

Reti telematiche - punti salienti & riferimenti libro

Canali comunicativi (p. 28)

- Cavo spesso (thick): RG-215: spesso molto schermato
- thin (soft): RG-58: sottile, flessibile, MENO SCHERMATO, ma + facile da installare.
- UTP (unshielded) / FTP (Foiled) / STP (Shielded) - TWISTED PAIR: riduce il modo comune e diafonia (cross-talk). → modalità con cui si intrecciano!

↳ Frequenza del segnale: cambia l'attenuazione nella guida!

- Fibre: due strati concentrici con diversi indici di rifrazione, in modo da far propagare nel nucleo se l'angolo di incidenza è piccolo.

- ↳ Multimodali: le onde si propagano secondo diversi percorsi: nucleo ≈ manubrio
 - ↳ Monomodali: " " " " " " Secondo un unico percorso: nucleo << manubrio
- ATTENUAZIONE BASSA: $\approx \frac{1}{2} \text{ dB/km}$.

Nota: linee simplex e half-duplex richiedono 2 solo cavi, full-duplex almeno 2.

- ↳ Reti di...
 - ↳ Trasporto ⇒ 4 fili (metà trasmissioni DIVERSE)
 - ↳ Accesso ⇒ 2 fili (metà Tx - condivisi)

Commutazione: modo di operare dei NODI DELLA RETE DI COMUNICAZIONE

- individuare canale uscente dal nodo
- ↳ Instradamento → software: SELEZIONE dal nodo di uscita
- ↳ attraversamento → realizzazione HW della DECISIONE presa

Multiplexing:

- ↳ CBR: constant bit rate
- ↳ VBR: variabile " " ⇒ multiplexing statistico: si moltiplica a seconda di come serve statisticamente il chan; impiegata con sorgenti DISCONTINUE.

UI: Unità Informativa → header → info. di controllo } Store & Forward: nel nodo si memorizza sul buffer e appena si può si trasmette.

Buffer piccolo ⇒ ciò che non ^{sta} nel buffer vien perso
Payload: generalmente ha una length MASSIMA → segmentazione.

Comm. circuiti: INSTAURAZIONE, Tx DATI, RILASCO.
Per ogni chiamata le risorse assegnate son FISSE → i ritardi di Tx son COSTANTI.
(trasparenza temporale). Trasparenza semantica. Le trasmissioni arrivano UGUALI a come son state inviate: NO ERRORI.

I ritardi dovran esser "uguali": la rete a comm. di circuiti NON CONVALIDA LE UI.

A pacchetto SI, grazie alla memorizzazione e al "processing".

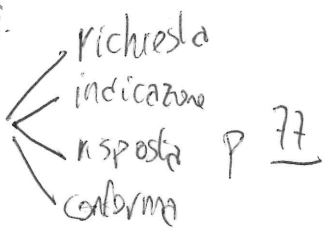
Legger BENE P. 75

Protocollo: insieme di regole.
Arch. tra protocolli: GERARCHIA TRA protocolli

RAGGRUPPAMENTO
STRATIFICAZIONE

SAP: unisce due livelli;

PRIMITIVE: tecniche per lo scambio di informazioni tra entità (di strati diversi)



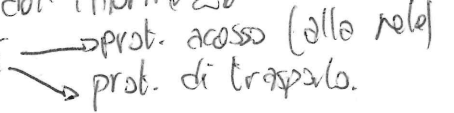
PDU: busta + lettera (Prot. dato unit)

PCI: busta (inf. di controllo: P. control unit)

SDU: lettera: DATI elaborati dallo N-entità (Service d. unit).

Le PDU "logicamente" vengono mandate da un livello N di un sistema A a un livello N di un sistema B: perché ciò avvenga FISICAMENTE, scendendo da N a N-1, da N-1 a N-2 ecc, ci deve essere la SEGMENTAZIONE, in modo da adattare le PDU del livello N a quello dei livelli inferiori; si arriva su B e, man mano che si sale, si RIASSEMBLA.

Strati di util. => generazione e utilizzo del DATO, dell'informazione
"a" trasporto => trasporto sulla rete delle informazioni



Modello ibrido: 3 strati

- Application: INTERFACCIA A UTENTE per accesso servizi informatici.
 - Trasporto: realizza il prob. di trasporto mediante scambio di info. Lavora con SEGMENTI MULTIPLEX, INDIRIZZA, SEGMENTA/RIASSEMBLA, gestisce la qualità del servizio.
 - Rete: MASCHERA al "trasporto" la tecnica di commutazione usata. Lavora con PACCHETTI, commutando e pacchetti. INSTRADA, MULTIPLEX RETI (PRIMA MULTIPLEXA TRASPORTO), controllo il flusso (prevenendo congestioni).
 - Collegamento dati: TRASFERIMENTO DATI TRA NODI (trame); rileva errori sulle trame, recupera quelle perse, controlla il flusso delle trame, controllo l'accesso.
 - Fisico: cavo/fibra/etere; gestisce il collegamento, le procedure di Tx (half/full duplex), CODIFICA i bit.
- Vedere modello OSI, p 84-86 (e altri modelli): OSI agisce in 7 livelli "sessione" (scambio di dati tra entità di presentazione) e presentazione (interlavoro tra applicazioni che lavorano con formati diversi): compressione, traduzione codici, trasduzione alfabeti, scureti
- Modello TCP/IP: 2 prot. di trasporto (TCP & UDP), strato rete (IP), perché dove INTERCONNETTARE RETI ETEROGENEE. lo strato di accesso gestisce FISICO & DATA-LINK, mascherabili a IP, in modo da trattare ALLO STESSO MODO con IP LAN o WAN! :-P

CONNOTAZIONI:

- Circuito: NO ELABORAZIONI: dopo l'instaurazione della connessione, tutta la capacità della rete è impiegata sul circuito non su elaborazione dell'informazione.
- Pacchetto: prima solo livello fisico ORA livelli 1-3: 1 trasf. fisico, 2 recupero errore, 3 controllo di flusso. Ciò è fatto dai nodi, che devono essere INTRINSECAMENTE elaborano.
- Frame relay: evoluzione pacchetto; dati mezzi trasmissivi di ALTA QUALITÀ, il controllo su ogni nodo sarà superfluo, dunque solo CERTI nodi faranno tutte le operazioni, gli altri solo alcune rispetto allo c.a pacchetto. I nodi di transito controllano la presenza di errori, quelli di accesso FAN TUTTO: controllo e recupero errori / congestione.
- RECUPERO: solo tra nodi di accesso.
- Livello 2 in 2 parti: 2H (solo nodi di accesso), 2L (accesso & transito), dividendo lo strato data-link in 2 parti e alloggiando il nodo di transito.

Data link

Obiettivo: migliorare le prestazioni della linea fisica con un certo tasso di errore, fornendo allo strato di rete una connessione SENZA ERRORI. Si trasportano 3-PDU all'interno di 2-PDU, dette TRAME → TRAMA ≡ 2-PDU.

OPERAZIONI: GESTIONE COLLEGAMENTO, DELIMITAZIONE TRAMA, ^{INNESTAZIONE} DESTINAZIONE TRAMA, CONTROLLO DI ERRORE/FLUSSO. → si usa un BUFFER per TX & / o uno per RX.

FEC: una volta trasmesso il messaggio, si CORREGGE con dei codici; informazioni aggiuntive che permettono al Rx di APPRENDERE gli errori della trama.

ARQ: invece di correggere, si "butta via" e fa ritrasmettere.

ARQ ricorda l'errore (con codici con ridondanze molto minori), quindi il Rx ordina la ritrasmissione.

Si introduce un BUFFER di Ritrasm, spesso coincidente con quello di Tx, che si svuota non meno che il ricevitore riceve, segnalando mediante un riscontro.

Stop & wait, nota: sia per le trame sia per i ACK si NUMERANO con 1 bit le cose mandate, per far sì che in caso di errori, NON SI MANDINO E CONSIDERINO DUE VOLTE LE STESSA COSE.

Continuous ARQ - Pigsybacking: nel caso di Tx bidirezionale, si fa che mandano il riscontro ASSIEME A UNA TRAMA CHE VA INDIETRO.

Ripartire da HDLC: p. 121 RICORDA che p. 290 va fatta o scannata. Merc. mattina ARRIVA FINO A P. 290

HDLC: stazioni PRIMARIA (comandi), SECONDARIA (risposte), TERZIARIA (ultime)

configurazione bilanciata (1 pr. diverse sec.), bilanciata di linea

mobilità di Tx

ABM: ogni stazione può rispondere (trasmettere SENZA AUTORIZZAZIONE (bilanciata) → LAP-B)

NRM: si risponde solo a comando (sbil.)

ARM: una stazione può parlare senza aut. della primaria.

Trama: 6 campi: Flag & Flag: delimitano la trama (8 bit): 01111110

Address: specifica la stazione trasmittente (se una trama viene da una staz. secondaria), o quello ricevente (per comandi emessi dalla primaria). TUTTI → Trama per tutte le stazioni.

Control: 8 o 16 bit, specifica il tipo di trama e gestisce i meccanismi di num. e di riscontro

Informazioni: contiene il dato

FCS: controllo errore

Ricorda: BIT stuffing: si introduce uno zero ogni 5 bit a 1 consecutivi.

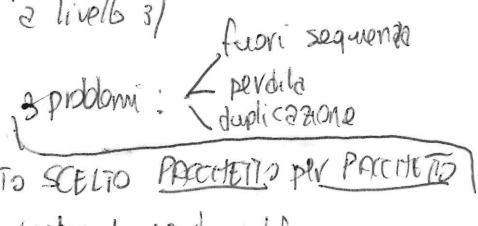
RIVEDERE COMANDI! :-P

WAN - strato rete (3)

Funzioni strato rete: trasferire unità informative tra entità dello strato di rete, fornendo il servizio allo PDU livello trasporto. A fornire servizio ALL strato rete sono le ~~N~~-pdu: al livello 3 è mascherato il data link. Le N-PDU son i PACCHETTI.

Solito ragionamento: Si piglian dati dallo T-PDU, si segmentano/oloborano, si mandan giù allo 2-PDU, e diventeran il loro payload.

- Funzioni: - instradamento (determinare quale ramo di uscita vada scelto)
- moltiplicazione (usare un unico collegamento a livello 2 per diversi flussi a livello 3)
- controllo di flusso: ottimizzazione delle prestazioni di traffic
- controllo di congestione: EVITARE IL RIEMPIIMENTO DEL BUFFER
- interoperabilità



Servizi
 - datagram → la sequenza dipende da alg. di instradamento
 - circuito virtuale → simile alla comm. di circuito: si crea un instradamento del a priori, e si instrada sempre lì. CREAZIONE, TX, RILASCIO.

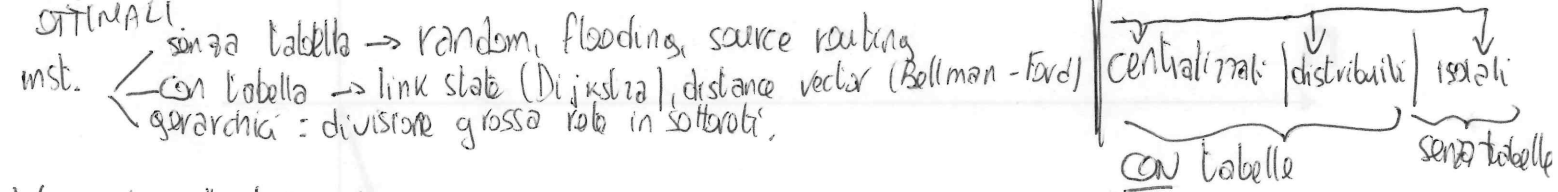
Un pacchetto "traccia" un circuito, e sul nodo si memorizzano informazioni. Tutti i pacchetti andran per sto circuito. Poi si rilascia.

Differenza: nel c.v., vi è la PREVENTIVA SEGNALAZIONE.

Nodi più difficili: tabelle di instradamento E DI INDIRIZZI. Nel caso di perdite, maggior controllo con CV (reset dei circuiti e avviamento di proc. di recupero).

DG: impossibile, tabella di recupero pacchetto.

Algoritmi instradamento: SEMPLICI (computazione), ROBUSTI (non bloccano la rete), STABILI (convergono), OTTIMALI.



