

Classificazione delle reti: **a)** reti a commutazione di pacchetto (pacchetti vengono trasmessi da una sorgente ad una destinazione→condivisione della capacità trasmissiva della rete); **b)** commutaz. di circuito in cui è stabilito un collegamento fisico tra una qualunque coppia di utenti→rete telefonica (voce + dati). Altra classificazione: **c)** canali punto-punto→molte connessioni tra coppie di nodi (rete telefonica in cui centrali telefoniche sono interconnesse attraverso collegamenti punto-punto). **d)** canali broadcast→per realizzare reti a commutaz. di pacchetto in ambito locale e metropolitano (es un bus cui è connesso un certo num di workstation; un pacchetto inviato da una staz è visto da tutte le altre e la dest è tale perché descritta in un campo dest nel pacchetto). **Classificazione basata sulle dimensioni:** scheda→0.1m; sistema→1m; (locale→10m; edificio→100m; campus→1km LAN); città→10km (MAN); nazione, continente→WAN; pianeta→Internet. **Modello della rete:** topologia a stella, a bus, anello, reti parzialmente e totalmente connesse.

Rete a commutaz. di circuito: esiste un collegamento fisico tra chiamante e chiamato che permane fino alla terminazione della chiamata. La tecnologia è analogica; la tratta vincolante è quella che separa l'utente dal più vicino centro di rete urbana→sistemi progettati apposta per la voce, ma può essere estesa anche al traffico dati attraverso il modem (trasforma il segnale da analogico a digitale).

Reti a commutaz. di pacchetto: reti di calcolatori in cui pacchetti vengono trasmessi individualmente sulla rete. 2 modalità: a) circuito virtuale: è stabilito un percorso end to end tra sorgente e dest; b) Datagram: non viene stabilito alcun percorso a priori; i pacchetti possono seguire strade diverse per arrivare alla destinazione→le operaz di instradamento si basano sull'indirizzo di dest del datagram; se connectionless→pacchetti non in ordine.

Architetture di comunicazione a strati: Il problema della comprensibilità della comunicazione è centrale, nel caso di trasmissione dati; per affrontare questo problema (si veda comunicazione tra PC e una macchina UNIX, in cui i comandi e i caratteri di controllo devono essere specificati con precisione e devono essere definiti protocolli ad hoc per lo scambio di informazioni) in termini di trasporto tempestivo e consegna in formato riconoscibile→architetture a strati.

Modello ISO-OSI: modello astratto che prevede 7 strati diversi: i primi 3 livelli garantiscono la capacità di networking, i quattro più alti presentano i dati all'utente in un formato riconoscibile. Principi di scelta: a) un livello è aggiunto ogni volta che serve un diverso grado di astrazione; b) ciascun livello deve svolgere una precisa funzione; c) obiettivo di ciascun livello è la definizione di protocolli standardizzati; d) I confini tra i livelli devono essere scelti in modo da minimizzare lo scambio di info tra i livelli; e) num livelli abb alto da garantire che ciascun livello non abbia troppe funzioni, quindi sia troppo complesso.

1) Livello fisico: ha il compito di trasmettere i bit sul canale di comunicazione. Problematiche di progetto connesse al mezzo trasmissivo, su cui il livello fisico si appoggia (es: collegamento tra un modem e un PC→definizione di un'interfaccia std, descritta da un protocollo di livello fisico: RS232; es2: rete locale: tutti i calcolatori che si affacciano su tale rete devono adeguarsi a scelte di supporto trasmissivo, velocità, come trasmettere, se fare uso di codifica, livelli di tensione per 1 e 0.

2) Livello di linea: ha il compito di trasformare il mezzo trasmissivo in un collegamento percepito libero da errori trasmissivi dal livello di rete. I dati in ingresso vengono spezzati in pacchetti (frame), trasmessi in sequenza. Aggiunge poi un header e un tail che consentono di individuare inizio e fine del frame. Sincronizza ricevitore e trasmettitore, riordina i pacchetti in sequenza, rileva e corregge errori trasmissivi, rallenta la stazione sorgente, se trasmette troppo velocemente. Esistono 2 protocolli distinti per questo livello: a) HDLC per reti geografiche distribuite, realizzate mediante collegamenti punto-punto, che garantisce una comunicazione affidabile su ogni nodo; b) protocolli che descrivono l'accesso al mezzo trasmissivo per reti LAN (es per tipologia a bus, uno std prevede la trasmissione mediante token).

