICATA: tutti i processi partecipano a tutte le attività
    - ↳ Se cooperano tutti i processi alla pari nella gestione e interagiscono tramite esplicito consenso => Democratica

Avere più processi "gestori" che sanno fare la stessa cosa allora se si rompe un gestore ~~allo~~ il processo "attività" si interrompe.

=> Resistenza ai guasti

più ho processi in grado di lavorare sulle medesime risorse, ho + resistenza ai guasti.

Standard

- Possono essere imposti dalla casa produttrice
- Imposti da organizzazioni internazionali
- Informatico
- ambiente delle telecomunicazione

30/09/21

MODELLO ISO-OSI

RITMO: BANDA

LATENZA: T. necessario ad un bit per propagarsi

Architetture astratte

- Iso: international standard organization
- Osi: organization sistems interconnection

STRATI DEL MODELLO

- MODELLO: standard

-> Descrive una rete di calcolatori sfruttando dei livelli.

APPLICAZIONE

PRESENTAZIONE

SESSIONE

TRASPORTO

RETE

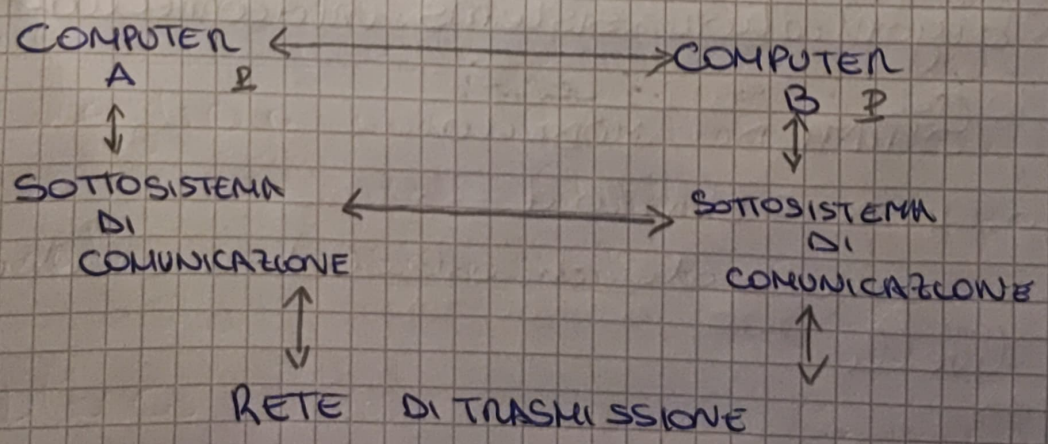
DATA LINK

FISICO

Esempi di architetture stratificate

- SISTEMA OPERATIVO
- Ogni strato fornisce a quello immediatamente superiore dei servizi
- Più si sale più il livello è astratto, man mano che si aumenta di astrazione più la macchina è utilizzabile
- Si cambia tecnologia senza cambiare interfacce

ARCHITETTURA A STRATI DI RETI



Tipicamente sono i processi che comunicano tra loro

LIVELLO ISO-OSI

APPLICAZIONE
PRESENTAZIONE
SESSIONE
TRASPORTO
RETE
DATA LINK
FISICO

- si possono avere moltissimi strati, il 7. ISO-OSI sono un compromesso in modo tale che ogni livello passa info al suo superiore e ognuno abbia un compito specifico.
- troppi livelli sono complessi da mantenere

STRATI ISO-OSI AGLI ESTREMI E NEI SISTEMI INTERMEDI

Applicazione $\xleftrightarrow{\text{mex}}$ applicazione

↳ software livello 7.

↳ comunicazione di livello 7

poiché questo strato non può e non sa spedire direttamente il messaggio e lo passa tramite un'interfaccia al livello direttamente sottostante che ha il compito di tradurre il mex in un linguaggio adatto ad essere inviato sulla rete. (codifiche diverse)

Applicazione $\xrightarrow{\text{mex}}$ App.

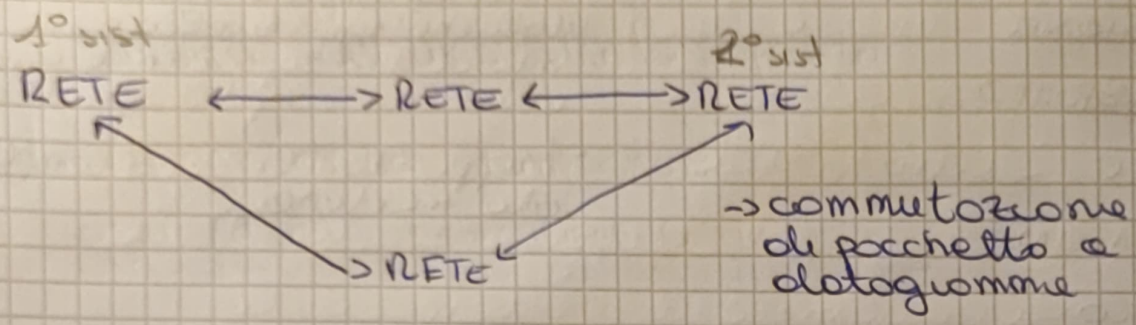
↓

PRESENTAZIONE $\xrightarrow{?x?x}$ PRES.

Tuttavia non lo sa passare nemmeno lui e lo cede in carico al blocco 'SESSIONE' che ha il compito di sincronizzare la speci-ficazione del mex. (divide il mex in più parti così che anche con prob. di ricezione alcune parti già

Lo crea anch'esso al livello subito sotto 'TRASPORTO' che crea pacchetti da spedire. Tra i diversi software il livello di TRASPORTO deve avere lo stesso linguaggio

TRASPORTO $\xrightarrow{\square\circ\times \quad \times\square\circ \quad \times\times\circ}$ TRASPORTO



- Gli strati 4-5-6 (quelli verso alto) si trovano solo nei COMPUTER. Quelli sotto si trovano nei sistemi intermedi.
- Il livello di rete si organizza con i sistemi intermedi per portare i pacchetti a "cose". Ciò significa che il PC ignora tutti i sistemi intermedi che trasportano e inviano i mex.
- Il blocco di rete per inviare i dati <u>deve conoscere</u> la struttura di rete per scegliere a quale sistema intermedio a cui inviare i dati. Struttura e mappa di rete sono <u>sconosciuti</u> ai livelli superiore
- Il blocco di rete individua la strada che il pacchetto deve fare ma non la sa spedire => chiede al blocco sotto

Il livello 1 'FISICO' vede il livello 1 del blocco a cui deve arrivare il pacchetto e trasforma il pacchetto in una sequenza di bit che poi invia al sistema intermedio successivo

DATA LINK: si occupa di trasmissione e cerca di codificare i dati dal livello fisico in modo tali che a destinazione arrivino pacchetti corretti privi di errore.

- Se cambio qualcosa di dialogo al livello di 'applicazione', quindi cambio protocollo, questo non implica un cambiamento degli altri strati. (vale questa procedura per ogni blocco del modello)

- Se cambio il blocco di rete su un pc (determina un cambio di protocollo) devo cambiarla su tutti i pc collegati in rete => determina la lunghezza di cambiamento molto più lento rispetto al cambiamento degli altri blocco

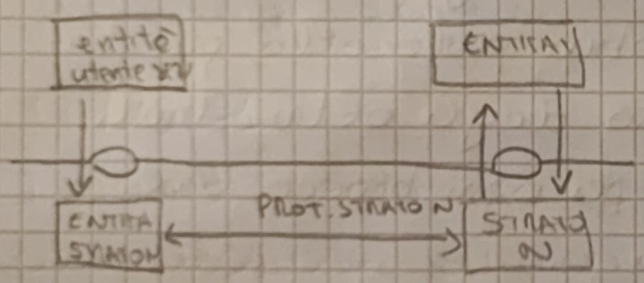
PROTOCOLLO: ... linguaggio tra i vari blocchi

ENTITÀ O SAP

In uno strato si possono individuare una o più entità (software che adempie alcuni compiti)

Il punto logico in cui uno strato offre il servizio ad uno strato superiore

· Indipendenza Funzionale:
ogni strato è indipendente dagli altri



· Se n+1 chiede un qualcosa al livello n ho una richiesta di servizi

· Se n dice a n+1 l'informazione dove trovarla n emette un'indicazione a n+1

· N+1 emette una risposta verso n così che non si riemettono le stesse informazione e contestualmente n emette una conferma a n+1

I processi applicativi comunicano nell'architettura ISO-OSI

FUNZIONE DI LIVELLO

APPLICAZIONE — AH | DATI

PRESENTAZIONE — PH | AH | DATI

SESSIONE

TRASPORTO

RETE

DATA LINK

FISICO

- il livello di applicazione aggiunge dei dati in modo tale che si possono eseguire le istruzioni AH "application header"
- Presentazione deve inserire dati su quello che deve fare ... codifica di livello PH "presentation header"
- Ogni livello di conseguenza aggiunge dei dati che gli servono per eseguire il suo lavoro
  => non si può avere un numero illimitato di STRATI.
- Alla fine lo strato fisico trasforma tutto in una sequenza di bit

Quando si passa l'informazione ad un'altra macchina man mano che usa gli strati viene 'ripulito' da tutte le intestazioni (H) e arriva all'ultimo livello senza H.

I dati generati da un protocollo di livello n sono detti n-pdu = n-protocol data unit

PDU = PAYLOAD + HEADER

Di volta in volta si aggiunge ad un payload "ossia i dati già esistenti" provenienti da uno strato precedente e ci aggiunge un suo Header ossia un'intestazione

04/10/21

I pacchetti sono tipici del livello 3 con la commutazione di pacchetto e poi pervade tutta la rete

. Nell' ISO-OSI PACCHETTO = PDU

. Al livello di trasporto il pacchetto di dati iniziale viene scomposto e trasmesso come piccoli pacchetti.

Le intestazioni possono essere anche sviluppisi ma al momento in cui i pacchetti arrivano queste vengono ricomposte

Sono ammessi da soli all'interno dei 7 livelli.