Variante "selective": controllore che la destinazione non sia per caso il nodo adiacente (serve lista di adiacenza ai nodi) → in tal caso, trasmetter li' :-P

Random: casuale: semplice robusta, lento

Flooding: molto robusta semplice, veloce, MA RISCHIA DI BLOCCAR TUTTO.

Ricorda: o si limitano i "bounces" del pacchetto o loch.

Source routing: il nodo sorgente determina l'intero percorso
 path server: un server calcola e fornisce il percorso
 path discovery: mediante flooding proattivo e riscontri si determina il percorso

Instr. fisso: simile al circ. virtuale; si stabilisce il percorso UNA VOLTA PER TUTTE.

ALGORITMI (distanza minima):
 Distance vector: più semplice del link state; ogni nodo STIMA la distanza tra sé e la destinazione, memorizzando il PASSO successivo, ossia "verso quale nodo adiacente si deve andare". Ciò è semplice ma introduce errori: se A pubblica percorso verso C, i vicini non sanno se son inclusi da A; posson formarsi cicli. Il vettore di distanze vien mandato ai nodi adiacenti, che sommano la loro distanza calcolando la distanza della destinazione.

Altro problema: se un nodo va fuori servizio, capita che quello adiacente se ne accorge ma quello adiacente a quest'ultimo no. Capita che le distanze si aggiornano con l'adiacente all'adiacente, e si conta fino a ∞. SOLUZIONI: evitare di pubblicare le route attraverso le stesse interfacce da cui arrivano le slime originali. → SPLIT HORIZON

LINK STATE → misura

Ogni nodo calcola la propria distanza dai vicini e la manda IN FLOODING A TUTTI, con il "link state packet" → si possono individuare i percorsi o minima distanza verso ogni altro nodo della rete.

Svantaggio: richiede MOLTA MEMORIA e MOLTA ELABORAZIONE, per quanto rapido e stabile.

SEGNALAZIONE ripartire da p 278

→ scambio di info da nodo a nodo o da utente a nodo.

S. di utente: consentire interazione tra utente e rete

S inter-nodo: coordinare azioni dei nodi.

CAS: segnalazione e fonìa sullo STESSO CANALE } in o fuori BANDA FONIA

- CCS: canale dedicato per le segnalazioni.
- Segn. di utente: caso della fonìa, quelli per crear la chiamata
- di supervisione: richiudere la linea (prender la cornetta) o ~~se~~ agganciarla
- di indirizzo: far il numero
- di servizio (tam & annunci).

Il ~~di~~ internodo: quando i nodi devono collegarsi tra loro o segnarsi.

Rete di segnalazione di CAS, 3 elementi:

- SP: Signalling Point: ORIGINE E DESTINAZIONE della segnalazione
- STP: nodo di transito per segnalazione
- SL: COLLEGAMENTO DI NODI (SP o STP).

Reti:

associata: ogni canale, collegamento

quasi-ll: per la maggioranza dei canali

non-associata: SOLO collegamenti tra SP e STP, NO tra SP & SP.

SS7: Sistema di Segnalazione n. 7: sistema CCS più diffuso.

Numerico! → va a pacchetto: per ogni messaggio si ha 1 pacchetto.

Usa datagram come commutazione. 3 entità:

- SSP: Signal Switching Point: commutatori (centrali) che generano, commutano e terminano le chiamate. Sono punti TERMINALI delle rete SS7. CENTRALI TELEFONICHE
- STP: commutatori di pacchetto, INSTRADANO.
- SCP: Signaling Control Point: gestiscono servizi, tariffano, ec.

Ricorda SIP: Session Initiation Protocol (NO Pattavina: -C)

CLIENT che c'è da sapere di SS7 & SIP

ISDN (Integrated Digital Services Network)

Primo esempio di rete integrata (Integrata di Digital Services Network).
Sfruttare al meglio le infrastrutture presenti: integrazione FONIA e DATI BASSA VELOCITÀ.
→ inoltre trasmissione PURAMENTE NUMERICA, sia per dati sia per fonia

↳ servizi eterogenei → necessità di segnalazione ELABORATA.
Trasporto di informazione $\left\{ \begin{array}{l} \text{circuiti } (\cong 64 \text{ kbps}) \\ \text{pacchetto (frame relay)} \end{array} \right\}$ USO protocollo SS7 poiché in grado di segnalare.

2 (+1) tipi di canali: - B ("Boarer"): dati in BANDA FONICA/Video bassa velocità

(64 kbps) (codifica PCM)
- canale D (Data): 16 kbps o 64 kbps

interfacce $\left\{ \begin{array}{l} \text{BRI (Basic Rate Interface): accessi a v. contenute, utenza residenziale} \\ \text{PRI (Primary Rate Interface): utenza business.} \end{array} \right.$

BRI: due canali B e uno D; D → segnalazioni utente - rete

PRI: canali H (banda larga), e D per segnalazione

NT: Network Termination: dispositivo di accesso.

TE: Terminal Equipment: apparecchiatura utente

TE1: SOLO usati compatibili ISDN

TE2: NON compatibili → TA (terminal adapter) si adatta allo standard

NT1: presso sito di utente alimentazione clock
NT2: funzioni AGGIUNTE (commutazione locale, multiplexing...)
C'è uno schema con punti di riferimento, p. 332 libro (o 20 slide)

non fatto da p. 332 e p. 338. serve? :-C

Protocolli di linea: → p. 346

LAP-D: a livello di collegamento.

LAP-B →

LAP-D: trama Flag - Add - Control - Inf - FCS - Flag

Da LAP-D a LAP-B cambia poco.

Da LAP-B a HDLC cambia che LAP-B ha meno tipi di trame di HDLC.

DSL : Digital Subscriber Line

Usare doppini & portare in caso dell'utenza alte velocità.

Limite storico: banda del doppino.

Idco: TRAFFICO ASIMMETRICO (ADSL)

ATU (ADSL Transceiver UNIT)
 ATU-C (Central)
 ATU-R (Remote) : presso utente

Filtro splitter: separa banda audio da banda dati.

DSLAM: DSL Access Multiplexer ; regola l'accesso inet per tot doppini. (centralina)

PPP : protocollo che definisce il trasferimento di unità informative di livello 3 su una pluralità di linee. PROT. LIVELLO 2.

Usa trama definita su byte invece che su bit (HDLC usa BIT :c)

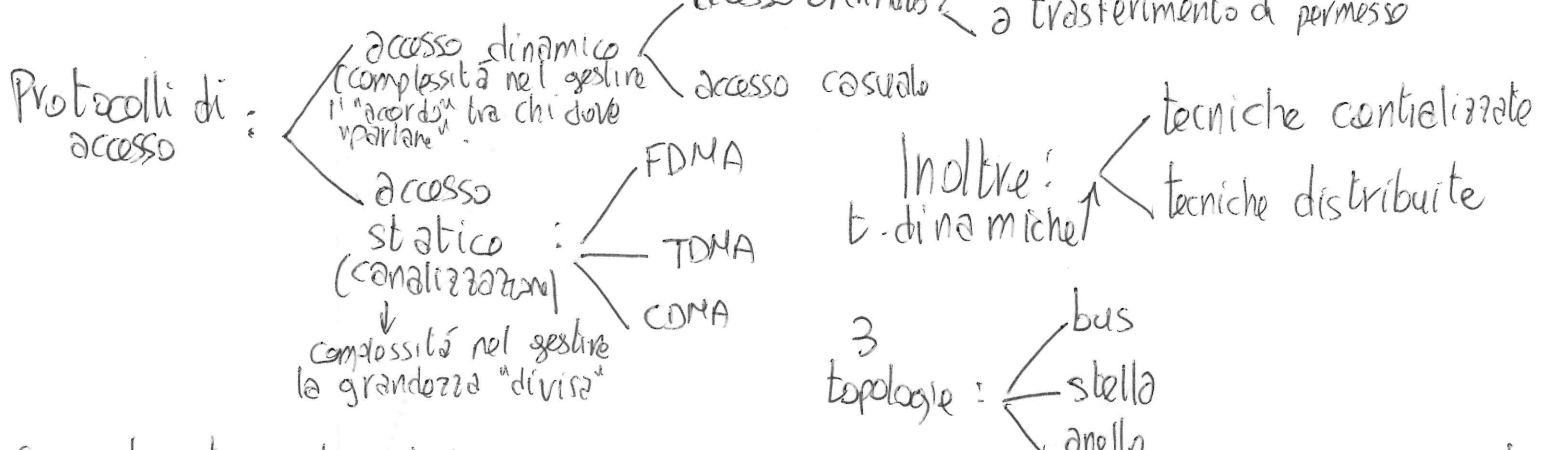
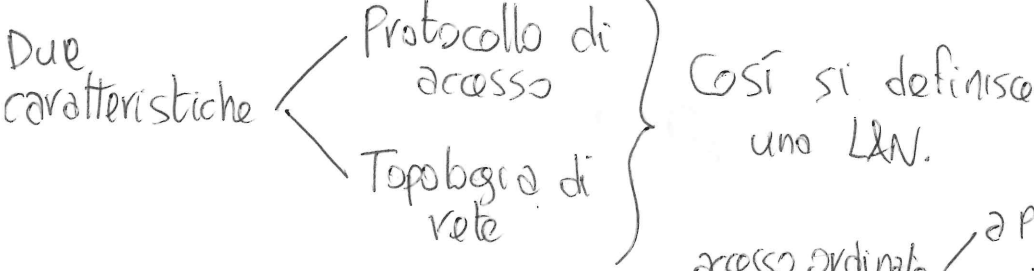
→ lunghezza di trama in byte

Pacchetto: Flag - Address - Control - Protocol - Payload - FCS - Flag
 TUTTI i
 Control:
 & punto
 punto: p1
 Tipo protocollo
 livello 3
 1500 byte
 SERVE LO
 STUFFING !!!

LEGGERE E TENERE PRESENTE SLIDE "COLLEGAMENTO".

LAN
A differenza delle WAN, operano su 100 m \approx 1 km; semplici, robuste, economiche, affidabili, ma con basse garanzie sulle prestazioni.

Una LAN per esser espansa richiede una RIDEFINIZIONE della topologia di rete, mentre un cambio di tecnologia semplicemente modificando terminoli, raramente l'infrastruttura.



Generalmente è la destinazione, che deve capire che le è indirizzata la trama!
Bus: mezzo trasmissivo condiviso, e si han "terminazioni" che ottimizzano il trasferimento dei dati.

Prenotazione: ciclo di 2 parti: nella prima si "prenota il diritto a parlare", dunque si trasmette (se si vuol parlare) una trama, si prenota, e si parla "a turno" impostando internamente il giusto tempo di inizio.

A contesa: si verifica se qualcun altro vuol parlare; se no si parla; se si, si cerca di rilevar collisioni, star zitti e ritentare dopo. CSMA/CD: Carrier Sense Multiple Access/Collision Detection.

Il "CS" si verifica guardando che nel canale ci sia solo il segnale TRAMATO DA ME!
Max 16 tentativi di TX. Tener presente P 431 \approx 432

Top ad anello: TRASFERIMENTO DI PERMESSO (token ring!)
Il token si trasmette da una stazione o un'altra, e chi lo ha può parlare! :-)
Il token VA GESTITO: non deve esser PERSO, non deve esser DUPLICATO.
Queste reti son poco affidabili.

Rivedere da p. 377

& partir da 433

Interconnessione reti locali

Due tipi:
+1

- Repeater: RIGENERATORE DI SEGNALE; funziona solo a livello FISICO; non attribuisce significato ai bit (EVENTUALMENTE CAMBIA FASENATO).
- bridge: liv. 2
- router o >: liv. 3

Bridge: più complesso del repeater: ELABORA LE TRAME. Interconnette LAN, o comunque reti con lo stesso protocollo di L. 2 (NAE), ma anche con DIVERSI MAC (translating bridge).
Fa instradamento anche se a livello 2, e facilmente.

Cosa fa: MEMORIZZA le trame, e le invia AL POSTO GIUSTO. Lo "switch" è un bridge ETHERNET.
Un bridge può unire i domini di BROADCAST (ins. indirizzi a cui vengono trasmessi messages broadcast), SENZA unire quelli di collisione: NO collisioni per broadcast os.
Tabella di inoltra: ind. MAC destinazione, porta a cui è connesso, tempo di vita in tabella. Quando non si ha il MAC in tabella si trasmette a tutti, tutti scartano e quello giusto si fa sentire, quindi si identifica porta ↔ mac.
Lo popolano avviene man mano che si manda roba per il bridge, che legge l'ISA.