3) Livello di rete: compito di trasferire pacchetti da S a D. I problemi di questo strato sono quelli di instradamento (routing) risolti con tabelle di routing. Altri compiti: interconnessione di reti diverse, gestione di situazioni di congestione (congestion control)

4) Livello di trasporto: ha il compito di fornire a livello superiore un servizio efficiente di trasporto dati, da S a D. Ne esistono di 2 tipi: a) connection-oriented → fase di apertura, trasferimento, chiusura della connessione; b) connectionless: non orientato alla connessione. L'esistenza del livello di trasporto garantisce che i servizi resi da tale livello a quelli superiori siano più affidabili di quelli offerti dal livello di rete (rete gestita dal gestore)

5) Livello di sessione: fornisce livelli a valore aggiunto rispetto a quelli del livello di trasporto. Ha il compito di condurre il controllo del dialogo. Una sessione può consentire lo scambio di info full-duplex o in un'unica direzione alla volta, half-duplex. Ulteriore servizio di sincronizzazione.

6) Livello di presentazione: consente al livello applicativo di eseguire le primitive del servizio fornite dal livello di sessione; converte strutture dati, comprime i dati trasmessi, garantisce la privacy e l'integrità dei dati, mediante crittografia.

7) Livello applicativo: Qui troviamo applicativi veri e propri (es file transfer, http, ecc)

Per quanto riguarda l'invio dei dati, ogni livello, dall'alto verso il basso incapsula i dati, aggiungendo un nuovo header, che a lato ricezione verrà tolto man mano che si sale. Il dialogo tuttavia avviene tra coppie di livelli (peer to peer).

Modello di riferimento: TCP/IP: basata su uno strato di rete di tipo connectionless. **L'internet layer** definisce un formato di pacchetto a cui è associato un corrispondente protocollo **IP**. Compito dell'IP è l'instradamento dei pacchetti. A livello superiore troviamo il **livello di trasporto**, che comprende 2 protocolli: a) **TCP**, affidabile, connection-oriented, che frammenta lo stream di byte in ingresso in messaggi consecutivi e li passa all'internet layer; a D, è compito del TCP riassemblare il messaggio nello stream di Output. TCP fa inoltre controllo di flusso; b) **UDP**: protocollo connectionless non affidabile, orientato alle applicazioni che non richiedono il controllo di flusso (applicazioni real time). Al di sopra c'è il **livello di applicazione**: contiene i protocolli di alto livello (telnet, http, ftp).

RS232E (livello fisico): standard dovuto all'EIA, in cui il calcolatore è chiamato DTE, l'apparecchio che consente la comunicazione → DCE (ad es modem). Lo standard regola la comunicazione tra DTE e DCE.

Specifiche meccaniche: connettore a 25 pin, largo 47.04 mm. La fila superiore contiene i primi 13 pin, da sx a dx, poi sotto gli altri. In generale il connettore femmina si trova sul DCE.

Specifiche elettriche: 1 binario = -3V; 0 binario = 4V; livelli compresi tra -3 e 4 non sono riconosciuti come validi; velocità di max 20Kbit/s e cavi non più lunghi di 15m. Specifiche funzionali: ci dicono quali circuiti sono connessi a quali pin, + significato. Esistono 9 pin sempre implementati: GND → pin1 → massa di protezione: collegato alla massa dell'apparecchio per ridurre il rumore; SG (signal ground) → pin7 → riferimento di tensione per tutti i segnali → trasmissione sbilanciata.

Funzionamento: il calcolatore acceso invia Data terminal ready, settando on sul pin 20 → modem acceso invia Data set ready sul pin 6 → quando individua la portante sulla linea invia Carrier Detect sul pin 8 → Calcolatore per inviare dati manda Request to Send su pin 4 → Clear to Send del modem su pin 5 → Dati trasmessi su pin 2 (transmit circuit) e pin 3 (receive circuit). Specifiche procedurali: protocollo basato su coppie azione-reazione. Si possono connettere anche 2 calcolatori con RS232, utilizzando un dispositivo di NULL modem. Il nuovo standard **RS-499** individua l'interfaccia sotto i profili dell'RS232, tranne quella elettrica, divisa in 2 std: **RS-432-A** simile all'RS-232 → trasmissione sbilanciata; e **RS-422-A** che utilizza la trasmissione bilanciata: ciascuno dei circuiti principali richiede 2 fili → Velocità a 2 Mbit/s, connettore a 37 pin, lunghezza cavi: 60 m.