STRATO FISICO

- Si interfaccia direttamente con il mezzo di trasmissione e offre i servizi al livello superiore indipendentemente dal mezzo di comun. sottostante

- SERVIZI OFFERTI:
  - comunicazione di singoli bit tra sistemi adiacenti (trasmette al livello 1 del sistema intermedio)
  - consegna dei bit in sequenza

STRATO DATA-LINK

Fronteggia errori occorsi al livello 1 in modo tale da individuare e correggere errori

SERVIZI OFFERTI

- TRASMISSIONE DI PACCHETTI
  ↳ usa due code per inserire dati in ricezione e trasmissione

STRATO RETE

- CONOSCE l'infrastruttura della rete e porta dei pacchetti da un punto A ad un punto B (no adiacenti)
- Si effettua qui la commutazione

SERVIZI E PROTOCOLLI CONNESSI E NON

- Per tutti i livelli >1 si possono avere 2 modalità operative, che danno luogo a 2 servizi e 2 protocolli: CONNESSI E NON

SERVIZIO CONNESSO: ci si mette d'accordo per effettuare qualcosa insieme come una forma di dialogo che al termine del servizio si ha una risposta ed un messaggio di chiusura. (TELEFONATA)

SERVIZIO NON CONNESSO: COMUNICAZIONE SENZA incontri sullo stato e sul termine del servizio. INVIO messaggi fuori dal contesto di un dialogo (MAIL)

DA QUESTI SERVIZI HO 2 PROTOCOLLI:

NON CONNESSO: più efficiente (+veloce) ma meno affidabile

CONNESSO: meno efficiente (-veloce) ma più affidabile

STRATO DI TRASPORTO

- Ho un pacchetto da inviare a destinazione e ha il compito di gestire eventuali problemi del livello 3 (perdite dati)

- OFFERTI AL LIVELLO 5:
  - INSTAURA CONNESSIONI
  - TRASFERIMENTO affidabile di dati
  - Da terminale a terminale

TECNICHE USATE IN PROTOCOLLI DI LIVELLO 2, 3 e 4

- Si usano riscontri (AKNOWLEDGE)
- Tecniche di controllo a finestre

LIVELLO	LAN	WAN
2	NO	SI E NO
3	NO	NO
4	SI	SI

STRATI + VICINI ALL'APPLICAZIONE (5, 6, 7)

- Sessione: sincronizzazione tra 2 processi
- PRESENTAZIONE: scambio messaggi indipendente dalla sintassi

APPLICAZIONI: Accesso alla rete per un processo applicativo

Spesso si usano contemporaneamente

Tutti i servizi di n-esimo livello insistono sempre l'infrastruttura di livello più basso (1).

- A livello applicativo l'architettura di rete è molto duttile perchè la connessione all'utente sembra ~~uguale~~ stabile anche se ho avuto delle commutazioni etra varie connessioni (WI-FI e DATI) (continuità servizio)
  => sviluppato per non far vedere ciò che succede a livelli sotto

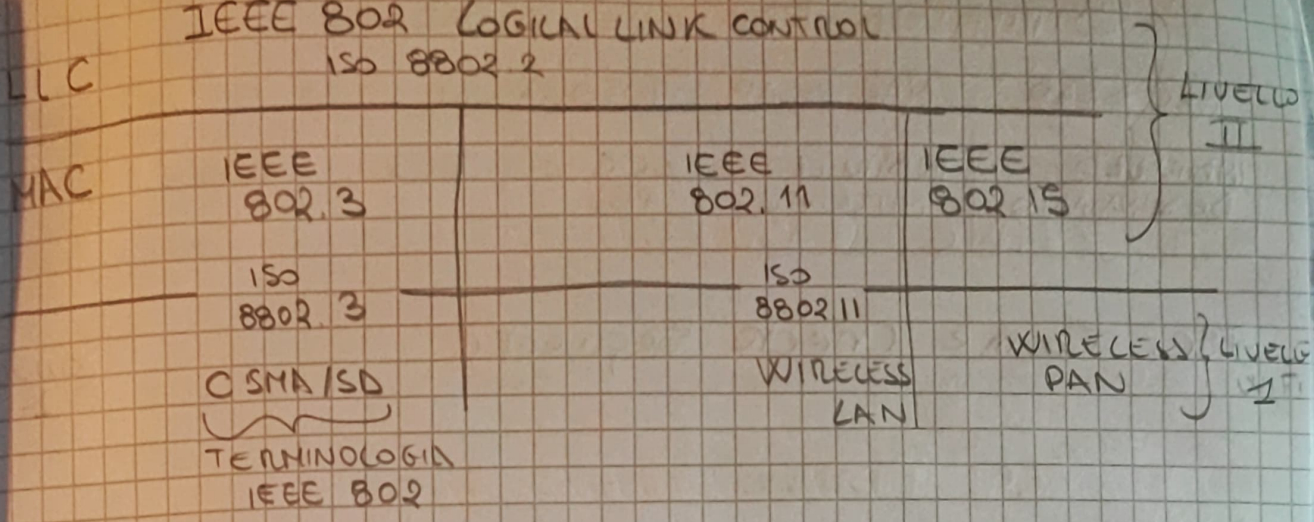
IEEE 802 : livello ISO-OSI 1 e 2

PROGETTO IEEE 802

- Si vuole definire un insieme di standard di livello fisico (1) e livello data-link (2) per:
  - RETI LOCALI (LAN)
  - RETI PERSONALI (PAN)
  - RETI METROPOLITANE (MAN)
- Si occupa di pacchetti di lunghezza variabile

802.1 higher layer and management
802.2 logical link Control
802.3 Carrier sense, multiple access, collision detection
- 802.3u fast ethernet
- 802.3z gigabit ethernet
- 802.3ae, 802.3ak 10 gb

802.11 WIRELESS LAN
802.15 WIRELESS PAN



Lo strato LLC è comune per ogni LAN

SOTTOLIVELLO MAC - MEDIA ACCESS CONTROL

È specifico per ogni lan. Assume che tra dispositivi non esiste un filo di connessione, ma si ha un mezzo trasmissivo comune tra loro.

PROBLEMI:

2 dispositivi condividono dati allo stesso tempo
↳ si verifica a priori se il canale (trasmissione) di trasmissione a priori così che non si verifica sovrapposizione.
Si usa per questo un algoritmo distribuito su ogni macchina

In ricezione per distinguere mittente e destinatario (infatti essendo la LAN CONDIVISA TUTTI HANNO TUTTO) si usano indirizzi MAC.

MAC PUNTO PUNTO: un computer ad un altro
PUNTO GRUPPO: un computer a gruppo di PC
BROADCAST: un PC a tutti i PC

Struttura PDU livello MAC

MAC DSAP	MAC - SSAP	INFO (PAYLOD)	
INDIRIZZO DESTINATARIO	INDIRIZZO MITTENTE	LLC PDU	FCS

FSC: Frame Check Sequence

Ciò che in ISO-OSI è SAP (service access point) in IEEE è indirizzo, pk quando invio un dato al livello ~~sei~~ inferiore devo dire chi lo deve ricevere.

I MAC sono di 6 byte

Sono unici a livello mondiale:

- 3 byte sono assegnati al costruttore
- 3 byte progressivi (schede di rete)

Sono unici al livello mondiale per facilitare la gestione delle reti

I 3 byte successivi possono essere riassegnati

07/10/21

Guardare a LLC si ha che ciò che gli sta sopra vede la struttura sottostante a LLC come singolo

LLC - DSAP	LLC-SSAP	CONTROL	INFO (PAYLOAD)
		tipo PDU	PDU LIV. 3

- LLC-dsap e LLC-SSAP servono a denotare i protocolli di livello 3 a cui sono destinati i pacchetti
- consente la convivenza di protocolli diversi di livello 3 sulla stessa lan sulla stessa macchina

→ sono + processi che gestiscono protocolli di livello 3

- L'indirizzo llc specifica anche a che PDU è destinato quel protocollo (cosa che non fa il mac)

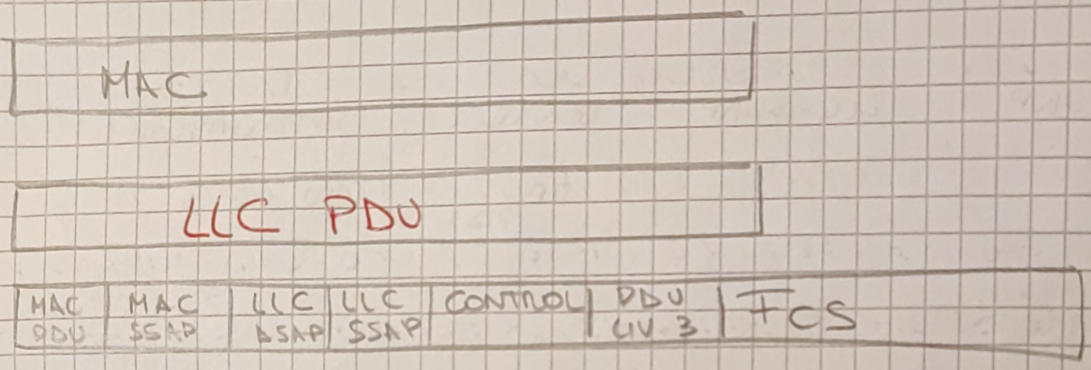
INDIRIZZI LLC

- Ho alcuni protocolli riconosciuti da llc. (IP non è riconosciuto)
- Se in LLC SAP trovo 2 valore AA (in esadecimale) allora quel protocollo non è standardizzato

(SNAP-PDU)

- in caso di llc non riconosciuti si aggiunge un campo di 5 byte chiamato PROTOCOL IDENTIFIER
  ↳ che identifica univocamente il destinatario

MAC PDU



nella mac PDU ho i campi di llc

CSMA/CD
e rete ethernet

- metodo di accesso al canale trasmissivo usato da ethernet nelle prime versioni IEE 802.3

→ Primi anni del 70': mizio Dix

- Rete locali a mezzo trasmissivo unico, poi perso perchè si evoluzione il sistema di comunic.

- Obbiettivo: Equivalenza tra nodi

- Non prevista: Priorità o garanzia su tempo di risposta

CARRIER SENSE: una stazione che deve trasmettere prima di farlo verifica che il mezzo sia libero

MULTIPLE ACCESS: se il canale è libero tutti possono trasmettere, quindi ho collisione

COLLISION DETECTION: durante la trasmissione la stz che trasmette comunque ascolta, così può rilevare una collisione

- Gira su tutti i "computer" con CSMA/CD

→ Se ho una collisione rilevata da una stazione trasmittente si introduce una sequenza di JAMMING (bit in sequenza di lungh casuale, con l'unico scopo di invalidare FCS)

- Se ho jamming le altre stazioni scartano pacchetto

- La trasmittente prova per un numero di volte limitato a ritrasmettere il pacchetto in istanti di tempo casuali

La stazione che invia pacchetti ascolta comunque il canale così da vedere se ci sono collisioni

ROUND TRIP DELAY: 2τ è il tempo che impiega il segnale a percorrere da un estremo all'altro il mezzo moltiplicato per 2.