LOOP: si usa lo spanning-tree.
3 porte: R (root): da bridge a root-bridge; D (Design): da L2 a root-b; B - Backup.

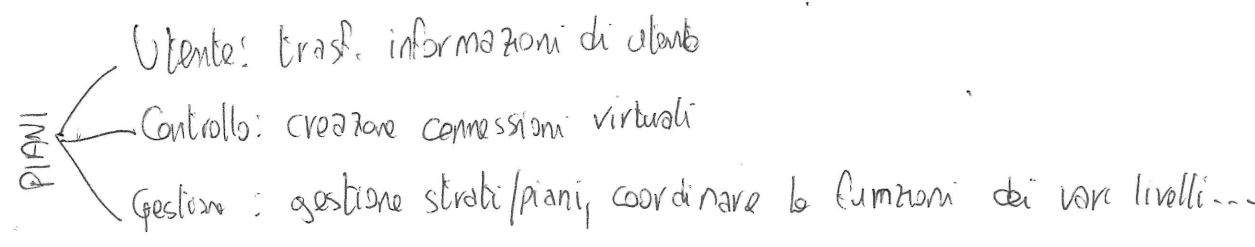
Router: lavora a livello 3; con PACCHETTI. Si deve instradare indipendentemente da livelli FISICI e DATA-LINK.
VLAN: risulta utile differenziare dei flussi, per controllo di congestione o altro; Si disgiunge il dominio broadcast delle reti, creando sotto-LAN virtuali, INDIPENDENTI DALLE TOPOLOGIE FISICHE.

ATM: Asynchronous Transfer Mode: reti ad ALTA VELOCITÀ e integrate - rappresentano l'evoluzione dell'ISDN. A PACCHETTI. Unica modalità di base di trasporto, per tutti i tipi di data.

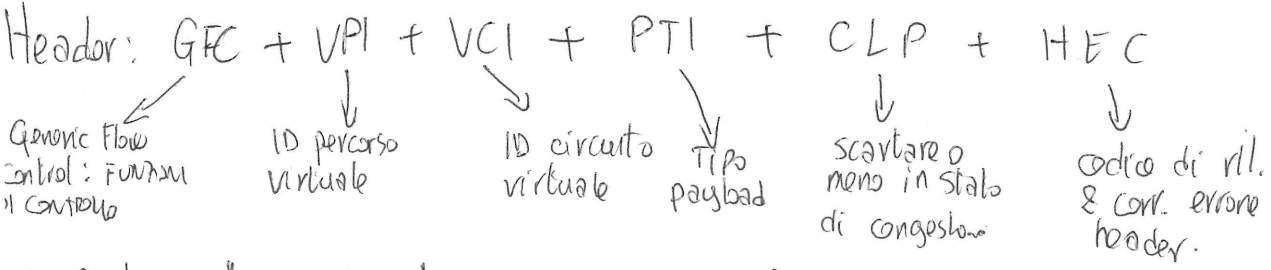
ATM: uso di CELLE, dunque pacchetti A DIMENSIONE FISSA; l'header è l'instellatura, il payload il dato. Si usa CIRCUITO VIRTUALE. Termine "ASINCRONO" indica il fatto che NON SI PREALLOCA STATICAMENTE LA BANDA, ma lo si alloca dinamicamente. La Tx NON è asincrona. 2 interfacce:
UNI: User Network Interface: modalità di lavoro utente es nodo ATM
NNI: Network Node Interface: interlavoro tra nodi.

Cella: header lungo 5 byte - il payload è fisso dunque i sistemi avranno un RITARDO di pacchettizzazione al fine di non aver pacchetti grossi ma anche troppo piccoli (ciò causa ritardi), 48 byte.
Connessioni virtuali: VC: virtual circuit. Ciascuno è identificato da un VCI.
VP: Virtual Path: unione di vari VC. Ciascun VP ha un VPI, diversi VCI associati.
(VPI, VCI): ID di una con. virtuale.

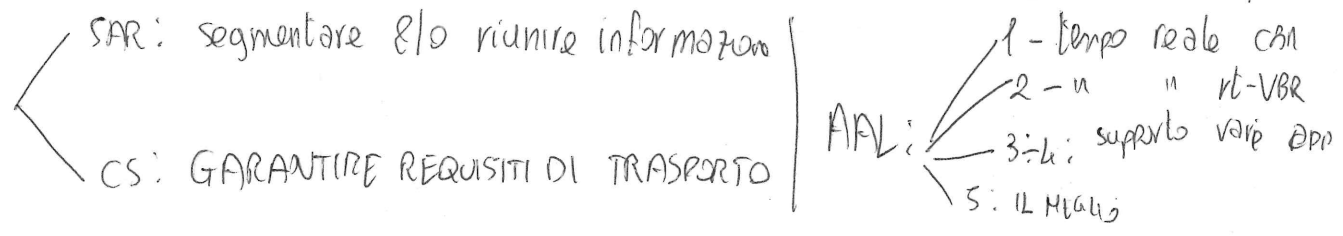
Architettura: PIANI & STRATI.



PM: interfaccia di fisico; TC: conversione cella -> bit
ATM: definizione cella (5 + 48 byte), contestualizzata nella connessione virtuale.



AAL: "arricchire" ATM per meglio fornire servizi, a livelli superiori.



AAL 5: ATM Adaption Layer 5

Riduce la complicazione riducendo i campi di controllo, supponendo che ATM fornisca servizio di alta qualità.

- CPCS - PDU: la PDU.
- Payload: max lunghezza 65365 byte, resa MULTIPLA di 48 byte e con padding (in modo da non aver celle mezzecelle).
- UV: indicazioni da utente e utente
- CPI: Common Part Indicator
- LEN (Length)
- CRC

Architettura TCP/IP

ATM: architettura troppo complicata: si usa su reti estese, ma la complessità non ha permesso la diffusione nel LOCAL.

TCP/IP: definisce modalità di trasferimento utilizzabile in OGNI APPARECCHIATURA.

→ INTERNET: TCP/IP; TCP: Transport Control Protocol; IP: Internetworking Protocol

IP sta SOPRA lo strato di rete, ed è "client" rispetto ad esso (usa i suoi servizi).

Sotto IP, che dunque lavora con DIVERSE RETI, vi sono: network, data-link e physical layers.

4 servizi fondamentali: RMP (controllo), OSPF (Open Shortest Path First): instradamento; ARP (Address Resolution Protocol): da un indirizzo IP tira fuori un MAC; RAPP: da un MAC, IP.

Nota: si usa il livello 3 da "server", ma non sono livello 4: sono nei router, che lavorano a liv. 3.

Strato di rete - IP (e dunque anche sopra).
INDIRIZZO: net-ID: identificatore della rete a cui l'host è connesso
IP: host-ID: identificazione dell'host rispetto alla rete
Subnet-mask: x.y.z.w, tutti "1" identificano la net-ID, gli "0" la porzione di indirizzo che fa da host.

5 classi di indirizzi (3+2):
A: poche reti, tanti host: 7 bit rete, 24 bit host
B: medio: abbastanza reti e host
C: tante reti, pochi host
D: indirizzi usati per servizi multicast
E: indirizzi riservati per uso futuro

net-ID: classe A, 1 byte; Classe B, 2 byte; Classe C, 3 byte (di 4)

Le reti si identifica con host "0" e il BROADCAST con "1" (tutti "1").

IANA (Internet Assigned Number Authority) → "registries", organizzazioni che suddividono grossi blocchi di IP in altri più piccoli.

Indirizzi PRIVATI: utilizzabili LOCALMENTE, internamente. NON possono usare così all'esterno!

3 blocchi di ind. privati: 10.0.0.0/8, 172.16.0.0/16, 192.168.0.0/24

Subnetting: suddividere una rete di classe generica in sottoreti senza interessare l'instradamento da Internet verso l'interno della rete. → Subnet-mask

Questo aumenta la possibilità di utilizzazione della rete, dagli indirizzi.

VLSM: sottoreti con DIVERSO NUMERO DI HOST ossia con diversi bit di mask a "1".

NAT: non è detto che tutti DEBBANO per forza uscire dalla rete su internet con un proprio indirizzo: molto del traffico potrebbe essere interno. IP privati: macchine INTERNO, IP pubblici: macchine ESTERNO. Il NAT permette di far ciò.

NAT: associa indirizzi presenti in inet con indirizzi INTERNI ALLA RETE, sostituendo gli indirizzi IP daagrammi esterni con uno noto, ~~esterno~~ redirigendo così. Da dentro a fuori: un PC di una LAN vuol andar su internet, va al gateway che con la NAT, cambia l'indirizzo in uno pubblico esterno indirizzando così. Si NATea anche su indirizzi TCP/UDP (porte): NATP.

IP: connection-less (no uso di connessioni prima del trasporto, datagrammi). Interagisce con TCP/UDP con primitive send e deliver. send: controllo informazioni a IP per il trasporto, deliver annuncia l'arrivo di un'unità dati.

Header pacchetto IP:

- Version: versione prot. usato (IPv4)
- HLLEN: lunghezza header
- Type of service: dal livello trasporto, si han richieste
- Total Length: lunghezza datagramma IP

- ID: identifica il datag.
- flag: si decide se il datag. può essere frammentato & se è l'ultimo.
- frag. offset: pos. del frammento nel datagramma.
- TTL: Time to live

- Protocol: specifica prot. livello superiore
- Checksum: ril. errore
- Source/Dest IP Address
- Options: altre op. opzionali

Padding: rende il header multiplo di 32 bit.

Sarebbe bello avere payload >> header, per aumentare eff., ma spesso FRAMMENTAZIONE SERVE, per far parlar reti eterogenee.

MU: Maximum Transfer Unit: max valore di informazione (payload) + controllo (datagram).
 SE DF=1, NON si frammenta. **GUARDA**

MUORA FINO A A 507

Due classi di prot. di instradamento:

- IGP: interni a un sistema autonomo
- EGP: Exterior gateway protocol: TRA ROUTER APPARTENENTI A DIVERSI SISTEMI
- BGP: ↑ principale EGP.

Inoltro:

- diretto: il datagram può essere recapitato direttamente
- indiretto: bisogna trasmettere a un altro router.

 Si trasportan pacchetti a IP con trama, dunque serve corrispondenza tra liv. 2 e 3: ARP.

Si fa un AND di netmask e tabella di un router, e si confronta; se si trova corrispondenza, consegna diretta. Se no, next-hop.

ARP: un computer lancia alla sottorete una ARP request, inviando un IP; se questo c'è, risponderà con il suo IP e con la sua MAC address. Nella request c'è il sender MAC address e il suo IP. Di solito temporaneamente, si fermano in memoria le associazioni MAC-IP.

RARP: assegnazione IP: dato il MAC address di una macchina (per esempio senza disco fisso) con RARP può farsi assegnare.

RIVEDERE P 384

Strato di Trasporto (TCP/UDP)

T-PPU: SEGMENTI; si identificano con IP+PORTE (poiché IP gestisce diverse applicazioni, anche sullo stesso host). → SOCKET: IP+PORTA

Porte: Well-known: 0 ÷ 1023: bendefinite + dynamic: TOTALMENTE LIBERI
Reserved: 1024 ÷ 49151: usabili senza vincoli.

UDP: funzioni ridotte → molto semplice. NON NUMERATI → no poss. di controllo errori di perdita.

header: SOURCE PORT, DEST. PORT, UDP length, checksum (opzionale)

Non affidabile ma molto semplice, dunque veloce.

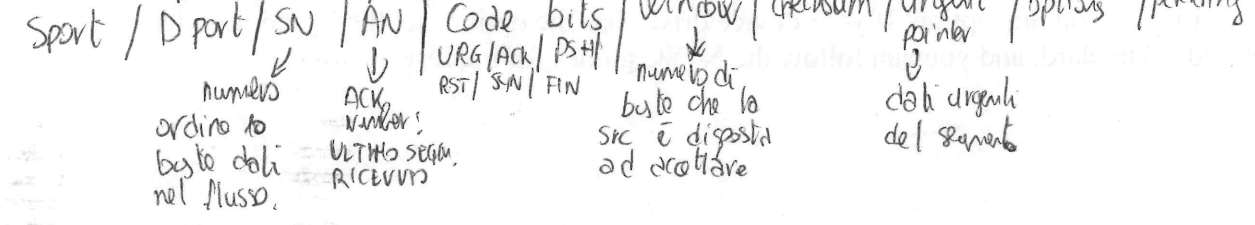
TCP: AFFIDABILI ma molto più complesso di UDP.