STD USB (Universal Serial Bus): Qualunque tipo di dispositivo può essere collegato ad una porta USB → semplicità d'uso → alto livello di complessità a livello HW e SW. Specifiche USB: definizione di requisiti meccanici per tutti i dispositivi; protocolli e meccanismi per configurare e trasferire dati.

Caratteristiche generali: due elementi che costituiscono il sistema sono: a) dispositivo USB, nel quale sta un microcontrollore e un codice per la comunicazione con l'host; b) l'host che supporta l'USB. Velocità trasmissiva: 12 Mbit/s (1.5 a low-speed mode).

Vantaggi: sostituzione di tutti i connettori con una combinazione di porta e spinotto; non necessario spegnere il PC per aggiungere una periferica; alimentazione USB di molte periferiche; connessione di + dispositivi alla stessa porta. Topologia: ciascun device si connette ad un hub → topologia tiered star.

Livello elettrico: impiega un cavo a 4 fili (GND, Vbus=5V, D+, D-) ; i segnali viaggiano su 2 fili su ciascun segmento punto-punto. Dispositivi connessi: sono divisi in classi, che presentano diverse richieste di I/O; ad ogni classe è associato un set di driver generici, std nativi del S.O → classi servono per raffinare meccanismi di base. Esistono sempre dispositivi unici → i vendor differenziano i loro prodotti con la concorrenza per il driver → driver vendor-specific. Protocollo di comunicazione: utilizzo di polled bus → i trasferimenti sono iniziati dall'host controller che interroga periodicamente i dispositivi connessi alla tried star; una transazione inizia quando l'host controller invia un frame USB che descrive il tipo e il verso della transaz, l'indirizzo del device e il num di end-point (token packet). Il dispositivo USB si riconosce come destinatario interpretando il campo indirizzo. I dati sono trasferiti da host a device e viceversa. Per comunicare con il device, l'host apre delle pipes logiche, impiegate per trasportare i dati (end point di tipo HW). Data transfer: costituito da un certo num di frame scambiati tra host e device → 4 tipi: bulk, isocrono, interrupt, controllo. Frame data: Syn (1 byte), Pid (1 byte), Data, Crc (2 byte). Descrittori: i dispositivi USB usano strutture dati predefinite, i descrittori, per descrivere i dati significativi, capacità, richieste (vendor, configuration, interface, endpoint). Enumeration: processo con cui il S.O riconosce un nuovo dispositivo connesso al bus, determina i requisiti e attribuisce al device 1 indirizzo univoco. Evoluzione: USB 1.1, High speed USB (transfer rate 480 Mbit/s, connessione fino a 127 dispositivi, compatibilità con le precedenti versioni).