VINCOLI DI CSMA/CD

-> Se una trasmissione di una stazione non ha almeno una durata pari al round trip delay della seconda stazione (che anch'essa vuole trasmettere) affinchè la seconda stz rilevi la collisione e possa emettere la seq. di jamming.

-> se la trasmissione è troppo corta la stazione s_1 non se ne accorgerà mai.

-> Per impedire questo problema dovrei costringere s_1 a trasmettere + bit in modo tale che il tempo di arrivo del segnale (1 bit di s_1) all'altra stazione permetta di riconoscere la collisione

Data la velocità di propagazione (2/3 v. luce) e stabilita la velocità di trasmissioni devo scegliere:

- Dimensione minima dei pacchetti
- Massima lunghezza di rete

Infatti maggiore è l'estensione della rete maggiore dovrà essere la lunghezza minima del pacchetto, così da permettere al bit trasmesso di ritornare indietro (2τ)

- 10 Mbs 2/3 vel luce nel vuoto

devo trasmettere abbastanza bit per arrivare al tempo di 2τ (infatti più la velocità di banda è ampia maggiore sarà il n. di bit da trasmettere)

il t. di immissione del pacchetto + corto è 10^7

quindi $512/10^7 = 512 \cdot 10^{-7}$ sec $\approx 51,2$ µs

Round trip è $2\tau = 51,2$ µs

Se divido 512 per $10^7 \cdot 2 \Rightarrow \frac{512}{2 \cdot 10^7} = 5$ Km

Algoritmo truncated binary exponential backoff (calcolo tempo random di attesa)

FRAME = Pacchetto

Suppongo di stare nel 1° tentativo di ritrasmissione allora ho:

MAX = 2 pk tentativo = 1

se tentativo ≠ 1 ho: (sempre se sono sotto al n. max di tentativi concessi)

MAX = MAX · 2

aspetto per ritrasmettere (2τ x random(0, max-1))

Se tentativo = limite backoff

· abortisco trasmissione

· Se ho tanto traffico sulla linea mi autolimito pk il tempo di attesa cresce esponenzialmente rispetto al n. di tentativi tentati

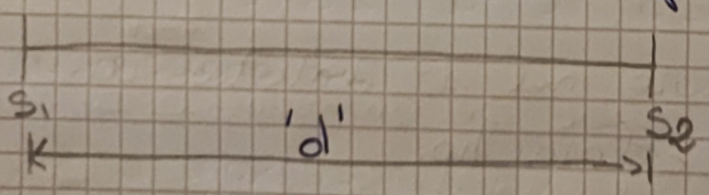
· Randomizzo unint. pk la probabilità che fra numeri interi escano numeri uguali (in str. diverse) è molto bassa rispetto alla randomizzazione su Reali. => Meno collisioni

· La randomizzazione può portare a alla ritrasmissione in poco tempo perciò moltiplico per 2τ => discretizzo il tempo

11/10/2021

ANALISI DI 1 COLLISIONE

Suppongo di avere 2 str. (S_1 e S_2) che si trovano a distanza temporale 'd'



d' è il tempo necessario per un bit per andare da S_1 a S_2.

→ S_1 inizia a trasmettere in t_1 e S_2 in t_2 con $t_1 \leq t_2$

Suppongo $d < \tau$

τ è la distanza max tra 2 stz, se la supero ho una rete troppo ampia

Ho 1 collisione tra i dati di S_1 e quelli di S_2. Per far ciò che la collisione avvenga $t_2 < (t_1 + d)$ solo così avviene, infatti se $t_2 > t_1 + d$ la collisione non avviene pk il primo bit di S_2 non partirebbe nemmeno poiché S_1 vede prima di inviare i pacchetti che il canale è occupato

→ Sia dunque $t_2 = t_1 + \Delta$ con $0 \leq \Delta \leq d < t$

Percezione collisione: t_{1p} e t_{2p} ($t_{1p} \geq t_{2p}$) tempi di percezione della collisione da parte di S_1 e S_2

$t_2 = t_1 + \Delta$

$t_{1p} = t_2 + d = \Delta + d + t_1$

↑ inizio trasmissione di S_2

↳ tempo per i bit di S_2 per arrivare davanti a S_1

Δ - tempo che intercorre tra t_1 e t_2

La stazione deve essere libera per far ritrasmettere la S_1 e S_2. (CARRIER SENSE)

Il canale sarà di nuovo disponibile per S_1 e S_2:

sia J - tempo trasmissione seq. di jamming

l'ultimo bit emesso da S_2 passa davanti a S_1 all'istante $t_{2p} + J + d = t_1 + J + 2d$

↳ tempo propag. ultimo bit jamming

① tempo di rilevazione collisione

② trasmetto per J tempo la seq jamming

Canale libero di fronte a S_2:

$t_{1p} + J + d = t_1 + J + 2d + \Delta$

Ora il canale è libero S_1 e S_2 potrebbero ritrasmettere.

t_{1R} e t_{2R} tempi di Ritrasmissioni S_1 e S_2

- Per ritrasmettere una stazione deve:
  - Verificare la disponibilità di canali
  - Attendere il tempo di backoff

$t_{1R} = \max(t_1 + \Delta + d + J + n2t, \underbrace{t_1 + J + 2d}_{\text{tempo x canale libero}}) = t_1 + J + d + \max(\Delta + n2t, d)$

$t_{backoff} = t_1 + \Delta + d + J + n2t$

↳ numero estratto da backoff

il tempo di ritrasmissione ~~iniz~~ è il tempo maggiore tra backoff e tempo attesa canale libero

$t_{2R} = \max(\underbrace{t_1 + d + J + m2t}, t_1 + \Delta + 2d + J) = t_1 + J + d + \max(m2t, \Delta$

backoff => tempo per percepire la collisione su t_2
m è il n. sorteggiato - + tempo jamming
+ m2t

$t_{2R} = t_1 + d$

Distanza temporale tra ritrasmissione S_1 e ritrasmissione S_2:

$t_1 + J + d + \max(\Delta + n2t, d) - t_1 + J - d - \max(\Delta + d, m2t)$

$t_{1R} = t_1 + J + d + \max(\Delta + n2t, d)$

$t_{2R} = t_1 + J + d + \max(\Delta + d, m2t)$

se i moduli di t_{1R} e t_{2R} fossero troppo vicini (differenza formula) allora avrei una nuova collisione

$|t_{1R} - t_{2R}| = |\max(\Delta + n2\tau, d) - \max(m2\tau, \Delta + d)|$

↳ dist temporale tra ritrasmissione

Δ = distanza temporale fra 2 ritrasmissione

Caso in cui $n = m = 0$ $\quad n2\tau = 0 \quad m2\tau = 0$

$d > \Delta$

$|t_{1R} - t_{2R}| = |d - (\Delta + d)| = \Delta$

↳ poichè $\Delta < d$ allora il segnale trasmesso da 1 stazione entrerà in conflitto con quello dell'altra stz. poichè la 2 stz inizia a trasmettere prima che gli arrivi l'ultimo bit di S1

NUOVA COLLISIONE

Caso $m = n > 0$ stesso numero $\neq 0$

$\tau > \Delta > d$

$|t_{1R} - t_{2R}| = |\Delta + n2\tau - n2\tau| = \Delta \quad n > 1$

NUOVA COLLISIONE

Caso $n = 0$ $m > 0$ e $m \neq n$ $\quad \max(\Delta + n2\tau, d) - \max(m2\tau, \Delta + d)$

$n = 0 \Rightarrow \max(\Delta, d)$

$\Delta < d$

$m \neq 1$

$m2\tau$

$|t_{1R} - t_{2R}| = |d - m2\tau| > \tau$

ASSENZA COLLISIONE

- [redacted] poichè il tempo risultante è maggiore di τ se S1 trasmette S2 non inizierà mai a trasmettere per carrier sense dato che riceverà il primo bit di S1 prima della sua trasm.

Caso $n > 0$ $m = 0$

$|t_{1R} - t_{2R}| = |n2\tau - d| > \tau$

ASSENZA COLLISIONE

Caso $m \neq n$ $n, m > 0$

$|t_{1R} - t_{2R}| = |\Delta + (n - m)2\tau| > \tau$

ASSENZA COLLISIONE

2 stazioni collidono se estraggono stessi numeri

IEEE 802.3 11/10/21

→ (CSMA/CD canonico)

Csma/CD: immagina il canale come coassiale unico e condiviso.

- cavi in rame twisted pair
- fibre ottiche

Tipologia: bus e stella

Velocità 10 Mb/s

4 km al max 1024 sta max

CSMA/CD => Ethernet

LIVELLO MAC

Sono diversi in base alla scheda ma hanno in comune:

Preambolo: 56 bit di sincronizzazione

SFD (start frame delimiter): inizio pacchetto (8 bit)

Destination e Source MAC ADDRESS

LENGHT: 16 bit che indicano quanto è lungo la MAC

PAYLOAD: LLC PDU

PAD: RIEMPIMENTO. Devo arrivare a 1500 byte

FCS (frame check Sequence): valore CRC calcolato, 32 bit

Impongo 1500 byte di massima trasmissione pk così non ho troppo tempo da aspettare per liberare il canale ossia dopo quanto la si può trasmettere.

Minimo n. bit: 512 + Preambolo + SFD = 576 bit

Tra un pacchetto e il successivo ho un gap così da permettere al sistema di capire quando un frame è finito.

INTER FRAME SPACE (IFS) o INTER FRAME GAP (IFG)
- 96 bit

- Numero max ritrasmissioni: 16
- lunghezza jam: 32 (96 per repeater)
- lungh. min-max pacchetto senza preambolo e SFD
  64 byte -> 1518 byte

DOMINIO DI COLLISIONE: zona della LAN in cui 2 pacchetti possono collidere.

Funzioni svolte da MAC

Riceve un pacchetto dal livello fisico, controlla che non ci siano errori (altrimenti lo scarta), lo incapsula nel pacchetto MAC. e lo riconsegna a livello fisico.