Nonostante IP sia CONNECTION LESS e CON PERDITE, TCP colma e fa in modo da far vedere alle applicazioni la possibilità di servizio connection-oriented e affidabile.

Gestisce il controllo di flusso, di congestione, permette di ordinare pacchetti fuori sequenza, dunque usa numerazione. La numerazione è byte per byte.

↓
come quelli di IP.

Header (p 594-595):



Three-way handshake: una sorgente A manda richiesta di connessione a B, 1) SYN }
Comprensiva di ISN. B, se accetta, specifica il suo ISN e manda a A. A conferma. 2) SYN-ACK }
3) ACK }

ISN: Initial Sequence Number: si fa in modo da avere intervalli di numerazione non sovrapposti, specificando da dove si parte. Può capitare che una linea parli con la vecchia numerazione facendoci interferenza. Si usa un'assegnazione pseudocasuale, basata su di un contatore a 32 bit che continua e cambia e del quale si prenderà il valore che assume all'apertura della connessione.

Si introduce un tipo di TTL: MSE (Maximum Segment Lifetime), per evitare che si ricevano buste rimasti in giro per la rete. (2 min).

Può esserci, a causa del full-duplex, una collisione al momento di iniziare la connessione; se le porte son uguali, UNICA connessione; se diverse, due conn. distinte, Le conn. bidirezionale inizia per ordine di una delle due. Una manda pacchetto con FIN=1, l'altra dà ack e la prima II° conferma mandando FIN=1, dunque le 1° FIN=ACK=1.

MSS: massima dimensione segmento. $MSS = \frac{MTU + \text{header IP} + \text{header TCP}}{(IP)}$

Proprietà TCP

- Si immagazzinano pacchetti anche fuori sequenza, se ci stan in finestra
- Solo ACK, non NACK. AN: x+L
- Indica che fino a x, tutto bene.
- o risposta singola, o delay ack (ogni 2 o ogni un certo tempo).
- ANCHE PIGGYBACKING.

Né go-back-N né Sel. repeat

Si accettan pacchetti fuori seq. non si può dire qual è il seqm. mancante.

Con window si specifica quanti dati si vogliono accettare.

Fast retransmit

Mecanismo alternativo di ritrasmissione: dati 3 ack riguardanti un pacchetto già recapitato, con lo stesso AN, si interpretano come un NACK; si fa evitare l'uso del delayed ack. Si fa in modo da precedere il timeout, anticipando la ritrasmissione del segmento.

Controllo di congestione

Il parametro "window" permette al ricevente di stabilire quanto può accettare "Window" stabilisce le posizioni libere nel buffer di ricezione. Questo si può far per controllo di flusso. Esiste una 2a finestra "congestion window", che permette di far controllo di congestione.

Un algoritmo di controllo vede quanto la rete riesce a mandare, dunque si cambia dinamicamente il valore della finestra in trasmissione. Si regola così la Tx.

Congestione: due tipi di operazione
- slow start → si parte piano a trasmettere, e poi si aumenta fino a raggiungere un livello accettabile (aumento moltiplicativo)
- congestion avoidance → zona in cui il traffico può aumentare ma molto meno. (1 MSS per RTT).

slow start: si parte con finestra unitaria e si raddoppia per ogni RTT.

Timeout: quando c'è perdita, si ritrasmette e la congestion window si riporta a 1.

Retrans: nel caso la ritrasmissione sia dovuta ai 3 ack, si riduce la cwnd alla ssthresh; con timeout, a 1.

Urgent (URG): si dà priorità ai dati dell'urgent pointer (rSN).

PSH: FORZARE la trasmissione, INDIPENDENTEMENTE DALLA FINESTRA.

Selective ack: ack NON PER FORZA ADIACENTE.

Esercizi reti telematiche - libro

①

1) Nota B_{N-1} , vel. max entità strato $N-1$; N -PDU, dim. fissa a p byte; $(N-1)$ -PDU, c byte

Cambia la $(N-1)$ -PDU, due casi; fissa s byte, variabile, $1 \leq s \leq p$ byte.

Dallo strato inferiore, $N-1$, la N -PDU genera un numero di $(N-1)$ -PDU pari a $\lceil \frac{p}{s} \rceil$:

p è la dimensione della N -PDU, s i dati contenuti nella $(N-1)$ -PDU; passando da superiore a inferiore, le $(N-1)$ -PDU verranno costruite considerando l'intera PDU dello strato N come informazione, dunque "impacchettandola" ulteriormente nella $(N-1)$ -PDU, aggiungendo le info. di controllo; si generano $\lceil \frac{p}{s} \rceil$ $(N-1)$ -PDU, di dimensione $(c+s)$ byte;

Nel II° caso, capita che avremo tutte meno una PDU (di $N-1$) di lunghezza MASSIMA, e una di lunghezza s , dove $s \leq p$. Dunque, si generano PDU pari a $\lceil \frac{p}{s} \rceil (c+s) - \lceil \frac{p}{s} \rceil s + 1$

- 1) RIVEDERE LA TEORICA, CAPIRE SE HO SCRITTO CORRETO
- 2) CAPIRE L'ULTIMO RAGIONAMENTO; perché $\lceil \frac{p}{s} \rceil (c+s) + p - \lceil \frac{p}{s} \rceil s$ e non $\lceil \frac{p}{s} \rceil (c+s) + p - s$?

2) File 9300 byte, no errori, $T_p = 5ms$, max dim pack: 1500 byte

→ 6 pacchetti da 1500 byte, 1 da 300 byte.

$V_{Tx} = 1 \text{ Mbit/s} \rightarrow \frac{1500 \cdot 8}{10^6} = 12 \text{ ms (1500)}; 4 \text{ ms (300)}$

→ $5 \text{ ms} + 12 \cdot 6 + 4 = 81 \text{ ms}$

3) Come prima, 6 da 1500 e 1 da 300. S manda a N , N a D in pipe. Dovrò dunque mandare tutto il primo pacchetto, poi gli altri, e infine l'ultimo.

Proviamo:

$\frac{1500 \cdot 8}{10^6} = 12 \text{ ms}; \frac{300 \cdot 8}{10^6} = \frac{1}{3} \cdot 12 = 4 \text{ ms}; T_{p1} = \frac{l}{2 \cdot 10^8} = 2 \text{ ms}; T_{p2} = 3 \text{ ms}$

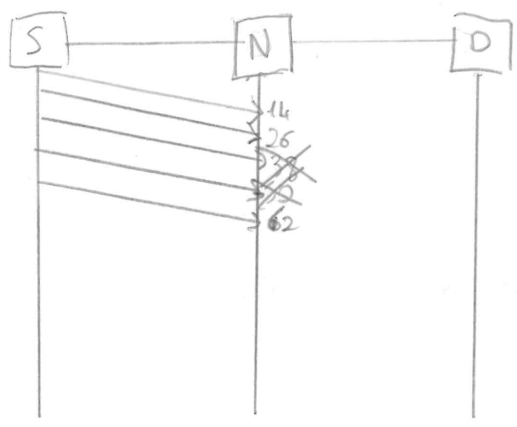
A $12ms + 2ms$ il pacchetto è giunto su N , dunque ancora $12ms + 3ms$ per venir trasmesso al II° canale da N ed esser propagato. → 25ms.

Altri 5 pacchetti che arrivano dopo $12 \cdot 5 = 60ms$ (il pipe permette di contare una volta sola il tempo di propagazione), dunque a $89ms$ son arrivati 6 pacchetti.

Altri $4ms$ per arrivare al nodo N per il 7°, ma una nota: questo pacchetto è più corto, dunque ci si impiega meno a farlo arrivare a N ; l impiega $89 - 3 = 86ms$ per lanciare l'ultimo pacchetto, D $86 - 12 = 74ms + 4ms = 78ms$ per lanciare il 7°.

Questo pacchetto 7° rimane in memoria fino a 86 , dunque $4ms$ per esser messo da N sul II° canale, 3 per esser propagato, dunque a $93ms$ finisce tutto ($86 + 4 + 3$)ms.

4) N al più ora può immagazzinare 2 pacchetti, compreso quello in tx. 2ms I° chan, 3ms II° chan; 1500 byte, file da 9500 → 6 da 1500 e 1 da 500.



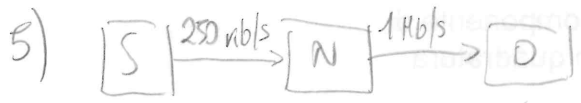
Allora: per il V 1500,

$$\frac{1500 \cdot 8}{1M} = 12 \text{ ms}; \quad \frac{1500 \cdot 8}{250k} = 4 \cdot 12 = 48 \text{ ms}.$$

Dopo 2+12 ms, quindi 14 ms, arriva al 1°; dopo altri 12, il II°; (26 ms);

da 14 ms, ce ne van 48 prima che il pacchetto si è trasmesso; da 26, arriva un pacchetto a 26+12=38 che vien scartato, perché il primo spazio in mem. è a 14+48=62 ms; 38+12=50, 50+12=62 → pigli

Dopo 62, impiegherò altri 48ms per inviare il II° pacchetto (quello giunto a 26 ms), arrivando a 110 ms; arrivati qui, trasmetto il 3° pacchetto, scartando gli altri. Di 9500 byte ne mando 3 · 1500 = 4500, ossia il 48% circa.



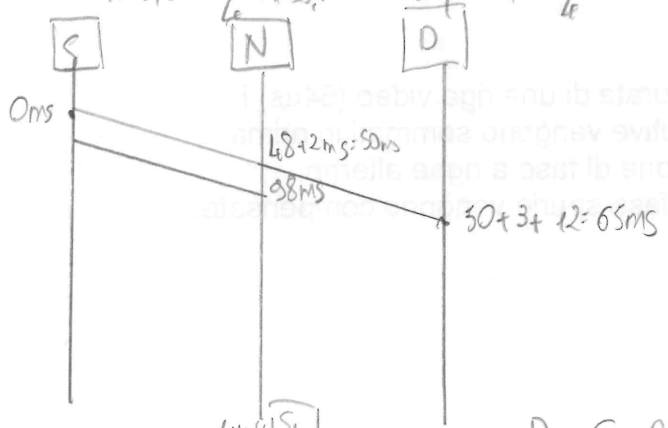
N può memorizzare al più 2 pacchetti; 2ms I° chan, 3ms II° chan

$$\rightarrow \left\lfloor \frac{9500}{1500} \right\rfloor = 6; \quad 9500 - 6 \cdot 1500 = 500 \rightarrow 6 \text{ pacchetti da } 1500, \text{ uno da } 500$$

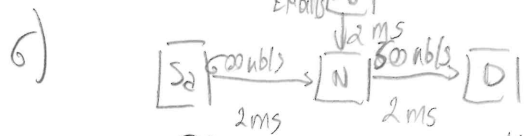
$$\rightarrow T_{1500,1} = \frac{1500 \cdot 8}{250 \cdot 10^3} = 48 \text{ ms}; \quad T_{500,1} = \frac{500 \cdot 8}{250 \cdot 10^3} = 16 \text{ ms}$$

$$T_{1500,2} = \frac{1}{4} T_{1500,1} = 12 \text{ ms}; \quad T_{500,2} = \frac{1}{4} T_{500,1} = 4 \text{ ms}$$

S & D sono senza prot, N è "store & forward".



Il flusso di pacchetti tolti da N è più rapido di quelli inviati; NON SI PERDONO PACCHETTI.



Da Sa & Sb parte un file di 6000 byte, pacchetti da 1500 byte.