Modem dial-up: dispositivo che consente la trasmissione di un segnale numerico su una linea telefonica. Compito del modem è modulare una portante con il segnale numerico da trasmettere, in modo da generare un segnale modulato la cui banda sia compresa tra 300 e 3300 Hz, poiché la freq di taglio < 300Hz non consente la trasmissione in continua e vel max è limitata superiormente → Teorema di Nyquist: in assenza di rumore, la max freq di simbolo (Baud rate) su un canale di banda H è: $B = 2H$; vincolo imposto dall'interferenza intersimbolo. Anche in assenza di interferenza intersimbolo il Teorema di Shannon dice: anche quando è garantito un rapporto segnale-rumore pari a S/N, max bitrate è: $R = H \log_2(1 + S/N)$ → per garantire un buon utilizzo del canale i modem devono equalizzare il canale → Equalizzazione adattativa: all'inizio del collegamento il modem testa il canale. Per raggiungere l'annullamento dell'interferenza inter-simbolo i modem operanti a vel trasmissive di 4800 bit/s impiegano equalizzatori fissi; quelli con vel maggiori, testano il canale e compensano le eventuali non linearità. Modulazione: prevista con una specifica costellazione: >efficienza di modulazione → > bitrate (ad es 16 posizioni → baud rate di 2400simboli/s, bitrate = 9600 bit/s). Std v34 → costellazione di 768 punti → bitrate di 28.8 kbit/s. Modalità di trasmissione: half-duplex → a tempi alterni, impiegando l'intera banda; full duplex → comunicazione contemporanea in 2 sensi → 2 sottobande (oggi esistono modem full-duplex occupanti tutta la banda del canale, grazie alla cancellazione dell'eco. Molti modem implementano la *rilevazione e correzione d'errore*, *compressione dati* → > velocità. STD ITU per i Modem: La international telecommunication union ha approvato gli std: a) **V.90**: std per modem a 56 Kbit/s; adotta il metodo di compressione lempel-ziv; b) **V.92**: - usa V.44 come metodo di compressione; - quick connect (riduce il tempo di negoziazione → dial-in + rapido → si esegue la memorizzazione delle sequenze di trainig nella fase di equalizzazione adattativa; - modem on hold: stessa linea telefonica per le chiamate e i dati → possibilità di telefonare anche con una connessione modem aperta; - migliorato upstream. V44 offre un rapporto di compressione rispetto a V42 (implementato in molti modem). **PROCEDURE ARQ**: siamo a livello di linea → obiettivo: garantire una comunicazione affidabile su un link ptop; sul link vengono trasmessi frame=unità logiche strutturate, viaggianti in entrambe le direz. La comunicazione è asincrona → i pacchetti devono trasportare info di sincronizzazione. I pacchetti con errori devono essere riconosciuti errati → procedura di ack per indicare se i pacchetti sono stati ricevuti correttamente o meno; numerazione dei pacchetti per garantire una consegna nell'ordine corretto allo strato superiore; aggiunta di campi di controllo (num seq. Frame), payload e un campo di controllo d'errore (CRC). Lunghezza del frame = 1 bit (payload) + 1' bit (header + coda). Modalità di invio degli ack: a) esplicita: all'arrivo di un frame corretto faccio partire un

frame speciale → ack esplicito, senza payload, inviato in modo cumulativo; b) piggybacking: ricevuto un frame corretto, invio l'ack in un frame dati che lo contenga → ci sarà un campo speciale nell'header in grado di contenere l'ack; in tal caso è cmq necessario prevedere frame ack espliciti, da utilizzare nel caso un nodo non abbia più dati da mandare; c) uso di Nack.

In ogni caso è necessario prevedere l'uso di un timer per prevenire situazioni di deadlock → se non si riceve una risposta entro un TO il frame viene ritrasmesso (i frame trasmessi stanno in un buffer). Esistono 3 procedure per gestire la risposta agli ack/nak: **1) Stop and Wait:** si può trasmettere solo un frame alla volta; il trasmettitore si ferma ed attende un ack/nak; se arriva un nack o il To scade, il frame viene ritrasmesso. Protocollo appropriato se la comunicaz è half-duplex, in cui i nodi alternano la trasmissione → evidenti problemi di throughput in caso full duplex, se $t_{prop} \gg t_{frame}$. **2) Go-back-n:** i frame dati sono trasmessi in continuazione, senza attendere un ack/nak per ciascuno di essi. Se nak o TO senza ack/nak, il frame in questione e quelli seguenti vengono ritrasmessi. **3) Selective repeat:** viene ritrasmesso solo il frame per il quale dal nodo ricevente è arrivato un nak, oppure se scaduto il TO → miglioramento delle prestazioni. Necessario in ricezione un buffer per il riordino dei frame (molto grandi in ambito satellitare, per questo Selective repeat non ha avuto molto successo).