Si occupa di verificare se ho collisioni.
(si calcola in base il backoff)

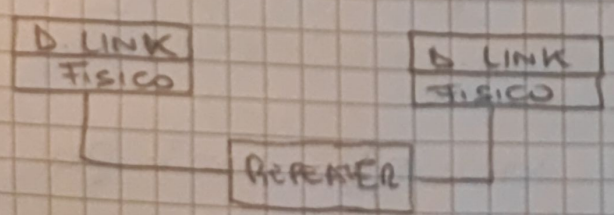
In trasmissione: genera un FCS per verificare se ho errori e verifica inoltre che questo sia uguale all'FCS ricevuto

- verifica lunghezza pacchetto
- Rimozione/inserimento preambolo
- Scarto pacchetti errati.

Ripetitore (HUB)

Ha varie porte alle quali trasmette lo stesso identico bit ricevuto dall'esterno.
(Non è store and forward: non memorizza prima di trasmettere.)
Dato che trasmette 1 a 1 i pacchetti trasmette anche le collisioni => perciò se ce ne sono troppe su una porta allora la può disabilitare

Il ripetitore è situato solo a livello fisico perciò il livello data-link non si accorge del repeater.
Per il mac il mezzo trasmissivo rimane unico e condiviso anche se ho ripetitori



LLC IEEE 802.2 LLC ISO 8802.2

MAC MAC 802.3

10 base t (RJ-45)

- twisted pair
- 2 coppie - trasmissione / ricezione hanno coppie diverse
- topologia a stella
- Ripetitori multi porta, che svolgono il ruolo di mezzo trasmissivo)

10 base F (come prima ma con fibra)

Ethernet 2.0 e Ieee 802.3

- I pacchetti di entrambi gli standard convivono.
- In Ethernet 2.0 non ho gli LLC perciò il pacchetto di livello 3 entra direttamente in MAC
- Se in campo len. ho un numero > 1500 allora quello è un pacchetto di standard Ethernet 2.0 se < 1500 ho pacchetto 802.3.
- In 802.2 - 802.3 ho dsap per sapere a chi è destinato il pacchetto
- Sopra a 1500 ossia ≥ 1501 sono byte usati per denotare tutti i protocolli liv 3 che viaggiano nel pacchetto

In Ethernet 2.0 => type al posto di length e a metti il tipo di protocollo

- Ci sono più Ethernet 2.0 che quelli 802.3 e 802.2 perchè utilizzano un pacchetto 802.3 con protocolli non standardizzati da IEEE 802 è poco efficiente in quanto bisognerebbe inserire in DSAP ('AA') inserire un Header SNAP che contenga il codice effettivo del protocollo da usare e ciò comporta un uso eccessivo di byte che poi non sarebbero + utilizzabili.

Uso perciò pacchetti ethernet 2.0 per tutti quei protocolli non standardizzati (IP)

14/10/21

BRIDGE / SWITCH (nome commerciale)

- Il bridge nasce per risolvere problemi di:
  - -> distanze imposte da 802.3
  - -> massimo numero di sist interconnessi
  - -> massimo traffico
- Il bridge ritrasmette solo i pacchetti che devono transitare sulla LAN. TRASMETTE da una porta di lan ad un altra: FUNZIONE DI FILTERING
- ↳ Memorizza i fili prima di inoltrarli: STORE & FORWARD
- Connette anche lan con mac differenti
  - -> Il bridge lavora sia con 802.3 (Ethernet) e 802.11 (WIFI) a differenza di ripeater che comunica solo con solo 802.3

- Se il pacchetto che arriva in un mac e lo devo trasmettere ad un mac diverso devo ricostruire il nuovo pacchetto conforme al nuovo mac

- Fra un mac non IEEE 802 e quelli IEE 802 devo ricostruire anche LLC

SEPARAZIONE LAN: il bridge fa passare pacchetti solo se [illegible] sono pacchetti di interlocutori si trovano nella sua partizione di LAN.

- Il pacchetto risulta visibile a macchine di arrivo e di partenza

- Si trovano sul livello 2 della pila ISO-OSI e sottolivello MAC di IEEE 802 (uso generale anche per 802.3)

- Instradamento: non lo svolgono a livello 3 ISO-OSI ma al livello 2. L'instradamento compiuto da Bridge è effettivo nel confine della rete LAN.
  → STANDARDIZZATO 802.1D

TRANSPARENT: Le macchine in rete ignorano la presenza di bridge

ENTITÀ RELAY: prende un pacchetto mac e lo fa transitare anche verso altri mac.

LEARNING.

- La tabella di instradamento (tabella che ci dice dove si trova una macchina e in che porta è collegata) (filtering database) e viene costruita con processo di LEARNING

- Quando si accende il budge questo non sa in che porta è collegato il destinatario, perciò lo fa passare su tutta la rete lan ma si "segna" il MAC address del destinatario e quando il pacchetto entra in una macchina vede dove si trova il destinatario. Ovviamente si "segna" l'indirizzo anche del mittente al momento di ricezione.

→ Una volta capita l'ubicazione delle macchine decide dove inviarli e che zone di lan farli transitare.

La tabella si trova memorizzata in RAM.

- Se sposto una macchina da una lan A a B?

- L'informazione di ubicazione la ritengo vera per un tot di tempo, qualora cambiassi posizione allora la tabella si aggiorna subito. (le macchine di fatto parlano sempre)

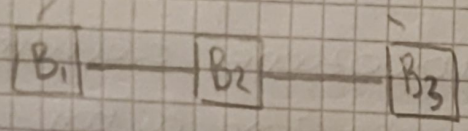
Al Budge serve leggere il pacchetto fino agli indirizzi mac di destinatario (instradamento) e di arrivo (forwarding)

Le 2 non possono lavorare insieme cioè senza mai attenderlo.

Se una macchina non invia mai pacchetti il bridge non possono sapere mai la pos.

Funziona questa tabella se la ~~tra~~ rete è ad albero ma senza ciclo

Di solito le reti sono a grafi ph ho dei cicli per motivi di affidabilità, calcolo alberi ricoprenti del grafo in modo tale che di volta in volta un solo albero sia funzionante



- senza spanning tree allora ho dei cicli ph per esempio il pacchetto inviato da B1 verso B3 arriva a B3 su due porte diverse una per B1 e una per B2.

Le macchine (bridge) collaborano tra loro e ricostruiscono uno spanning tree (albero ricoprente) con algoritmo democratico. Scambio pacchetti spanning tree tra bridge per decidere quali porte sono attive e quali no.

PRESTAZIONI

1) N max pacchetti processabili in un'unità di t.
2) tempo percorrenza bridge: latenza.
3) più ho pacchetti ho + decisioni devo prendere.

Preferibilmente deve essere FULL SPEED ossia con parametri pari al max teorico
↳ almeno deve esser pari al numero di pacchetti che gli arrivano

+ i pacchetti sono piccoli più aumenta il numero di decisioni che il bridge deve prendere.
Invio ad ogni porta il n. max di pacchetti inviabili sulla propria rete (Ex 10 Mbit/s)
→ Ex. Full speed se processa 14880 pacchetti/sec
se ho una porta a 10 mbit/s
576 (bit minimi) + 96 bit (inter packet) = 672 bit
10MB = 10000000 bit $\frac{10\,000\,000 \text{ bit/s}}{672 \text{ bit}}$ = 14880 pacchetti/sec

La latenza dipende dalla lunghezza dei pacchetti

Architettura

- Schede Asic: già so le macchine connesse ad una porta così da non sovraccaricare l'unità centrale

Architettura logica

- Ho un'entità di livello superiore che permette il calcolo di spanning tree.
- Se il pacchetto è multicast lo passo al livello superiore

18/10/21

Stato Porte

- posso attivare o disabilitare alcune porte
- Forwarding / Blocking: quando si fa l'albero una porta può essere disabilitata (Blocking) o attiva forwarding. Ogni porta ha un mac.
- Il mac viene consolidato vero per 5 min. poi si ricalcola
- Per il learning si effettua tramite pacchetti di broadcasting (sa dove sta il pc) - BPDU - bridge PDU con codice multicast 01 08 c2 00 00 00.
  l'indirizzo llc di spanning tree 01000010

Cut through

- evita che il bridge salvi tutto il pacchetto ma trasmette subito appena capisce mac -> riduce latenza
- Non funziona se:
  - pacchetti diverso protocollo
  - reti diverse velocità
  - porta occupata

Evoluzione Ethernet

DIFFUSIONE SWITCH E ETHERNET FULL DUPLEX:
uso una comunicazione bidirezionale ossia mentre ricevo trasmetto anche perchè con gli switch ho un cavo per computer.
=> Perdo il CSMA/CD con comunicazione unidirezionale. (HALF-DUPLEX) => se ho repeater ho CSMA-CD, ritrasmesso rep. ho 2 coppie per andata e ritorno -> no collisioni

Da 802.3 si generano sottogruppi che di volta in volta standardizzano collegamenti più veloce (100BaseT..)

Controllo di flusso (IEEE 802.3x e 802.3bd)

- Uso controlli di flusso con lo sviluppo di ethernet
- Di fatto posso avere porte con diverse velocità il bridge può spedire un pacchetto di pausa per al server che invia dati, così che questo mettendosi in pausa permette al bridge di scaricare il buffer di ingresso.

=> Pacchetti di controllo. (802.3x)

ho un nuovo livello mac che deve gestire i pacchetti di controllo.
se non ho 802.3x non posso fare affidamento su pacchetti di pause

21/10/21

RETI LOCALI SENZA FILI (802,11)

- È suscettibile a problemi di consumi, di limitazioni sull'area coperta e di sicurezza, poiché trasmetto in aria => ciò comporta una scarsa affidabilità del mezzo trasmissivo.

- Il wifi nasce dal consorzio Wireless Ethernet Compatibility Alliance poi Wifi che deve certificare la funzionalità.