Allora: $\frac{6000}{1500} = 4 = 4$ pacchetti. Ogni pacchetto esce nel seguente tempo: $T_A = 20 \text{ ms}$; $T_B = 12 \text{ ms}$;

Dopo 12 ms + 2ms = 14 ms, il I° pacchetto è da Sb a N; da qui, 24+2 ms e arriva a D; a (20+2)ms da Sa arriva il I° pacchetto su N; questo finisce su D dopo 24 ms (ora è inglobato il 12ms), dunque a (14 + 2 · 24 + 2) ms ho mandati 2; uno da Sb, uno da Sa. Mentre passano i 24ms, arriva un pacchetto da un chan o dall'altro; per mandare il file "B" devono aspettare 6 · 24 + 14 + 2 (4 pacchetti di B più 2 di A), per "A" 8 pacchetti.

7) $[S] \rightarrow [N_L] \rightarrow [N_R] \rightarrow [D]$ Prop. & elaborazione trascurabili. (3)

a) $n = 40$ byte \rightarrow Pac. = 1040 byte $\rightarrow T_{Tx,1} = \frac{1040 \times 8}{10^6} = 8,32$ ms

Osservazione: un pacchetto entra e fa tutto il giro da cima a fondo; gli altri fan pipeline, dunque il temp dei canali va calcolato solo per 1 pacchetto; gli altri 6 seguono.

$\rightarrow T_{Tot} = T_{TxL} \cdot C + 7 \cdot T_{TxL} = 83,2$ ms

b) Stesso ragionamento con header di 20 byte, MA tempo equivalente; $T_{Tx2} = \frac{1020 \cdot 8}{10^6} = 8,16$

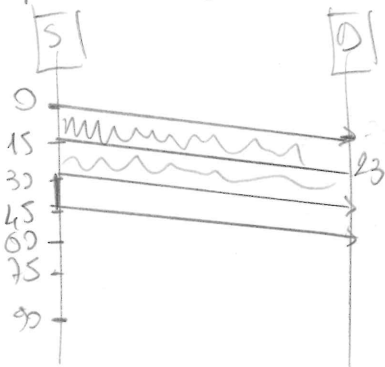
$T_{Tot2} = T_{CV} + T_{Tx2} \cdot C + 7 \cdot T_{Tx2} = 4,6$ ms + T_{CV}

T_{CV} deve esser uguale alla differenza: $|T_{Tot1} - T_{Tot2}| = 0 \rightarrow T_{CV} = 1,6$ ms.

27) 14 kbyte \rightarrow 9 pacchi da 1500, 1 da 900 byte. $T_p = 8$ ms.

Cono all'inizio pacchetti 0-5; $\frac{1500 \times 8}{8000 \times 3} = 15$ ms; $\frac{900 \times 8}{8000 \times 3} = 9$ ms.

Trasmetto da 0 a $15 \times 6 = 90$ ms, ricevo da 23 a 38 ms.



Trasmissi: perdo su [S]; dunque, ho che perdo il pacchetto da 30 a 45: il (2). Dalla ricezione di L₁ a 30+8 ms, altri 8 ms e arriva l'ack, che fa spostare, per $t = 46$ ms, la finestra; la finestra da $t = 46$ ms ha dentro 2, 3, 4, 5, 6, 7: il Tx va avanti a trasmettere fino alla fine di ciò che c'è in finestra, ma essa NON AVANZA, dunque $8 \times 15 = 120$ ms; + 20, 30 ms. Da qui si fa riportare la finestra, e a $320 + 8 + 7 \times 15 + 9 = 442$ ms.

8) Usando la teoria, $\frac{600m}{c} = 3 \mu s$; il I° pacchetto giunge su B dopo $t_0 + 3 \mu s$, e B parte a trasmettere dopo $t_0 + 2 \mu s$; B non può ancora sapere che il canale è occupato dunque c'è collisione. Non c'è un meccanismo che possa fermare la Tx, dunque il canale sarà occupato anche ci sarò trama:

$$\left[\frac{400 \times 8}{10 \cdot 10^6} \right] \circ \left[\frac{64 \times 8}{10 \cdot 10^6} \right] ? \text{ La } I^a : 2 \mu s + \frac{400 \times 8}{10 \cdot 10^6} = 322 \mu s$$

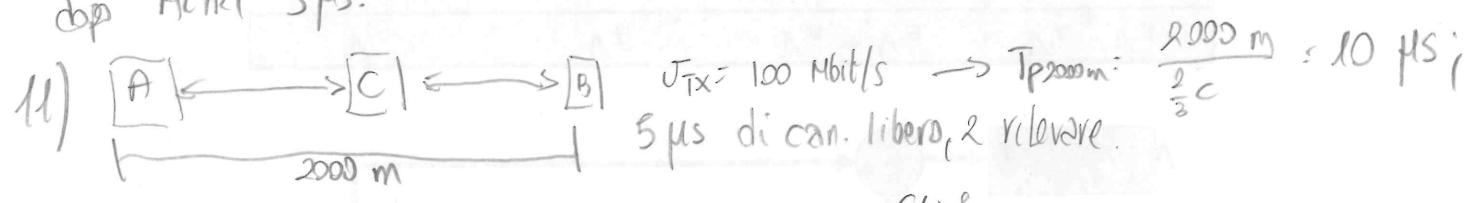
Questa è la durata della trama; C sente il chan libero dopo $t_0 + 322 \mu s + \frac{L}{2} T_p = t_0 + 322 \mu s + 1,5 \mu s$.

9) $T_{pAB} = 3 \mu s$ come prima; trama da 64 byte; questo significa che il tempo di invio è $\frac{64 \times 8}{10 \cdot 10^6} = 51,2 \mu s$; che capita?

Beh: da A parte la trama; il I° pacchetto arriva a C dopo $1,5 \mu s$, a B dopo $3 \mu s$; questo fa vedere la Tx al canale, sia a B sia a C; dopo $51,2 \mu s + 1,5 \mu s$ C vede tutto vuoto, e altri $5 \mu s$ per accettarsi; B dopo $51,2 + 3 \mu s$ vede tutto fermo; a $51,2 + 1,5 + 5 \mu s$, C parte a trasmettere, e il I° pacchetto va a B dopo $1,5 \mu s$. Al contempo B ha lo stesso comportamento, solo $1,5 \mu s$ dopo (a causa della maggior distanza da A), dunque B parte a trasmettere PRIMA AVENDO ARRIVATO IL PACCHETTO DA C.

10) Ora abbiamo $\frac{6}{20}$ lo spazio di prima: $\frac{2000}{213c} = 10 \mu s$
A ha trama da 64 byte pronta da t_0 , B da $t_0 + 5 \mu s$

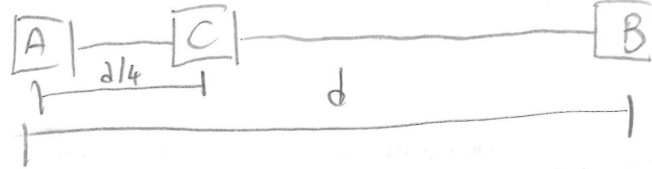
$\rightarrow T_B = 322 \mu s$; $T_A = 51,2 \mu s$
A $t_0 + 5 \mu s$ da A parte il I° pacchetto, che arriva in B dopo $10 \mu s$
 $\rightarrow t_0 + 15 \mu s$; su B $t_0 + 5 \mu s + 5 \mu s$ (canale vuoto), dunque per $t_0 + 10 \mu s$ parte a trasmettere. B a $t_0 + 15 \mu s + 2 \mu s$ si accorge di avere altro segnale, dunque smette di trasmettere; A a $t_0 + (10 + 10 + 2) \mu s$ si accorge di aver segnale di B e smette di trasmettere. C vede il chan vuoto a $(22 + 5) \mu s$, e completamente vuoto dopo ALTRI $5 \mu s$.



\rightarrow A ha 64 byte pronti, dunque a $T_A = \frac{64 \times 8}{100 \cdot 10^6} + 5 \mu s$ invia. $\rightarrow T_A = 10,12 \mu s$.
L'altra, B è $\frac{400 \times 8}{100 \cdot 10^6} = 32 \mu s$, pronta a $t_0 + 5 \mu s$.

Cosa capita? A $t_0 + 10 \mu s$ B trasmette, A a $5 \mu s$. Dopo $10 \mu s$, in B inizia la collisione: $5 \mu s + 10 \mu s = 15 \mu s$; a $15 \mu s$, B smette di trasmettere. A si accorge della collisione $10 \mu s$ dopo che B trasmette, dunque $20 + 2 = 22 \mu s$ per smettere.
C vede libero dopo $23 \mu s$. Prima lo vede libero da $0 \mu s$ a $10 \mu s$.

12)



$T_A = 64 \mu s$ (l₀)
 $T_B = 80 \mu s$ (l₀ + 5 μs) $T_P = 5 \text{ ns} \cdot d$

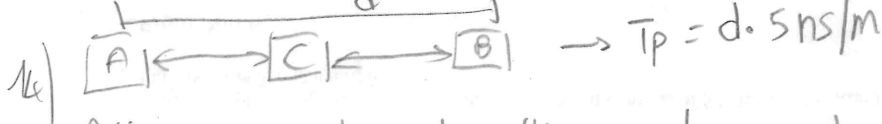
Il primo punto, A₁ manda un pacchetto che arriva a B a $5 \text{ ns} \cdot d$; Si suppone che la collisione ci sia, dunque che entro $l_0 + 5 \mu s + 5 \mu s$ non sia ancora arrivato nulla. $T_{P \min} = 5 \mu s \rightarrow d_{\min} = 1000 \text{ m}$. B rileva la collisione dopo $l_0 + T_P + 2 \mu s$; A rileva la collisione dopo $l_0 + 5 \mu s + T_P + 2 \mu s$; questa è meno di un fatto! che il fib sia così lungo da non far tornare indietro dati: mentre A ne trasmette ancora. A finisce di trasmettere a $64 \mu s$. B a $80 + 5 \mu s$. Entro i $64 \mu s$ di A₁ dove arrivare qualcosa di B. $64 \mu s = T_P + 2 \mu s + 5 \mu s \rightarrow d = \frac{(64 - 7) \mu s}{5 \text{ ns}} = 11,4 \text{ km}$. Con questo d, B si accorge della collisione a $T_P + 2 \mu s = 59 \mu s$; A $59 \mu s + \frac{3}{4} T_P$, C vedrà il canale libero.

13) Tipo il precedente: ora si chiede quale deve essere la minima dimensione delle trame.

$T_{\text{trama}} = 4,5 \mu s$; $T_A = T_B = D \cdot 0,8 \mu s$; per cercar la trama minima, bisogna usar il caso pessimo: il fatto che B inizi a trasmettere APPENA ARRIVA QUALCOSA DA A. Questo implica che quando la trama parte, essa deve durar abbastanza da rimanere nel canale in modo che anche A possa sentire la collisione, dunque che il segnale di B torni indietro mentre A trasmette ancora, + 2 μs:
 $T_{\text{rx min}} = 2 t_{\text{trama}} + 2 \mu s = 9 \mu s + 2 \mu s = 11 \mu s$; $T_{\text{rx}} = \frac{B \cdot 8}{10^6} = 11 \mu s \rightarrow D_{\min} = 13,75 \text{ byte}$

A $D_{\min} = 55 \text{ byte}$; $10 D_{\min} = 137,5 \text{ byte}$.

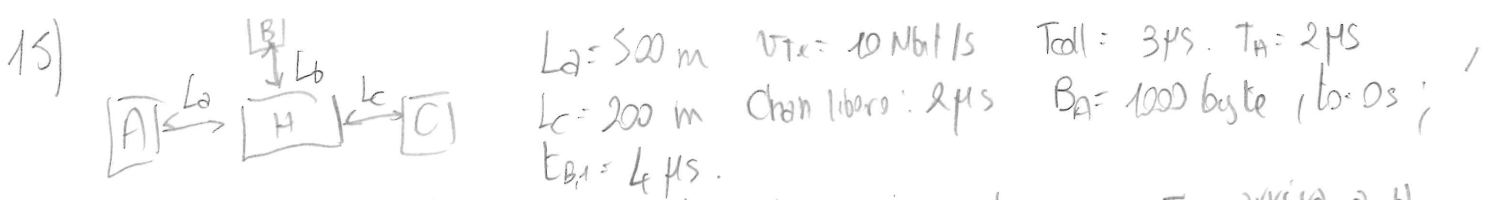
→ A $l_0 + 4,5 \mu s$ arriva il 1° bit, dunque la collisione è rilevata. (+ 2 μs) = $l_0 + 6,5 \mu s$



Allora: B rileva la collisione almeno dopo $l_0 + 4 \mu s + 2 \mu s = l_0 + 6 \mu s$. B smette di trasmettere, e dopo $l_0 + 6 \mu s + T_P + 2 \mu s$ A dove ancora trasmettere. Vogliamo il valore MASSIMO. $T_{\text{rx A}} = 51,2 \mu s$; $T_{\text{rx B}} = 160 \mu s$.