Protocollo STOP-AND-WAIT: supponiamo che l'intervallo di time-out sia uguale al sup valore minimo: $t_{out} = 2t_p + t_{proc} + t_s$, con $2t_p$ = tempo di propagazione, t_{proc} = tempo di processing, t_s = tempo necessario a trasmettere un ack; detto t_f = tempo di frame, si definisce $t_T = t_f + t_{out}$ il tempo che intercorre tra 2 frame successivi → freq max di invio = $1/t_T$ frame/s. Definendo le ipotesi: a) una stazione trasmette e l'altra riceve, mandando solo ack/nak; b) trasmettitore operi in saturazione → ha sempre un frame pronto da inviare; c) non c'è limite per il numero di trasmissioni consentite; d) i frame vengono colpiti da errore in maniera indipendente gli uni dagli altri; si può determinare il throughput nel seguente modo: definiamo $E[t_v] = t_T + E[n] * t_T$, tempo mediamente necessario per conseguire correttamente un frame; p = prob che un frame sia ricevuto errato alla stazione B; definiamo $Pr\{fare\ i\ ritrasmissioni\} = p^i(1-p) \rightarrow$ allora $E[n] = (1-p) * \sum_{(i=1, \infty)} i * p^i$, con n = numero di ritrasmissioni, var aleatoria di tipo discreto. Allora si ha che $E[n] = (1-p) * \sum_{(i=1, \infty)} i * p^i = (1-p) * p * \sum_{(i=1, \infty)} i * p^{i-1} = (1-p) * p * \sum_{(i=1, \infty)} (\partial/\partial p)(p^i) = (1-p) * p * (\partial/\partial p) \sum_{(i=1, \infty)} p^i$ ma ricordando che $\sum_{k=0}^{\infty} g^k = 1/(1-g)$ se $g < 1 \rightarrow$ è la serie geometrica, allora $E[n] = (1-p) * p * (\partial/\partial p) (\sum_{(i=1, \infty)} p^i) = (1-p) * p * (\partial/\partial p) (p/(1-p)) = (1-p) * p * ((1-p) + p)/(1-p)^2 = p/(1-p) \rightarrow$ così si ha che: $E[t_v] = t_T + E[n] * t_T = t_T (1 + p/(1-p)) = (1/(1-p)) * t_T$; introduciamo ora un parametro: $a = t_T/t_f = (t_f + t_{out})/t_f = 1 + (t_{out}/t_f) > 1 \rightarrow E[t_v] = (1/(1-p)) * a * t_f$; consideriamo ora $1/E[t_v]$ = numero medio di frame consegnati correttamente in un secondo → Si definisce S = throughput, il num medio di frame consegnati correttamente in un tempo di frame: $S(\text{stop and wait}) = ((1-p)/a * t_f) * t_f = (1-p)/a$, con $a = t_T/t_f = (t_f + t_{out})/t_f = 1 + (t_{out}/t_f) = 1 + (2t_p + t_{proc} + t_s)/t_f$, e $t_f = (1+l')/l$; → tanto + la qualità è buona, tanto + la p sarà bassa. Lavorare con a alti → t_p è ben + grande del tempo di frame. Per i link satellitari $a \gg 1$. Caso limite: $a \rightarrow 1$: $S(\text{stop-and-wait}) = 1-p$.

Protocollo GO-BACK-N: la comunicazione è full duplex → è meglio trasmettere in continuazione piuttosto che ogni volta ricevere l'ack del frame corrispondente → miglioramento del throughput, soprattutto se propagazione è trascurabile rispetto al tempo di frame. Le ipotesi sono le stesse del protocollo stop-and-wait. Poiché nel protocollo Go-Back-N i frame sono trasmessi in continuazione il tempo minimo che intercorre tra 2 trasmissioni è t_f , tempo di frame, anziché t_T . Allora vale la relazione: $E[t_v] = t_f + E[n] * t_T = t_f (1 + E[n] * a)$; sapendo che $E[n] = p/(1-p)$ (dimostrazione in stop-and-wait), ottengo che: $E[t_v] = t_f + E[n] * t_T = t_f (1 + p * a / (1-p)) = t_f * (1 - p + a * p) / (1-p) = t_f * (1 + p(a-1)) / (1-p)$ (numero di frame al secondo); da qui si può ricavare S , numero di frame in un tempo di frame: $S(\text{go-back-n}) = (1/E[t_v]) * t_f = t_f * (1-p) / (1 + p(a-1)) * t_f = (1-p) / (1 + p(a-1))$, con p = prob che un frame sia ricevuto errato; se la rete è buona $p \ll 1$. Si può concludere che $S(\text{go-back-n}) > S(\text{stop and wait}) \rightarrow$ La differenza si fa tanto maggiore quanto + $a \gg 1$. Si noti che per $a \rightarrow 1$, $S(\text{go-back-n}) = S(\text{stop and wait}) = 1-p$.