RETI AD HOC E STRUTTURATE

RETI AD HOC: reti in cui le stazioni comunicano direttamente (sono pochi)

RETI STRUTTURATE: La comunicazioni fra stazioni avvengono tramite access point (AP)

- I vari access point sono collegati tra loro con fili (802.3 => WIRED)
- È scalabile

ACCESS POINT:

- sono sostanzialmente dei bridge. L'AP ha il mac 802.11 e controlla un BSS (basic service set) ossia un insieme di stazione collegate al AP.
- Il wireless estende le reti wired

Ogni BSS ha un identificatore ossia BSSID

DISTRIBUTION SYSTEM: (DS) ha un backbone al quale una rete wifi si appoggia

Lo standard prevede la possibilità di avere una backbone in WIRELESS

· Fa learning: alcuni sistemi sono visualizzati in WIRED altri WIRELESS

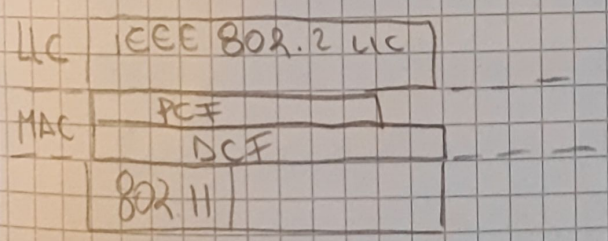
MAC

sottolivello LLC è 802.2

Il mac si occupa:

- accesso al mezzo
- sicurezza

STRUTTURA



PCF: Point coordination function

DCF: Distributed coord function.

- Nel wireless si usa un accesso distribuito al mezzo coordinato dal DCF
- L'accesso alla rete è coordinato da PCF se volessimo avere un controllo central.

CAMPI MAC

Oltre quelli di 802.3 ho:

- frame control: mi dice il tipo di pacchetto in analisi con informazioni su:
  - origine e destinatario
  - sicurezza
  - frammentazione
- Durata: quanto dura la trasmissione di un pacchetto
- Indirizzi: fino a 4
- Sequence controll: frammentazione
- Body: LLC
- FCS

INDIRIZZAMENTO

Ogni scheda wireless ha un mac e in base a questo indirizzo la scheda sceglie di raccogliere o meno pacchetti.

TODS: pacchetto indirizzato al sistema di distribuzione ossia se invio ad esempio ad access point bit = 1

FROMDS: dal centro di distribuzione

ADDRES 1: chi invia

ADDRESS 2: scheda che fisicamente invia pacchetto

ADDRESS 3: se FROMDS = 1 => Da AP a destin

TODS = 1 => DA MITTENTE a AP

↳ ha indirizzo destinatario

ADDRESS 4: Usato solo se backbone è WIFI

TABELLA

AP1))) AP2

TODS	FROMDS	ADD 1	ADD 2	ADD 3	ADD 4
0	0	DA = C2	SA = C1	BSSID	
0	1	DA = C2	BSSID = AP	SA = C1	
1	0	BSSID = AP	SA = C1	DA = C2	
1	1	RA = AP2	TA = AP1	DA = C2	SA = C1

C1 C2

RA: scheda ricevente TA: scheda trasmittente

DA: scheda finale SA: scheda che ha inviato realmente

DCF (distributed Coordination Function)

- Deve arbitrare l'accesso al mezzo

REQUISITI:

1) evitare che trasmissioni interferiscono tra loro
2) no controllo centralizzato
3) no clock

Usa CSMA/CA (collision avoidance)

CA: non identifica le collisioni, ma cerca di evitarle

Funziona come CSMA

CA:
- non faccio collision detection perché non riesco a identificare il segnale separandolo dal rumore
- Cerca di evitare le collisioni
- Uso:
  - backoff
  - acknowledgment (pacchetto arrivato?)
  - REQUEST TO SEND (RTS)
    CLEAR TO SEND (CTS)
  - Intervalli fra pacchetti consecutivi per gestire priorità

spazio i pacchetti:

DIFS:

SIFS: short inter frame space (SIFS<DIFS) più piccola è lo spazio più i pacchetti hanno priorità

DCF senza RTS e CTS

- posso usare questo meccanismo con RTS/CTS ma anche senza

Evito collisioni come:

- se una S1 vuole trasmettere ma il canale è occupato aspetta un tempo random [0, cw] ossia aspetta il tempo di backoff. Se il canale si occupa durante il backoff il timer (di backoff) si ferma. Se timer = 0 avviene la trasmissione
- Se rilevo una collisione cw × 2 con cw < cwmax
- Se il pacchetto è inviato correttamente cw = cwmin

tipico $cw_{min} = [7, 31]$

$cw_{max} = [255, 1023]$

- TRASMISSIONE:
  - Campo Duration: specifica la durata di trasmissione che viene salvato da ogni stazione in ascolto e scritto in NAV NETWORK ALLOCATION VECTOR

Il NAV sincronizza la rete stazione

- viene decrementato da ogni stz al passare del tempo
- Se NAV = 0 allora posso trasmettere
  se NAV ≠ 0 => canale occupato

Il nav dura come la trasmissione + SIFS + ACK

- I pacchetti sono riscontrati da acknowledgment (ack)
- Per trasmettere ack devo aspettare SIFS (SIFS < DIFS => priorità)
- NAV = DURATION + ACK + SIFS
- Ho collisione se al termine di una trasmissione un ack

TRASMETTO DOPO NAV + DIFS SE NAV = 0

LIVELLO FISICO dice che canale è libero

SIFS e DIFS sono noti a priori

25/10/11

ack e collisioni

COLLISIONE: pacchetto che non ottiene ack

→ Se una stazione sta trasmettendolo, il mezzo del pacchetto non si blocca se non termina il pacchetto. Se il pacchetto ~~arriva~~ va a buon fine ottiene un ack

PROBLEMI:

1) Se invio un pacchetto troppo grande ci vorrà molto tempo per riscontrare collisione

2) Al contrario di CSMA/CD non sempre CSMA/CA le stazioni "si vedono" ossia non so se effettivam. l'altra stazione sta già trasmettendo e trasmette anche lei sul medesimo canale sporcando il segnale dell'altro.

PROBLEMA STAZIONE NASCOSTA

Inserisco RTS/CTS (FAI schemino)

· Meccanismo richiesta canale.

Stazione che vuole trasmettere alla ricevente un piccolo pacchetto in cui si richiede lo spazio per comunicare (RTS) nel quale si dice anche quanto dura il pacchetto da inviare. Se la comunicazione è possibile la stazione ricevente invia a quella TRASMITTENTE un pacchetto con cui concede l'uso del canale e T per tutte

la durata della trasmissione dichiarata in RTS. Questo è un CTS

La duration viene salvata infine in NAV per sincronizzare stazioni.

La collisione dunque è dovuta all'invio simultaneo dell'RTS da parte di 2 stazioni. e di conseguenza il mancato ricevimento di CTS

LAVORANDO in CSMA/CA ogni stazione ha un raggio di azione che permette di ricevere i pacchetti inviati da una seconda stazione o da un gruppo di stazioni limitrofe.

Il NAV in questo caso si aggiorna in ogni stazione in base alla stazione che la vede:

-> se vedo sta trasmittente copio sul NAV il tempo presente in ~~CT~~ RTS

-> se vedo stazione ricevente la copio dal tempo presente da CTS

-> Se la stazione che sto analizzando non vede ne Tx ne Rx allora questa stazione non è interessata dalla comunicazione quindi non ha bisogno di tempo

1) CTS e RTS sono molto piccoli perché così percepisco in tempi brevi la collisione

Se i pacchetti da inviare sono molto piccoli preferisco non usare CTS/RTS ma, nel caso di collisione, farla avvenire direttamente sul pacchetto da inviare e non su CTS/RTS => + efficiente.

Definisco dunque una soglia S sotto la quale CTS/RTS non si usa.

-> spesso S = 1500 byte ossia il massimo n di byte per un pacchetto => non uso mai CTS/RTS pk introduco troppo overhead sul canale

ASPETTI MAC

HANDSHAKING:

L'insieme delle stazioni in un BSS cambia sempre. Perciò se una stazione cambia BSS allora la stazione che si vuole collegare invia pacchetti con il proprio indirizzo MAC (come anche AP)

2 modi per accedere a BSS

-> Uso pacchetti "beacon", inviato da AP per farsi riconoscere

-> Uso pacchetti "PROBE", inviato da stazione per verificare la presenza di AP

- Si possono definire varie reti logiche sulla stessa RETE FISICA.
  - ognuna ha un suo proprio SSID (service set ID)
  - Ogni AP con BSSID può dirsi disponibile a far passare dati di 1 o più SSID

ESS: extended service set ossia insieme delle stazioni appartenenti ad una stessa BSS con stesso SSID
-> Le stazioni possono comporre BSS unionendo quella stessa ESS

FRAMMENTAZIONE

- Uso frammentazioni nelle reti wi-fi per ridurre overhead nel caso di ritrasmissioni di pacchetti
  -> 1 pacchetto grande viene 'ripensato' come un insieme di pacchetti + piccoli considerati come pacchetti autonomi (stessi comp + uscenti) con presenza di un dettaglio che permette a Rx di ricomporre il pacchetto

I livelli superiori di MAC non si accorgono della frammentazione.

RAPPORTO TRA MAC - LLC

In 802.11 MAC si occupa degli acknowledg. come LLC in IEEE 802 ma "802.11 è fondamentale, in LLC può non essere usato, nelle reti locali non si usa di fatto.

Pacchetti broadcast in WiFi

Se ToDs=0 allora CTS/RTS e ack così che la collisione non si alzi

Se ToDs=1 si usano ack con RTS/CTS
↳ pacchetti verso AP

-> Si configura un AP in modo tale che il pacchetto BROADCAST sia ricevuto da tutte BSS

DATA-LINK (liv 2) fuori da LAN

LAN: trasmissione di 2 stazioni su un canale condiviso

WAN: trasmissione di 2 stazioni su canale PUNTO-PUNTO

RETE di RACCOLTA e RETE di ACCESSO: Si trovano tra lan e wan

RETE DI RACCOLTA:

FTTN (fiber to the NODE) e FTTC (fiber to cabinet) sono i punti di raccolta + conosciuti => XDSL

FTTH e FTTB: 1 gbit/sec
A cambiare è la distanza dal punto di raccolta a casa non in fibra

G PON

Gigabit-capable Passive optical Network
che è un punto di raccolta di traffico.