Al più, B rileva a $164 \mu s$ (non 160, perché trasmette da l_0 a 164 , poi lascia il canale libero).

A, invece, "sente" che B parla sopra di sé a $l_0 + 4 \mu s + T_P$. Ma A trasmette da l_0 a $l_0 + 51,2 \mu s$! Al più, d dovrà esser tale da far sentire, per $t = 51,2 \mu s - 2$, il proprio contributo; questo è l'ULTIMA collisione, dato che B parla dopo e $T_{\text{rx A}}$ è più corto. Se l'ultimo bit su A da B deve arrivare a $(51,2 - 2) \mu s$, dove PARTIRE a $51,2 \mu s - 2 \mu s - T_P$. Questo è l'istante in cui B blocca tutto per collisione. Si fa: $l_0 + 51,2 \mu s - 2 \mu s = l_0 + 4 \mu s + T_P$: poniamo uguali il T impiegato per trasmettere ad A il 1° pacchetto e l'estremo necessario per la collisione → $T_P = 45,2 \mu s$. → $d = 9,04 \text{ km}$. A smette di trasmettere a $51,2 \mu s$, dunque a $51,2 + \frac{T_P}{2}$ C sente libero.



Dunque: il 1° pacchetto di A parte a $t = 0 + 2 \mu\text{s}$, dunque a T_{LA} arriva a H_1 .
 A per T_A rigenera, si passa per L_b o si giunge a B.
 $T_{LA} = \frac{500 \text{ m}}{\frac{2}{3}c} = 2,5 \mu\text{s}$. $T_A = 2 \mu\text{s}$; Il primo pacchetto arriva a:

$T_{AB} = 2,5 \mu\text{s} + 2 \mu\text{s} + T_{LB} = 4,5 \mu\text{s} + T_{LB}$. Per evitare collisione, L_B deve essere tale da avere:

$$4,5 \mu\text{s} + T_{LB} \leq 4 \mu\text{s} + 2 \mu\text{s} \rightarrow T_{LB} = 1,5 \mu\text{s} \rightarrow 2 \cdot 10^8 \cdot 1,5 \cdot 10^{-6} \Rightarrow L_B = 300 \text{ m}.$$

b) $L_B = 350 \text{ m}$: ci sarà collisione.

Per $t = 4,5 \mu\text{s} + \frac{350 \text{ m}}{\frac{2}{3}c} = 6,25 \mu\text{s}$, arriva il 1° pacchetto su B; B si ne accorge a $9,25 \mu\text{s}$ o smette di trasmettere. Ora: il 1° pacchetto da B ad A parte a $2 \mu\text{s}$, dopo $1,75 \mu\text{s}$ arriva all'Hub, dopo 2 ne esce, dopo $2,5 \mu\text{s}$ arriva ad A, e A si sblocca: $15,25 \mu\text{s}$ → FAR MOLTA ATTENZIONE A "DA QUANDO SI COMA".

→ Fino a $15,25 \mu\text{s}$ A deve trasmettere

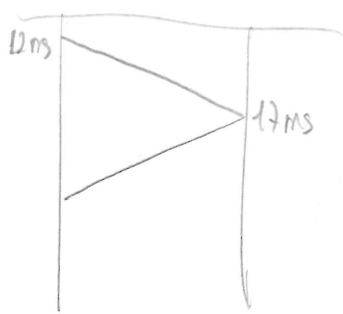
$$\rightarrow B_A = 15,25 \mu\text{s} \cdot 10 \text{ Mbit/s} = 152,5 \text{ bit} \approx \underline{19 \text{ byte}} \quad (0,20).$$

16)

17) File di 1 Gbyte, 10000 km di distanza, $W_{Tx} = 10 \mu\text{byte}$;
 $T_{PR} = \frac{10 \cdot 10^9 \text{ m}}{2 \cdot 10^8 \text{ m/s}} = 0,05 \text{ s}$; il Round Trip Time minimo è il tempo necessario a contare l'istante di Tx del 1° bit e quello di completa ricezione del ricevente $\rightarrow RTT = 2T_{PR}$

$V_{max} = \frac{10 \cdot 10^3 \cdot 8}{0,1} = 800 \text{ kbit/s}$

18) Canale con $V_{Tx} = 1 \text{ Mbit/s}$; file 9500 byte. Stop & Wait; 40 byte di instanziazione file da 1500 byte; ACK dim. trascurabile. $t_{p, \text{chan}} = 5 \text{ ms}$



Dunque: avremo 12ms per i pacchetti da 1500, e $9500 - (6 \times 1500) = 740$; un pacchetto da $740 + 40 = 780$ byte $\rightarrow \frac{780 \times 8}{1 \cdot 10^6} = 6,24 \text{ ms}$

Allora: da $t=0$, passano 12ms e parte il 1°, che arriverà a 17ms, dato le idealità, l'ack torna a 22ms. Da qui altri $12 + 5 \rightarrow 39 \text{ ms}$, arriva l'altro.

Si può direttamente fare: $(12 + 5 + 5) \cdot 6 = 132 \text{ ms}$; a $132 + 6,24 \text{ ms}$ parte l'ultima, 5ms e arriva, altri 5ms e finisce tutto: $148,24$ e l'ultimo ack arriva.

19) Uguali a prima, ma ora errore sulla ricezione del pacchetto 3 (0, 1, 2, 3). Che capita?

Riprendendo i dati di prima, avremo:

$T_{0 \rightarrow 2} = 3 \cdot (12 + 5 + 5) = 22 \cdot 3 = 66 \text{ ms}$; ora: 12ms e parte il 3, 5 o NON arriva; passano, invece dei 5, i 40ms di timeout;

dunque altri 12ms 5 di invio e 5 di ritorno $12 + 40 + 12 + 10 = 62 \text{ ms}$.

\rightarrow Fin qui, 140ms. Aggiungo 2 $\cdot (12 + 5 + 5) = 46 \text{ ms}$, e andiamo a 186;

Aggiungo $6,24 + 5 (+5) = 195,24$

20) $t_{SN1} = 24 \text{ ms}$ (metà del precedente \leftarrow metà della velocità)

$t_{SN2} = 12 + 4 \text{ ms}$ \leftarrow ; 5ms da S a N, 10 da N a D.

$t_{p1} = 12 \text{ ms}$; $t_{p2} = 6,2 \text{ ms}$

Dunque: per i pacchetti grossi, avremo $6 \cdot (24 + 5 + 12 + 10 + \underbrace{10 + 5}_{ack}) =$

$6 \cdot 66 = 396 \text{ ms}$; a ciò si aggiungano $12,4 + 5 + 6,2 + 10 \rightarrow 42,6 \text{ ms}$



Se D: stop & wait; N: store & forward
 Fib da 9500 byte, errore IV° pacchetto (0,1,2,3...)

1500 byte, timeout 40 ms.

→ 6 pacchetti da 1600 + 40 byte, L da 760 + 40 byte $\begin{cases} 1500 \times 6 \rightarrow 26 \text{ ms} \\ 780 \times 1 \rightarrow 12,48 \text{ ms} \end{cases}$

II° canale: 1500 → 12 ms;
 780 → 6,24 ms

→ ogni pacchetto grosso arriva dopo (24 + 5 + 12 + 10) ms = 51 ms;
 l'ack arriva dopo altri 15 ms, dunque 66 ms.

Dopo 24 ms parte il timeout, dunque appena prima che l'ack arrivi, il nod. ritrasmette. Intanto arriva l'ack, e dunque si spreca.

Dop 66 ms arriva l'ack, ma, dai 64, il Tx trasmette, dunque il pacchetto successivo è pronto e trasmesso a 88 ms: 88 ms è il tempo per la Tx di L pacchetto.

Pacch. 3: si ha un timeout supplementare, perché il pacchetto vien perso sul serio: $88 \times 3 + 64 + 88 \times 3$.

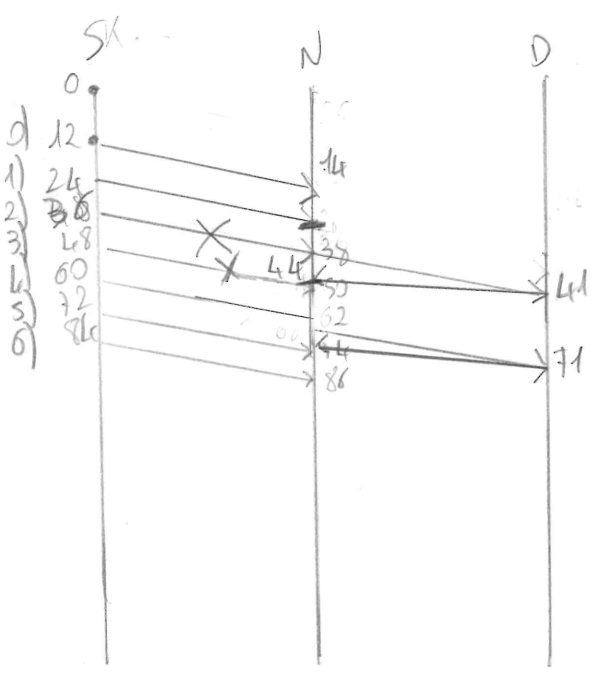
Ultimo pacchetto: ARRIVA E BASTA: $(24 + 5 + 6,24 + 10) \text{ ms} = 33,72 \text{ ms}$

22) $\frac{9500}{1500}$ = 6 pacchetti da 1500 byte, L da 500 byte.

Canale 1: $\begin{cases} 1500 \rightarrow 26 \text{ ms} \\ 500 \rightarrow 8 \text{ ms} \end{cases}$ canale 2: $\begin{cases} 1300 \rightarrow 12 \text{ ms} \\ 500 \rightarrow 4 \text{ ms} \end{cases}$

Da S a N non c'è alcun protocollo, da N a D uno stop & wait.

Vediam che capita: S manda in continuazione, N cerca di mandare appena riceve



In parole: parte D₁ a 14 arriva su M₁ a 14+24=38 è completamente fuori da N₁ a 44 vien cancellato; intanto arriva a 26 L₁ e a 38, 2; a 50 arriva 3, peccato che la memoria sia occupata dai 26 di L₁, dunque si perde parte di 3: si scarta tutto. 4 si piglia S no di nuovo, e 6 si.

Ragionando sugli spazi di memoria occupati. Si vede che N=5 è il min. numero di pacchetti da memorizzare.

23) $t_p = 20$ ms; 1500 byte, 40 di intestazione; Go-back-N, $W_T = 3$;

ACK trascurabile.

Al solito, $\frac{12000}{1665} \approx 8$; $12000 - 8 \cdot 1460 + 40 = 360$ byte. 12 ms per pacchetto grosso,

$2.88 \approx 3$ ms per quelli piccoli.

3 pacchetti per finestra $\rightarrow 0, 1, 2$ (mandati in $12 \cdot 3 = 36$ ms);

il primo pacchetto arriva a P dopo $12 \text{ ms} + 20 \text{ ms} = 32$ ms; P manda istantaneamente l'ack a S, dunque altri 20 ms; a 52 ms, la finestra di trasmissione va in avanti di L, dunque 12 ms e si manda il 4° \rightarrow altri 32 ms e se ne spediscono 3 e riceve indietro il ACK.

Al tal momento dovrà mandare altri $12 + 12 + 3$, e aggiunger altri 20 ms, ottenendo $27 + 20 = 47$ ms da aggiungere ai 104, ottenendo 151 ms.

28) Selective repeat. File da 15000 byte, 2 Mbit/s di V_{Tx} , $T_p = 15$ ms.

$W_T = 3$, $W_R = 2$.

\rightarrow 10 pacchetti giusti. Finestra Tx da 3, dunque $12 \times 3 = 36$ per finestra, e $12 + 15 + 15 = 42$ ms prima di poter mandar la successiva.