Protocollo SELECTIVE-REPEAT: per questa versione bisogna tener conto del fatto che bisogna avere un buffer di dimensione infinita per il riordinamento dei frame. Qui si possono trasmettere i frame singolarmente, pertanto quelli ricevuti correttamente dopo un frame errato non devono essere ritrasmessi. Allora si procede al calcolo di $E[t_v] = t_i + E[n] * t_i$; con $E[n] = p / (1-p)$, si ha che $E[t_v] = t_i (1+p/(1-p)) = t_i * 1/(1-p) \rightarrow$ allora $S(\text{selective-repeat}) = (1/E[t_v]) * t_i = ((1-p)/(1-p+p)) * t_i = 1-p$, che è il max che potevamo ottenere, ricordando che $a = t_r/t_i = (t_i + t_{out})/t_i = 1 + (t_{out}/t_i) = 1 + (2t_p + t_{proc} + t_s)/t_i$, con t_{out} = al suo valore minimo ammesso: $2t_p + t_{proc} + t_s$. Se non fosse così il time-out scatterrebbe prima della ricezione del pacchetto.

Data throughput: numero medio di bit di dati trasferiti con successo ogni secondo $D = S * l$ (soli bit dati)/ t_i ; **Data throughput normalizzato:** numero medio di bit di dati trasferiti con successo ogni secondo normalizzato alla capacità trasmissiva del canale: $D/C = (S * l/t_i) * 1(\text{uno})/C$, sapendo che $t_i = (l+1')/C \rightarrow D/C = S * l/(l+1')$. **Probabilità p che un frame sia colpito da errore:** ipotizziamo che i diversi bit del frame siano colpiti da errore indipendentemente gli uni dagli altri. Un frame è considerato errato quando almeno uno dei suoi bit è errato; $1-p_b = 1 - \text{prob di errore per bit}$; $(1-p_b)^{l+1} = \text{prob che tutti i bit del frame siano ricevuti}$; $(1 - ((1-p_b)^{l+1})) = p = \text{prob che almeno un bit sia errato}$; se p_b è continua si può scrivere $\rightarrow p = (l+1') * p_b = \text{prob che un frame ricevuto sia errato}$. $(D/C) * \text{capacità del canale } C = \text{velocità effettiva che riesco a garantire allo strato di rete}$. Da notare che nel selective repeat e go back n, se ho il payload contenuto, il D/C cala, poiché ci metto più tempo a trasmettere info di controllo che dati; tuttavia per $l \gg 1$ è $>$ la prob di errore nel frame.

Protocollo HDLC: specifico protocollo di linea (di livello 2), che sfrutta il protocollo GO-BACK-N, sviluppato da ISO, progenitore di molti protocolli punto-punto. Caratteristiche: orientato al bit; impiega una specifica struttura di frame; protocollo connection oriented. Descrizione del formato: 48 bit (l') destinati al controllo; il frame si apre e si chiude con un flag (01111110), delimitatore di inizio e fine frame; la seq di sei 1 consecutivi non può apparire da nessun'altra parte del frame \rightarrow Impiego della tecnica di *Bit stuffing*: il trasmettitore dopo 5 uni consecutivi nella seq da inviare inserisce forzatamente uno 0 spurio, che verrà rimosso in fase di ricezione \rightarrow qualunque frame con 6 uni al di fuori del flag è errato. Descrizione del pacchetto: flag (1 byte), indirizzo (1 byte), controllo (1 byte), dati (1 byte), CRC (2 byte), flag (1 byte). Il campo address contiene l'indirizzo di dest; il 3° byte è di controllo, \neq a seconda dei diversi frame; CRC consente la rilevazione d'errore. HDLC prevede 3 tipi di frame: a) I (information); b) U (Unnumbered); c) S (supervisory); I frame U e S non contengono l'information field, poiché destinati alla supervisione e al controllo. Relativamente al campo controllo: frame I: [0, N(S), P/F, N(R)]; frame S: [1, 0, S, S, P/F, N(R)]; frame U: [1, 1, M, M, P/F, M, M, M], con N(R) e N(S) di tre bit. I primi bit consentono di discriminare il tipo di frame; N(S) individua il num di seq del frame (mod8); il valore caricato in N(R) consente di fornire l'ack (per convenzione fornisce l'ack sull'avvenuta ricezione del frame N(R)-1, e aspetta N(R)); i 2 bit di S (S, S) consentono 2^2 diversi tipi per tale frame; per i frame U se ne individuano 2^5 relativamente ai 5 bit M. Il trasmettitore conserva in un buffer tutti i frame per cui non ha ancora ricevuto ack. Quando questo accade, il frame è eliminato dal buffer. Modalità operative di HDLC: HDLC prevede diverse modalità operative: a) Normal response mode: adottata in un contesto centralizzato \rightarrow una staz primaria interroga periodicamente le staz secondarie, che iniziano una trasmissione solo in risposta ad un comando primario; b) Asynchronous response mode: qui la staz secondaria può iniziare la trasmissione senza il permesso di quella primaria; c) Asynchronous balanced mode (ABM): utilizzata per la comunicazione punto-punto, con stazioni aventi ruoli paritetici \rightarrow base per l'X.25. Nel frame S, l'ack è in forma esplicita, nel frame I, in modalità piggy-backing. Modalità ABM: RR \rightarrow 00, ready to receive \rightarrow dà l'ack a tutti i frame fino a N(R)-1; RNR \rightarrow 10, not ready to receive \rightarrow dà l'ack, ma indica una condizione di busy del ricevitore; REJ \rightarrow 01, reject \rightarrow rifiuta tutti i frame da N(R) in avanti (ack fino a N(R)-1). Da notare che anche il frame I contiene il campo N(R) \rightarrow ack in modalità piggy-backing. REJ può fornire un nak \rightarrow con REJ il protocollo utilizzato è il GO-BACK-N. Procedure Bac in ABM: in corrispondenza a ciascuna modalità operativa esiste una classe di procedure con funzioni e opzioni ben precise; per