→ Si usa un (olt) optical line termination
che funziona come uno switch che dato
una rete unica si spartiscono m+ clienti.
Prima di olt ho un altro switch da
cui arriva 1 fb e ne escono 8 da olt
ne arrivano 8 e ne escono 64 (condivisione banda)

RETE DI ACCESSO

- Si trova fra WAN e punto di raccolta
  ⇒ si chiama MAN (metro Area Net)
- Si compone di switch connessi ad anelli (con + fibre)
  ogni switch raccoglie i dati dai
  vari punti di raccolta "sparsi" nella città
  e li inoltra su un nodo di aggregazione
  per poi essere inviati sul backbone.
- Si usa una rete ad anello così che
  alla rottura di uno switch solo i clienti
  che afferiscono a quel nodo senza
  interrompere la connessione su una zona metropolit.
  ⇒ Si usa spanning TREE

Vari aggregation NODE sono legati
da collegamenti PUNTO-PUNTO

28/10/21

Linee Punto-Punto affidabili

Come le realizzo?

ESEMPI DI PROTOCOLLO LIVELLO 2 WAN

- HIGH-LEVEL DATA LINK CONTROL (HDLC)
- POINT TO POINT PROTOCOL (PPP)

Per rendere affidabili un canale:

→ Ad ogni invio di pacchetti si richiede sempre un riscontro (acknowledgment, ack)

GESTIONE ACKNOWLEDGMENT

1) STOP AND WAIT: invio 1 pacchetto aspetto per inviare un secondo pacchetto, il riscontro dal ricevente. Se entro un periodo non arriva ack, ritrasmetto. (uso anche nack)

2) GO BACK N: TRASMETTO IN SEQUENZA I PACCHETTI, alla fine della trasmissione la ricevente trasmette l'acknowledgment o il nack (non acknowledgment). Se non arriva ack o nack in un certo tempo, ritrasmetto dal pacchetto ultimo di cui ho il riscontro e tutti i suoi successivi.

3) SELECTIVE REPEAT: trasmetto in sequenza e ritrasmetto solo il pacchetto che non ha ack esplicito

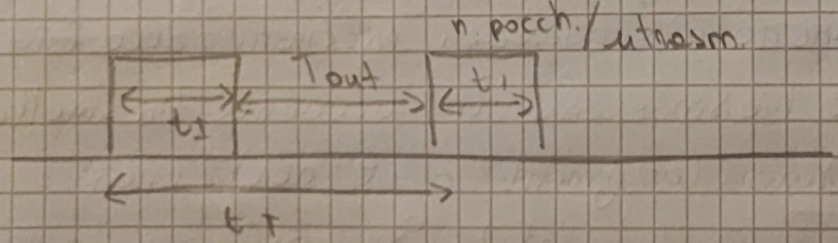
Hp x analisi: schema numerativo infinito
=> n. infinito di bit

In GO BACK-N ho bisogno di avere uscenti mentre invio. (canale di andata e canale di ritorno) => Questo è vero per reti Punto-Punto non per 802.11

ANALISI STOP-WAIT

t_1 = tempo per trasmettere
t_{out} = tempo tra un pacchetto e il succ.
t_p = t propagazione
t_{proc} = t processamento pacchetto
t_s = tempo per trasmettere ack
$\Rightarrow t_{out} \geq 2t_p + t_{proc} + t_s$

$t_T = t_1 + t_{out}$ - tempo minimo tra inizio 2 pacch.



Hp:

1) ho 2 stazioni in cui A invia e B trasmette ack
2) t. trasm ack/nack costante
3) A con coda spedizione piena
4) pacchetti di dim. uguali

finito t_{out} o ritrasmetto o invio altro pacchetto

OBBIETTIVO: determinare il massimo throughput (pacchetti ricevuti nel tempo)

$1/t_T$ massimo pacchetti trasmessi => massimo throughput

throughput $< 1/t_T$ perchè potrei avere errori o ritrasm.

GO BACK-N => $1/t_i$

ANALISI GO BACK N

1) $t_{out} = t_I$ => tempo costante pk ho trasmissione successiva di 2 pacchetti

2) max throughput $= 1/t_i$ che sarà minore a causa di errori e ritrasmissioni

SCHEMA NUMERAZIONE FINITO

- Numero i pacchetti con n bit limitati
- M è il modulo della numerazione ed N è il numero attribuito al pacchetto => $0 \leq N \leq M-1$
- Riscontro R, indica che ho ricevuto correttamente fino a R-1, in attesa dell'R-esimo
  -> riscontro cumulativi => un riscontro per + pacchetti
- In ogni istante posso avere al più M-1 pacchetti in attesa di riscontro.
- Se suppongo che mi arrivi il riscontro di 0 allora questo significa che:
  - ho ricevuto tutto
  - non ho riscontrato nulla

Questo è il problema dell'usare M numeri contempor.

GO BACK N CON SCHEMA FINITO

2 cose possibili:

$2t_p + t_s \geq (M-2)t_1$

continua o discontinua?

Dopo il riscontro di un pacchetto posso trasmettere di nuovo una serie di pacchetti

→ Se $2t_p + t_s > (M-2)t_1$
allora il riscontro del primo pacchetto arriva dopo la fine della seq. trasmissiva e di conseguenza avremmo una trasmissione frammentata (non consecutiva)

→ Se $2t_p + t_s < (M-2)t_1$
allora il riscontro del primo pacchetto arriva prima della fine della prima trasmissione
⇒ trasmissione continua.

Si sceglie M in base alla distanza $2t_p + t_s$

Se la trasmissione è continua Go Back-N ha throughput $= 1/t_s$

Se la trasmissione è discontinua allora throughput $= (M-1)/(2t_p + t_s + t_1)$

Protocollo PPP

Il suo contenuto è HDLC

BIT-STUFFING: un pacchetto HDLC è racchiuso tra 2 flag (0E) che indicano separazione tra pacchetti

- SINCRONIZZA
- SOLO FLAG ha 6 uno consecutivi, altrimenti ho 5 uno e uno zero eliminato in ricezione

INFORMATION FRAME: pacchetti in modalità connessa

SUPERVISON-FRAME e UNBEDDED: info di controllo

PPP

- collegamento punto-punto full duplex
- massimo 1500 byte
- non ho riscontri pk assumo che il canale sia affidabile
- pacchetti non numerati

-> LCP (link control protocol) che ha il compito di configurare e testare la connessione.

-> autenticazione ap

-> monitoraggio link

PPPoE

PPP over ethernet

usato per incapsulare ppp su ethernet (pacchetto)

2 livelli 2 uno nell'altro

04/11/2021

METODI E MODELLI PER LIVELLO DI RETE

- Il livello di rete sceglie una strada per i pacchetti
- La strada è composta da salti, di conseguenza il livello di rete stabilisce quali salti deve fare il pacchetto per raggiungere la destinazione.

- Il livello di rete fornisce al livello di trasporto un servizio indipendente dalla topologia di rete e dalle tecnologie usate sulla rete

È connesso o no?

→ Possono essere sia connessi che non connessi
il più diffuso protocollo di rete è IPv4

→ L'instradamento è di pacchetto è di pacchetto
- Protocollo non connesso è a datagramma
- Protocollo connesso è a circuito virtuale

Il livello 4 deve conoscere:
- Le macchine che deve attraversare, conoscendo indirizzi univoci
- deve avere delle API, indipendentemente dalla ubicazione della macchina ricevente

Per servizi connessi la primitiva che il livello 3 deve dare al livello 4 è più ampia perchè deve instaurare un dialogo prima di inviare pacchetti (+ Payload)

I sistemi sulla rete possono essere:

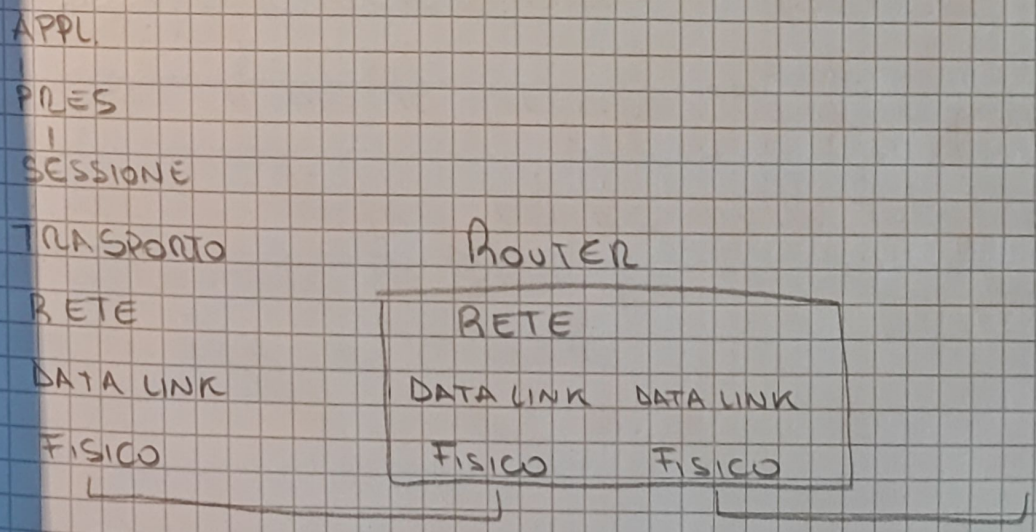
- End system
- intermediate system

Ad ogni sistema associo un indirizzo numerico per identificare il sistema.