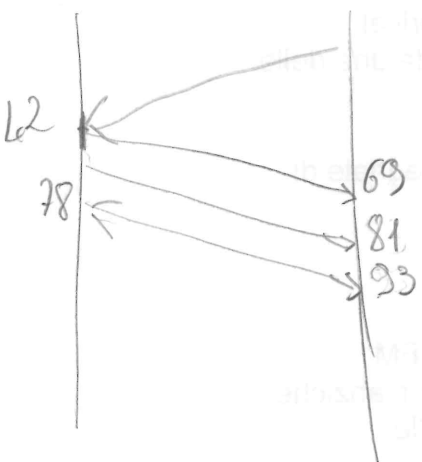
Senza errori, $42 \cdot 3 + 12 + 15 = 153$ ms

Con errori, capita ciò: abbiamo 10 pacchetti, da 0 a 9; 4 e 5 saltano, dunque quelli della II^a Finestra di Tx; il VI pacchetto si scarta perché la Finestra di Rx non riesce a tenerlo, e terminata la trasmissione del VI parte il timeout di 100 sec, dopo il quale si ritrasmette la II^a Finestra e così.

Dunque: $42 + 12 + 15 = 69$

Da 78 a 178 passa il timeout;

$178 + 42 \cdot 2 + 12 + 15 = 289$ ms



Ma se la finestra W_R fosse stata più grossa, tipo $= W_{Tx}$, potevamo tenerlo il VI^o pacchetto?

Nota: all'inizio si ha (0, 2); poi (3, 4, 5); 4 e 5 se scartati, dunque $ack(4) \rightarrow 84$ ms; $\rightarrow (4, 5) \rightarrow 184$ ($84 + 100$ ms considero timeout che parte da 0), dunque, essendo $W_R = 2$ e $W_T = 3$, non ci servono gli ack:

$\rightarrow 184 + 42 + 3 \cdot 12 + 15$

24) [5] → [0] 20 ms sul canale, 100 di timeout, 0, 1, ..., N, 4° perso
 40 header, 1500 byte totali. $W_t = 3$.

8 pacchetti da 1500, 1 da 360.
 Co-back-N è più rapido: si conta 1 volta solo il T_p.

T = 12, parte il 1°, a 32 arriva il 1°, a 32+12=44 il 2°, a 44+12=56 ms il 2°
 Poi si aspetta che la finestra si muova:

T_{0.2} = 20 + 4 × 12 = 68 ms: ora, arriva il 3. Intanto, a 32+20=52 ms,
 arriva l'ack del 1°, di "0", che muove avanti la finestra; a 52+12
 arriva quello di 1, a 52+24 quello di 2.

A 52+12 ms parte 3, e arriva dopo 20 ms: 64+20=84. Da qui,
 l'ack di 3 arriva a 104, facendo spostare la W_t su 4. Per t=64+12, 64+24,
 partono 4 (perduto) e 5 (scartato); ack di 3 fa spostare la finestra così
 che ora ci siano 4, 5, 6 dentro, e 6 possa partire per t=104+12=116 ms
 (104: momento ack 3); a 116 ms il timeout scatta, e a t=216 si
 ritrasmette l'intera finestra da 4 a 6: 216+12=228 + 20=248;
 a 248+12=260 arriva 5 su D, a 260+12 arriva 6 su D. Il ricevitore ack di 6
 arriva a 248+20=268. A 268+12+20+12+ $\frac{360 \cdot 8}{100}$ ms 302, 88 ms
 finisce la TX. **PRESTARE MOLTA ATTENZIONE.**

25) Si fa una specie di generalizzazione: senza errori, ogni "ciclo di 3
 dati" dura 52 ms; in ciascun ciclo si trasportano 3 pacchetti da
 1500 byte → 4500 byte in 52 ms → $\frac{4500 \cdot 8}{0,052} = 692 \text{ kbit/s}$.

L'utente userà solo 1460 dei 1500 bit: $L_u = \frac{1460}{1500} \text{ ts} = 676 \text{ kbit/s}$

Al fine di aumentare "t_r" si deve aumentare la finestra, senza però
 farla TROPPO grossa; si vede che il MAX è 5;

Ora, nota una volta per tutte; ogni volta che un ack "buono"
 fa avanzare la finestra, solo l'elemento INCLUSO ADESSO, dopo l'ack,
 nella finestra. Questo grazie a un contatore che si ricorda cosa è già stato
 mandato. L'unica condizione per mandare TUTTA la finestra è la fine
 del timeout, e il timeout parte con la trasmissione dell'ULTIMO elemento
 della finestra.

26) $\frac{50 \text{ kbyte}}{1500 \text{ byte} - 400 \text{ byte}}$; \downarrow ; 1 da 600.

a) Vediamo un ciclo: $\frac{1500 \times 8}{120} = 12 \text{ ms}$; $\frac{400 \times 8}{120} = 32 \text{ ms}$.

Ciclo normale:

$150 \text{ ms} + 12 \times 10 \text{ ms} = 270 \text{ ms}$; l'ack arriva a 312 ms.

Ci sono 3 cicli, dunque $3 \cdot 312 + 150 + \frac{1}{30} \times 12 + 1 \times 32 = 1137 \text{ ms}$

b) Perdo pacchetto 11, questo significa che, prima di $t = 2 \times 312 = 624 \text{ ms}$, si ha nella finestra i pacchetti 10-19; a (624 ms) parte il pacchetto 20, che va nella catena dopo l'ack di 10 ricevuto, dunque l'ultimo pacchetto mandato e scaricato è 20, a 12624 ms . 400 ms dopo riparte 11, dunque per $12624 + 400 + 12 + 150$ abbiamo 11 in D; (1198). A $+150 \text{ ms}$ 13-18, parte un nuovo giro, che farà caricare la finestra da 12 a 21; a $1348 + 12 + 150$ arriva 12 in D, dunque 13, 14, 15. $(1348 + 48 + 150) = 1546 \text{ ms}$. 15 manda il proprio ack a $1546 + 150 = 1696 \text{ ms}$. In questo istante, la finestra è carica da 16 a 25, dunque si trasmetterà, per $t = 1696 + 12$, il nuovo 25: A $1696 + 12 + 150$ finisce il timeout, e da qua si rimanda la finestra 16-25:

$2108 + 150 + 10 \times 12 = 2378 \text{ ms}$. A $2108 + 12 + 150 + 170$, dunque 2420, cioè la finestra 26-35, che finisce. **NOTA**: versione easy: se sbaglia 500 1 volta, ci sarà 500 timeout la 1ª volta!

c) Facciamolo giusto: primo 10, 0-9, ho 312 ms al 1º ack; entrambi ^{soglia} le finestre a 10-19. Ora: $312 + 12 + 150$ arriva 10, (474), ~~(486)~~, che a 636 ^{errore} di nuovo l'ack di trasmissione di 10, ma intanto partono, tra 312 e 624 ms 10, 11, 12, 13, 14, 15, 16, 17, 18, 19?

TIMEOUT FOR MECCA
È UN CASO.

29) Abbiamo 4 LAN, dunque:

Le interfacce sono: le porte dei router, e gli host H.

- 1: 10.0.1.0/24
- 2: 10.0.2.0/24
- 3: 10.0.3.0/24
- 4: 10.0.4.0/24

- R₁N: 10.0.4.1/24
- R₂N: 10.0.4.2/24
- R₁S: 10.0.2.1/24
- R₂S: 10.0.3.2/24
- R₃W: 10.0.2.3/24
- R₃E: 10.0.3.3/24

- H₄: 10.0.4.10/24
- H₃: 10.0.3.10/24
- H₂: 10.0.2.10/24
- H₁: 10.0.1.10/24

ora: la tabella di instr. per gli H: semplicemente rimanda a un router piuttosto che a un altro; l'host non ha scelto dunque solo per H₁ e H₆ vedremo.

* significa "direttamente collegato".

Host	Rete dest.	Genmask	GW	Router	Rete dest	Genmask	GW	
H ₄	10.0.4.0 default	24 10	* 10.0.4.2	R ₁	10.0.4.0 10.0.2.0 10.0.1.0 default	24 24 24 10	* * 10.0.1.3 10.0.4.2	
H ₃	10.0.3.0 default	24 10	* 10.0.3.3		R ₂	10.0.4.0 10.0.3.0 10.0.2.0 default	24 24 24 10	* * 10.0.4.1 10.0.3.3
H ₂	10.0.2.0 default	24 10	* 10.0.2.1			R ₃	10.0.1.0 10.0.2.0 10.0.3.0 default	24 24 24 10
H ₁	10.0.1.0 default	R ₁ 10	* 10.0.1.3					

Domande:

- 1) Cio' non ho specificato sulle reti, quindi avrei potuto usare i 192.168?
- 2) Io ho designato come gateway per le reti e antenna default. E' ok?

30) Unica nota (per il resto ok): con gli indirizzi /25 e /26 si puo' fare:
 /25: da 0 (rete) a 127 (broadcast); /26: da 128 (rete) a 191 (broadcast)
 /26.2: da 192 (rete) a 255 (broadcast).

Attenzione: verifica es 29:

Cio' che deve capitare, per non rischiare cicli, e non usare MAI due gateway diversi per la stessa rete a costo di fare il giro dell'oca.

32) $\left\{ \begin{array}{l} 1 - 10.0.1.0 \\ 2 - 10.0.2.0 \\ 3 - 10.1.0.0 \end{array} \right.$

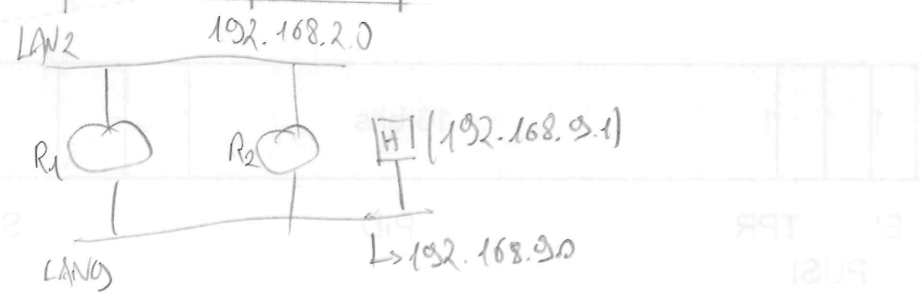
R1N - 10.0.2.252/24 H1 - 10.0.1.1/24
 R1S - 10.0.1.251/24 H2 - 10.0.2.1/24
 R3N - 10.0.2.253/24 R3E - 10.1.0.253/16
 R2S - 10.0.1.252/24 R2E - 10.1.0.252/16
 R4W - 10.1.0.254/16 R4E - 130.149.10.15/24

Facce direttamente la III:

host	dest. net	mask	gw	int.
H1	10.0.1.0	124	*	S
	default	10	10.0.1.252	S
H2	10.0.2.0	124	*	N
	default	10	10.0.2.253	N
R1	10.0.1.0	124	*	S
	10.0.2.0	124	*	N
	default	10	10.0.1.252	S
R2	10.0.1.0	124	*	S
	10.1.0.0	16	*	E
	10.0.2.0	124	10.0.3.253	E
	default	10	10.1.0.254	E
R3	10.0.2.0	124	*	N
	10.1.0.0	16	*	E
	default	10	10.0.2.251	N

Da R1 vedo a R2

34) Parallelo:



Una nota: per creare cicli nell'instadamento è sufficiente "concatenare" un tot di router indicandoli con il DEFAULT GATEWAY; quando si cerca, in una rete come questa di mandare FUORI da ciascuna sottorete (in modo che nessun nodo possa ritornare alla consegna diretta), il pacchetto continuerà a saltare da un nodo al default gate way e così via, per il TTL.

Si EVITA di far cicli usando, per ciascuna rete, SEMPRE UN SOLO GATEWAY. Se si fa così si allungano i percorsi ma si evitano cicli.

27-10-09 Reti

Livello 4: qui si fa multiplexare: tante applicazioni diverse, quindi il liv. 4 attacca tanti s.p.s a liv. 3; corregge errori. UDP: veloce; TCP: affidabile.

Affidabilità: TCP: trasportare informazioni con garanzia di trasmettere ~~correttamente~~.

TCP: interfaccia tipo stream: sequenza di byte non numerati. Ciò causa qualche problema alle applicazioni.

Controllo di congestione: throughput della rete relacionado a input, ingressi.

Se ci son troppi ingressi, la rete si blocca, ossia non spedisce più dati.