I sistemi intermedi che contengono anche il livello 3 si chiamano ROUTER / GATEWAY

L'instradamento si fa a livello 3 ma esistono instradamenti meno complessi a livello di link si fanno a livello 2. (BRIDGE)

ROUTER → ISO-OSI



Dato link e fisico di sx possono essere diversi da quelli di dx pk per esempio sx è 802.3 e sx è 802.11

INDIRIZZAMENTO

- Il indirizzo livello 2 (mac) può identificare solo macchine nella lan
- L'indirizzo livello 3 identifica tutte le macchine anche non in lan

→ Ho tanti indirizzi mac per ogni interfaccia di rete posseduta dal sistema

METODI INSTRADAMENTO

ROUTING BY NETWORK ADDRESS

- Nel pacchetto ho l'indirizzo del ricevente e faccio commutazione a datagramma

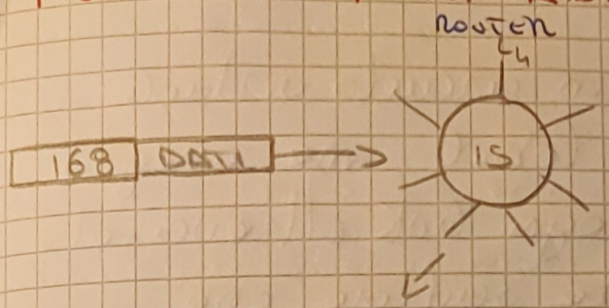
LABEL SWAPPING

- Ho una label che indica il cammino da fare e ho commutazione a circuito virtuale

SOURCE ROUTING

-

ROUTING BY N. ADDRESS



Nei router ho una tabella di instradamento verso ogni pc sulla rete e figura la linea di uscita

Dunque il router cerca il valore 168 e determina la ~~st~~ linea di instradamento

=> Commutazione a datagramma

-> La tabella può cambiare sempre, di fatto cambia la linea di instrad. => La tabella ha tante righe quante sono le linee (circuiti virtuali)

LABEL SENZA SWAPPING

La label identifica il circuito virtuale su cui deve ritrasmettere il pacchetto

La tabella della label è molto piccola => ho tante righe quanti sono i cammini virtuali

Il router dunque ricerca nella tabella di instradamento la label per trovare il cammino da far fare al pacchetto

LLC - SAP - ETHERTYPE identificano protocolli nello stesso macchina

LABEL CON SWAPPING

Quando costruisco un circuito virtuale non assegno un solo label ma assegno etichette diverse per ogni link attraversato
=> Così che non debba verificare che l'intero circuito sia occupato => numero di circuiti sono limitati

La label si sceglie su ogni link ossia lo decidono gli IS che label assegnare assegnare a link che conducono

Il n. di label totali non è un limite superiore al n. di circuiti virtuali attivabili
(PROVIDER SERVIZI INTERNET)

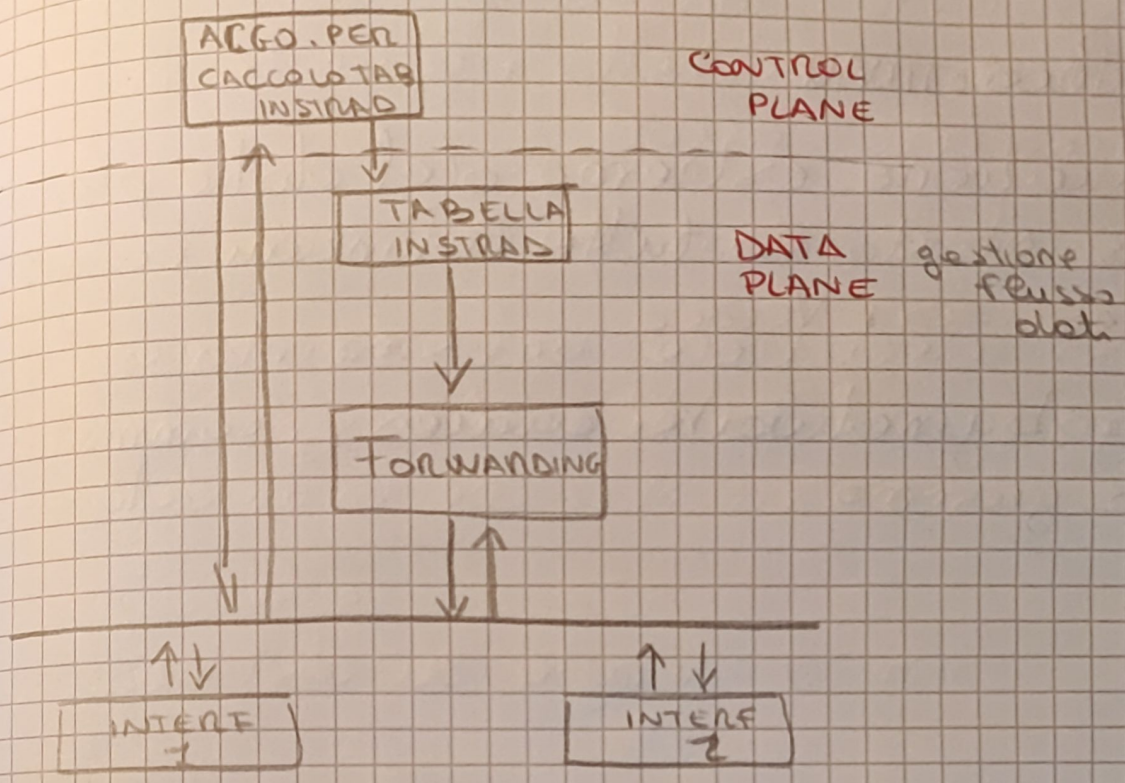
SOURCE ROUTING

- Nel pacchetto ho segnata tutta la strada da fare per arrivare al destinatario.
(ho tutti gli indirizzi)

Gli end-system sviluppano loro le strade migliori. | A | R | G | S | B | DATI |

Si può usare in IP

ARCHITETTURA ROUTER



I router possono essere multilivello in quanto posso avere più protocolli di livello 3 attivi differenziati da LLC o ethert IPe

I vari protocolli non sanno cosa fanno gli altri protocolli di liv 3 attivi allo stesso momento

IPv4 e SUITE TCP/IP

IPv4 - internet protocol versione 4 e

TCP/IP

Ques... ...ocol

rete ...ntot

I pro... son

de... 4) e

chi... on o

INTERNET:
- nome rete
- nome protocollo di livello 3: IP
- Nome insieme di protocolli realizzati attorno a IP

- IPv4: principale protocollo di rete della suite TCP/IP in cui TCP e IP sono 2 particolari protocolli di un insieme di protocolli. Quella suite che prevede protocolli standardizzati attorno a IP si chiama "Internet protocol suite"

TCP/IP

Principi progettuali

- Semplificazione estrema del data plane degli es. spostando tutte le funzioni + complesse sugli es. => + VELOCE
- Routing by network address, commutazione a data gramma

IPv4

- Protocollo principale di liv 3 descritto in RFC 791
- I pacchetti in IPv4 si chiamano datagrammi => IPv4 datagram e indica il pacchetto a liv 3.
- Funzioni di: instradamento, rilevazione errori (liv 4 obsoleto), frammentazione e riassemblaggio
- Riceve messaggi da TCP e UDP (liv 4) li instrada e li frammenta se necessario

Il datagramma ha una grandezza massima scelta dal livello 2, se è troppo grande rispetto a questi vincoli IPv4 frammenta

IPv4 non riassembla in un intera

Il routing è semplice nel caso in cui viaggi su una lan

PACCHETTO Linguaggio

Viene diviso in più righe "Lo

(4 bit)
ho version: mi dice che versione ha generato il pacchetto

(min 4 bits
IHL: lunghezza header in dong words
min 5 intest
=> intestazione variabile (dipende dai compl option)

Type of Service: Servizio richiesto sul pacchetto

si mantiene se ho un solo operatore altrimenti cambia a seconda il gestore

Total lenght:

Ident: FRAMMENTAZIONE

- Ogni mezzo fisico può avere un maximum trasmission unit MTU diverso.
- Se mi trovo in UmIs che deve instradare il pacchetto verso un altro system con MTU minore => Livello di rete frammenta
- Non riassembla ma qualora mi trovassi ad avere MTU' > MTU così da velocizzare

Type of service: specifica la priorità di un datagram. Diviso in:
DSCP: "Differentiate Service Code Point" a seconda del suo valore ho una priorità diverso.
ECN: Explicit Congestion Notification

FRAMMENTAZIONE E RIASSEMBLAGGIO:

- Divisione di un pacchetto + grande in più frammenti + piccoli
- Si riassembla solo su ES.
- Non posso identificare un frammento con un numero perchè potrei aver bisogno di una ulteriore frammentazione in un determinato IS

Uso OFFSET: numero di byte di distanza dall'origine

IDENT: (16 bits) identifica un datagramma, serve ad es. ricevente per riassemblare pk se IDENT è uguale si tratta dello stesso pacchetto

FRAGMENT OFFSET: (13 bit)
- usato per unire il pacchetto nel giusto ordine
- gestisce offset multipli di 8. (primi 3 bits = 0)
- 8192 frammentazioni totali

Flags (3 bit)
- sempre 0 (1° bit)
- Dont'fragment DF, indica se il datagramma può essere frammentato
- More Fragment (MF) indica quanti altri pacchetti ho frammentati

Poichè li ho frammentati in liv. 3
alla fine la ricostruirà il livello 3
il liv. 4 è indifferente.

TOTAL LENGHT (16 bits): HEADER + PAYLOAD max 65.535 byte

TIME TO LIVE (8 bits): Specifica il tempo di vita del pacchetto

- Se si genera un ciclo per non funziona il routing e il pacchetto ha vita infinita allora quel pacchetto gira all'infinito nella rete senza arrivare a destinazione. Per evitare questo lo scarto alla scadenza di Time to live. Il pacchetto di notifica di un pacchetto 'buttato' se viene perso non viene + notificato
- Il numero di time to live si decrementa ad ogni salto (fra Es-Is, Is-Es) e un pacchetto scartato viene notificato al mittente. Massimo 255 salti

PROTOCOL (8bit): identifica il protocollo di livello superiore che si trova nel payload nel pacchetto IPv4.

- Ho valori stabiliti in RFC 1700
  TCP = 6 e UDP = 17

Header checksum: riguarda l'header
si ricalcola ad ogni hop perchè fai decrementare TTL

Option

ogni opzione inizia con 1 byte che specifica il tipo.

Sono del tipo:

- Sicurezza
- Source routing.
- loose source routing (router da attraversare non tenendo conto dei router intermedi)
- Record route
- Time stamp.

Indirizzamento IPv4

. Sono associate a una interfaccia e non all'host => se una macchina ha + IPv4

→ Indirizzi di 32 bit (4 byte)

sono univoci nel mondo

sono rappresentati da decimali separati da punti

Se gli indirizzi IP fossero arbitrariamente allora ogni router dovrebbe avere una tabella di instradamento enorme

=> Assegno a macchine vicine tra loro stesso prefisso => Uso come routing le tabelle dei prefissi

Es

193.204.161.1 prefisso

193.204.161.1 prefisso Dip Ing

- Per rendere un prefisso un indirizzo IP reale associo come ultimo bit uno zero.