Si gestisce la finestra di trasmissione 0 per controllare il flusso, o per limitare gli ingressi nella rete e controllare la congestione.

Ricorda: indirizzo TCP: IP + PORTA

→ Controlli RTT, per FINESTRA

il ricevitore SPINA.

Ricorda: TCP lavora a byte

Tempo prima di wrap-around: tempo da aspettare prima di ricircolare.

Esercizi Reti Telematiche - Slides di esercizi

A.1 : capacità R bits, file F = M * L bit L = numero bit nel pacchetto, M numero pacchetti

a) $t_c + \alpha \cdot \frac{M \cdot L}{R}$; c) $\frac{(H+2L)(L+h)}{R}$

b) $Q \cdot \frac{(2h+L)(H+2L)}{R}$; d) $5 \cdot \frac{[(2 \cdot 20 \cdot 1000) \times 8] \cdot [(8+3) \cdot 1]}{1 \text{ Mbits}} = t_{cv} + \frac{[(20 \cdot 1000) \cdot 8] \cdot (8+3) \cdot 1}{1 \text{ Mbits}}$

Faccendo la differenza dei due, $t_{cv} = 1,6 \text{ ms}$

a) Non moltiplico per Q perché T_p è 0, dunque avremo $\frac{ML}{R}$, poiché la propagazione è istantanea: quando parte un bit sul primo nodo è come se partisse su tutti, il secondo avrà il pacchetto;

b) C'è PIPELINE; questo significa che quando si esce dal primo nodo il secondo avrà il pacchetto; il primo (i.e. 25ms) o vid. Il problema è che ciascun nodo può trasmettere solo dopo aver finito il precedente, e $T_{prop} = 0$; si ha per ogni stadio un $\frac{L+2h}{R}$ aggiuntivo, dovuto all'attesa del termine della trasmissione. All'ultimo nodo avremo $(Q-1)$ volte sto ritardo (ADDITIVO)

B.1
 $\frac{500}{1460} = 0,34$; $9500 - 6 \times 1460 = 740$; 6 pacchetti da 1500 B ; 5 ms di T_p
 1 da 780 B

$T_{1500B} = \frac{1500 \times 8}{2 \times 6} = 12 \text{ ms}$; $T_{780B} = \frac{780 \times 8}{100} = 6,24 \text{ ms}$

$\rightarrow T_{tot} = 6 \times (12 \text{ ms} + 5 \text{ ms} + 5 \text{ ms}) + 6,24 \text{ ms} + 5 \text{ ms} = 163,2 \text{ ms}$

B.2
 Uguale, con errore sul 10° pacchetto; timeout di 40 ms che inizia a pacchetto trasmesso.

Molto "selvaggiamente": da 0 a 2 tutto regolare, poi 12 ms, 40 ms, e poi di nuovo regolare; basta AGGIUNGERE 52 ms;

$\rightarrow 105,2 \text{ ms}$

B.3
 Sostanzialmente, ora si han 2 tempi per tipo di pacchetto:

$T_{Sx,1000} = 24 \text{ ms}$; $T_{Xp,1000} = 12 \text{ ms}$
 $T_{Sx,500} = 12,48 \text{ ms}$; $T_{Xp,500} = 6,24 \text{ ms}$

$\rightarrow T_{L \text{ pacchetto, } 1500} = (24 + 5 + 12 + 10 + 10 + 5) \text{ ms}$ (pacchetto + ack);

$T_{tot} = 6 \cdot T_{L \text{ pacchetto } 1500} + (12,48 + 5 + 6,24 \cdot 10) \text{ ms} = 429,72 \text{ ms}$ (lui approssima [male] allo I° cifra)

B.4

$$\frac{12000}{1460} = 8,2 \rightarrow 12000 - 8 \times 1460 = 320$$

8 pacchetti da 1500 ms $\rightarrow T_{rx1} = \frac{1500 \times 8}{106} = 12 \text{ ms}$ (2)

2 pacchetto da 360 ms $\rightarrow T_{rx2} = \frac{360 \times 8}{106} = 2,88 \text{ ms}$

$T_p = 20 \text{ ms}$.

Cosa capita? Dopo $12 + 20 + 20 \text{ ms}$ arriva indietro l'ack del 1° pacchetto, e dopo $12 + 12 + 12 + 20 \text{ ms}$ son arrivati 3 pacchetti su D. Diciamo che da S ogni 36 ms (12×3) si finisce di trasmettere, ma solo a S2 si può ricominciare.

\rightarrow Ci vanno $[(2 \times 52) + (12 + 12 + 2,88 + 20)] \text{ ms} = 150,9 \text{ ms}$. NON CI SERVE L'ACK DEGLI ULTIMI!

Come si può migliorare? Beh, si deve avere trasmissione INTERRUPTA, dunque quando il 1° ack arriva, lui non deve esser fermo. Ciò si può fare se $T_{rx} > 52 \text{ ms}$.

Con 4 pacchetti si han 48 ms, con 5 ∞ : $Wt \geq 51$

B.5
La trasmissione avviene in cicli da 60 ms, nei quali si trasmettono 3 blocchi da 1500 B. Dunque:

rete: $\frac{3 \cdot 1500 \cdot 8}{52 \text{ ms}} = 692,3 \text{ kbit/s}$ Se $Wt = 51$, i cicli diventano da 60 ms (non ci son "cicli", dal momento che la Tx è ininterrotta):

utente: $\frac{3 \cdot 1460 \cdot 8}{52 \text{ ms}} = 673,8 \text{ kbit/s}$

utente: $\frac{5 \cdot 1460 \cdot 8}{60 \text{ ms}} = 973 \text{ Hbit/s}$

B.6
File da 100 KiB da S a D, $T_p = 150 \text{ ms}$, 1500 B (140 B), ACK cumulativi

a) $Wt = 10, WR = 1$

\rightarrow 70 pacchetti da 1500 B, 1 da 140 B.

$T_{1500B} = \frac{1500 \times 8}{106} = 12 \text{ ms}$

$T_{140B} = \frac{140 \times 8}{106} = 1,092 \text{ ms}$

\rightarrow Il 1° ack torna dopo $(12 + 150 + 150) \text{ ms} = 312 \text{ ms}$.

Il sistema finisce di trasmettere la finestra a $12 \times 10 = 120 \text{ ms}$.
Avremo 7 cicli da 312 ms, e $1,092 + 150$ in più: 2,336 s.

b) Primi 312 ms, tutto tranquillo; poi, capita che parte il 10°, l'11°, e dunque dopo $(312 + 12 + 150 + 150) \text{ ms} = 624 \text{ ms}$ arriva l'ack dell'11°, che fa spostar la finestra dal 12° al 21°: altri $12 + 140 \text{ ms}$ e parte il timeout: $(624 + 12 + 140) \text{ ms} = 776 \text{ s}$.
Ora riparte tutta la finestra. Avremo dunque 4 cicli normali + il ciclo "Anno":

$1,036 \text{ s} + 4 \times 312 \text{ ms} + (12 \times 9 + 1,092 + 150) \text{ ms} = 2,856 \text{ s}$.

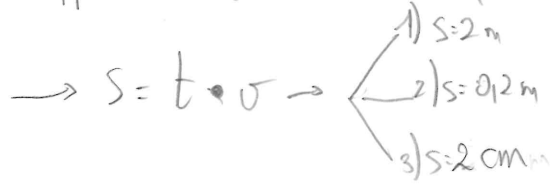
c) Cambia niente dal momento che l'ack(17) [relativo alla corretta ricezione della PDU 16] deve essere giunto su S prima che venga ritrasmessa 16; si deve avvertire l'assenza di 17 appena dopo che 12 parte.

RIVOLTA!

B.7 No store & forward significa che è inutile dividere in pacchetti:

$$\begin{aligned}
 t_{Tx1} &= \frac{100 \times 8}{1000000} = 0,0008 \text{ s} \\
 t_{Tx2} &= \frac{11}{1000000} = 0,000011 \text{ s} \\
 t_{Tx3} &= \frac{11}{1000000} = 0,000011 \text{ s}
 \end{aligned}
 \left. \begin{array}{l} \\ \\ \end{array} \right\} \begin{array}{l} \text{A questo si} \longrightarrow 0,0008 \text{ s} \\ \text{aggiunge il} \longrightarrow 0,000011 \text{ s} \\ \text{tempo di propagazione:} \longrightarrow 0,000011 \text{ s} \end{array}$$

Sappiamo che mediamente, $v = \frac{S}{E}$; il tempo di Tx di 1 bit è $\frac{1}{1000000} = \frac{1}{10^6}$



C.1
 $T_{PAB} = 3 \mu\text{s}$; $v_{Tx} = 10 \text{ Mbit/s}$; A: 64 B $\rightarrow T_A = 5,12 \mu\text{s}$
 B: 400 B (da 2 μs) $\rightarrow T_B = 320 \mu\text{s}$
 Il 1° bit di A arriva a B dopo 3 μs , il 1° bit di B arriva ad A dopo 5 μs .
 Il brutto è che è CSMA NON /CD, dunque la collisione c'è ma nessuno ci fa niente! C sente il chan occupato tra 1,5 μs e $320 + 2 + 1,5 = 323,5 \mu\text{s}$.

C.2
 $T_{PAB} = 10 \mu\text{s}$; CSMA/CD; $\left\{ \begin{array}{l} \text{A: 64 B} \rightarrow 5,12 \mu\text{s} \\ \text{B: 400 B} \rightarrow 320 \mu\text{s (parte da 5 } \mu\text{s)} \end{array} \right\}$ 2 μs per rilevare collisione.
 Sì, la inizia: dopo $t_0 + 10 \mu\text{s}$, arriva roba in B, e B la rileva e smette di trasmettere a $t_0 + 12 \mu\text{s}$. A, riceve dopo 15 μs , e a 17 smette di trasmettere.
 C sente il chan occupato da 5 μs a $17 + 5 = 22 \mu\text{s}$.

C.3
 Ora, $v_{Tx} = 100 \text{ Mb/s}$; questo significa che: $\left\{ \begin{array}{l} \text{A: 64 B} \rightarrow 5,12 \mu\text{s} \\ \text{B: 400 B} \rightarrow 32 \mu\text{s (parte da 5 } \mu\text{s)} \end{array} \right\}$ 2 μs per rilevare
 Prima parte, uguale; a $t = 10 \mu\text{s}$ si rileva la collisione in B, ed essa perdura fino a 15,12 μs ; B la rileva a $(10 + 2) \mu\text{s} = 12 \mu\text{s}$, quindi B la rileva.
 Tempo che B inizia a trasmettere, 5 μs , il 1° bit ad A arriva dopo 10 μs , dunque a 15 μs ; A non trasmette, dunque non sente NULLA.
 C sente il chan occupato da 5 μs a $12 + 5 = 17 \mu\text{s}$.

C.4
 $T_h = \eta \cdot \frac{C}{N}$ $N = 4 \text{ per } H_1$ $N = 3 \text{ per } H_2$
 $C = 10 \text{ Mb/s}$

$S_x = 0,6 \cdot \frac{10}{4} = 1,5 \text{ Mb/s}$
 $\rightarrow 1,5 \times 4 = 6 \text{ Mb/s}$

$D_x = 0,6 \cdot \frac{10}{3} = 2 \text{ Mb/s}$
 $\rightarrow 2 \text{ Mb/s} \times 3 = 6 \text{ Mb/s}$

$\rightarrow T_h = 12 \text{ Mb/s}$

D.1

(4)

$$L_a = 10 \text{ Mb/s}; L_b = 2 \text{ Mb/s.}$$

Quando si invia da ICMP, si passa per IP, dunque si aggiungono 20 B di intestazione IP; i pacchetti sono da 60 B.

$$T_a = \frac{60 \times 8}{1000} = 48 \mu\text{s}; T_b = 240 \mu\text{s.}$$

Come procede il transfer? Beh, parte il I° pacchetto, ed esso arriva su R₁ a 48 μs; parte, e dopo 240 μs arriva a D (240 + 48) μs = 288 μs.

Intanto, su R₁ è arrivato il II° pacchetto a 96 μs (48 + 48); esso attende fino a 288 μs, poi parte, e arriva a 528 μs.

Si noti che $\frac{60 \times 8}{240 \text{ ns}} = 2 \text{ Mb/s}$: è la capacità del canale che fa da collo di bottiglia.