193.204.161.0

- Gli indirizzi con prefissi uguali sono tutti su uno stesso ugo
- Un prefisso è univoco, non posso avere 2 net con prefisso uguale

funzioni di IP:

IDENTIFICAZIONE:

LOCAZIONE: stesso IP => macchine nella stessa rete

15/11/21

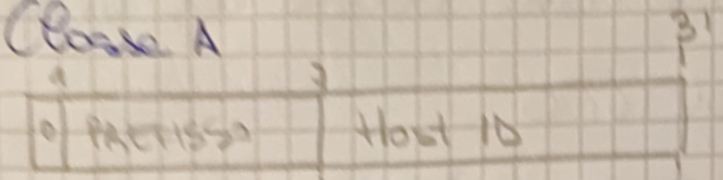
Indirizzamento IPv4 (2)

Gli indirizzo sono raggruppati in Net
=> stesso prefisso per le macchine vicine
=> 0 si usa solo per router non per macchine

Semplifico tabelle di instradamento

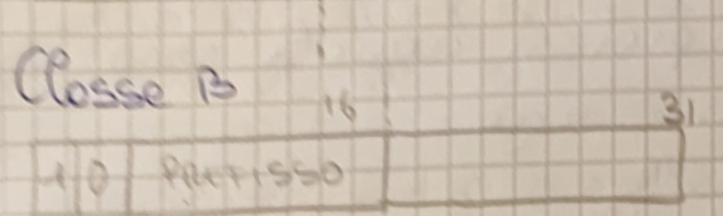
Indirizzamento IPv4 classful

Classe A



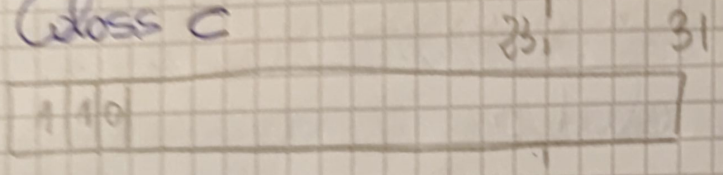
- poche lan molti indirizzi con 16.000.000 Host

Classe B



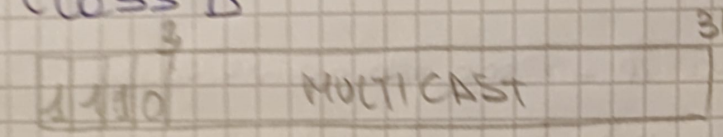
- più lan
- pochi pc per lan

Class C

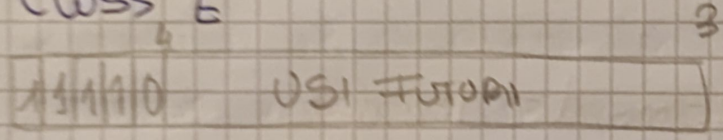


molte lan
pochi macchine

Class D



Class E



Indirizzamento classless - NETMASK

Per consentire agli indirizzi di avere al loro interno la parte prefisso di lunghezza variabile (che varia in base al tipo di lan) si abbandona l'indirizzamento <u>classfull</u>

- In classfull il prefisso ha una lunghezza prestabilita, mentre per classless si deve indicare quant'è lungo il prefisso => si usa netmask

La netmask è composta da valori di 1 contigui e di 0 contigui in cui gli uni corrispondono al prefisso e gli zero alla macchina specifica.

Ex 11111111. 11111111. 11111111. 00000000

leggo la netmask in decimale

$\underbrace{255 . 255 . 255}_{\text{PREFISSO}}$, $\underbrace{0}_{\text{HOST}}$

Ex2. 11111111. 11111111. 11111111. 10000000

255. 255. 255 128

11111111. 11111111 11111110.00000000

255 255 254 0

Gli indirizzi con stesso prefisso hanno la stessa netmask
=> netmask di un prefisso

INDIRIZZO IP

193.200.160.0 /24 → prefisso di 24 bit

netmask: 255.255.255.0

255.255.254.0 => 193.200.160.0/23

Estraggo il prefisso 'mettendolo in AND' l'indirizzo con la netmask perchè li confronto con 1 mentre ottengo 0 per host

Gli indirizzi IP sono dati da un ente di gestione Internazionale, così che gli IP siano diversi (disgiunti tra loro) altrimenti 2 indirizzi IP di diversi pc. uniti con la netmask ritornano IP uguali => impossibile

Invio Pacchetti

Si mette in And bit a bit l'indirizzo destinatario con la tua netmask
se il prefisso è uguale a quello del mittente mi trovo nella stessa lan allora invio direttamente alla macchina
Se indirizzi sono diversi il destinatario si trova fuori dalla mia lan => invio pacchetto a Is (router di default)

È la macchina mittente che si chiede se l'indirizzo suo IP, in particolare il suo prefisso, è uguale a quello del destinatario per capire dove inviarlo.

Esempio

- macchina mittente con indirizzo
  193.204.161.8
  con netmask: 255.255.255.128
- macchina destinatario: 193.204.162.0

METTE IN AND il suo indirizzo con la sua netmask e ottiene: 193.204.161.0

METTE IN AND l'indirizzo destinatario con la sua netmask e ottiene:
193.204.162.0

193.204.161.0 ≠ 193.204.162.0
=> pacchetto da inviare su LAN diversa

La macchina mittente non conosce la netmask del destinatario

Se gli indirizzi sono assegnati in modo disgiunto allora i PC quando estraggono il prefisso non possono sbagliare

Se è fuori dalla LAN posso anche sbagliare in estrazione perché comunque può funzionare il routing

IP LOOKUP

Ogni riga di una tabella di instradamento del router ha il prefisso e netmask ed effettua il matching (mette in and l'indirizzo e netmask) prima di inviare il pacchetto sulla linea di riferimento.

Si scorre verso il basso della netmask con prefissi maggiormente grandi a quelli + piccoli

Se non ho matching butto il pacchetto

L'ultimo indirizzo è composto da tutti 0
=> Default 0/0

IND: 128.96.40.151 => 128.96.40.0

MASK => 255.255.255.128

MASK: 11111111.11111111.11111111.$\underbrace{10000000}_{127}$

L'indirizzo 128.96.40.151 messo in and con 255.255.255.128 ritorna come risultato 128.96.40.128 quindi non può essere inviato su 128.96.40.0

Di fatto 255.255.255.128 ossia

11111111.11111111.11111111.10000000

AND

10000000.01100000.00101000.10010111
(128) (96) (40) (151)

10000000.01100000.00101000.10000000
(128) (96) (40) (128)

192.4.153.128 ∈ 192.4.153.0 ?

no perchè 192.4.153.0 sul 4° byte ha indirizzi che iniziano con 0 mentre 192.4.153.128 ha sul 4° byte un 1 e tutti 0 ossia

11000000.00000100.10011001.10000000
(192) (4) (153) (128)

11000000.00001000.10011001.00000000
(0)

Gli zero che sono nel 4° byte posso essere altre significative per il prefisso come nel caso della netmask 255.255.255.192 in cui le prime cifre del 4° byte sono 2 uno. Rappresentano dunque un indirizzo /26 e anche se instrado su questo un indirizzo del tipo 192.4.153.0 non posso considerare la netmask del tipo /24 in quanto l'ultimo byte è composto da soli 0 per questo indirizzo poichè i primi 2 zero del 4° byte continuano a rimanere cifre significative per il prefisso => 192.4.153/26

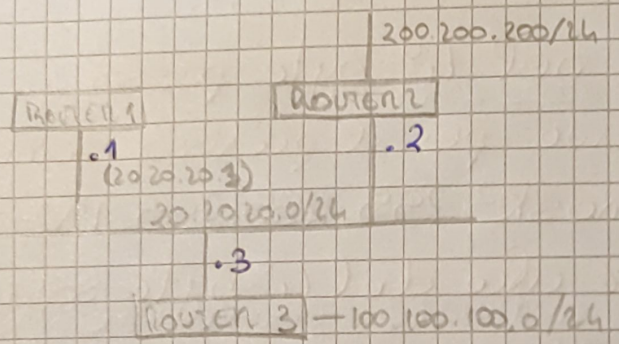
Quindi per trovare quanti sono gli host totali che possono essere ospitati da un prefisso si vede l'ultimo byte e si sottrae a 255 il numero decimale corrispondente a questo byte (nella netmask)

Quindi nel caso del primo prefisso 192.4.153.0 con netmask 255.255.255.192 ho al massimo 255-192=63 Host

18/11/2021

Control Plane costruisce la tabella.

NEXT hop: specifica l'indirizzo IPv4 della successiva interfaccia a cui inviare un pacchetto



NETWORK	NETMASK	INTERFACCIA	NEXT HOP
20.20.20.0	255.255.255.0	Eth0	dirette con (D C)
100.100.100.0	255.255.255.0	Eth0	20.20.20.3
200.200.200.0	255.255.255.0	Eth0	20.20.20.2

DIR CON: DC. significa che la rete è direttamente collegata al router

Il traffico di 100.100.100.0 per essere instradato correttamente deve essere inviato ad un secondo router che può inviare il pacchetto pk direttamente collegato al router. Si usa sempre Eth0 pk collega fra loro i router sulla lan

Negli end-System esiste una tabella di instradamento con:

- Il prefisso e netmask della rete a cui appartiene
- Una linea 0/0 con cui sul next hop ho l'indirizzo del default gateway.
- Una linea 127.0.0.0 con netmask 255.0.0.0 ossia una /8 => linea di loop (Co) back ossia si instradano qui tutti i pacchetti diretti alla stessa macchina => INTERFACCIA VIRTUALE, si usa per parlare tra 2 processi diversi sulla stessa macchina. => Risultato: Applicazioni disaccoppiate ossia che lavorano indipendentemente l'una dall'altra. Essendo una /8 allora sulla stessa macchina ho interfacce diverse (ne ho di più)

Indirizzi convenzionali

-> 0.0.0.0: solo per rotta di default
non si usa per altro

-> 127.0.0.1: Questo pc.

-> 255.255.255.255: Pacchetto [illegible] oto
si usa di fatto l'indirizzo [illegible] ii
si vuole inviare il pacchetto [illegible] ltimo
byte di tutti 0, un byte di [illegible]

-> 10.0.0.0/8 o 172.16.0.0/22 o [illegible] ii
privati usati per lan private non per internet in generale

PER IDENTIFICARE INTERFACCE IN UNA LAN:
ammettendo di avere una prefisso di 24 allora avrei 256 macchine, ma l'indirizzo di tutti zeri è il prefisso quindi non può essere usato e l'indirizzo di tutti 1 è riservato quindi non può essere usato
-> numero solo 254 macch.
VALE X OGNI CUNGH. PREFISSO

Ogni dispositivo ha + ~~m~~ schede per interfacciarsi con l'esterno.

=> diverse tabelle di instradamento
una per interfaccia