la ABM si tratta della Balanced Asynchronous Class (BAC), che ha 2 opzioni importanti: a) opzione 2 che introduce l'uso di REJ; b) opzione 8, che impiega il bit Poll/Final per implementare il meccanismo di risposta ai comandi. Opzione 8: definisce alcune regole: 1) un frame di tipo I è un comando per l'opzione 8; 2) i frame di tipo RR, RNR possono essere sia comandi (indirizzo staz ricevente) che risposte (indirizzo staz sorgente); 3) i frame REJ sono sempre risposte e recano l'indirizzo della staz sorgente; 4) Un comando con P/F settato esige una risposta immediata con P/F settato. (comando \rightarrow P = 1; risposta \rightarrow F = 1) \rightarrow si realizza una fase di checkpointing, utile per forzare un ack immediato, trasmissione REJ in caso di errore, verificare un collegamento, preparare una fase di disconnessione). Numerazione: il num di seq può assumere valori nel range: $0 \leq N(S) \leq M-1$; con $M = 8$ in modalità normale e 128, se setesa \rightarrow num max di frame che il trasmettitore può inviare, prima di doversi bloccare perché non ha ancora ricevuto l'ack per i primi frame costituenti la sequenza è 7, $M-1$, perché l'8 sarebbe lo 0 che si confonde con il primo \rightarrow meccanismo della finestra scorrevole in HDLC.

Protocollo PPP (Point-to-Point Protocol): protocollo di livello di linea che opera esclusivamente su collegamenti di tipo punto-punto, orientato al carattere (frame costituiti da un num intero di byte); non fa correzione d'errore, solo rilevazione, no fa controllo di flusso, consegna i frame non necessariamente in sequenza. Caratteristiche: possibilità di impiego su link eterogenei (ottici, low speed, ecc); incapsulamento in frame dei pacchetti ricevuti dal livello di rete; rilevazione d'errore, verifica dell'operatività del link; semplicità e negoziazione delle opzioni circa il livello di rete. Formato del frame: [Flag, Address, Control, Protocol, Data, CRC, Flag]; fa uso del byte stuffing come in HDLC. I campi address e control possono avere solo un preciso valore; il valore nel campo Protocol indica al PPP ricevente a quale protocollo + alto appartengono i dati del frame; il campo Data contiene il payload; Checksum è impiegato per rilevare bit errati. LCP Link Control Protocol: si occupa della fase di inizializzazione con relativa negoziazione di parametri, manutenzione, anomalie e chiusura di un link PPP. Attivazione e disconnessione: DEAD (rilevazione portante) \rightarrow Link established (negoziiazione dei parametri di connessione, attraverso un pacchetto LCP la cui risposta è configure ack) \rightarrow Network layer configuration (attribuzione degli indirizzi IP ai 2 estremi) \rightarrow Open (scambio dati e pacchetti echo-request e echo-reply) \rightarrow Terminate (terminate ack). Descrizione: 1) via modem si stabilisce una connessione fisica col provider, 2) Il PC manda al router una serie di frame LCP nel campo dati di uno o + frame PPP, per negoziare una serie di parametri; 3) vengono inviati una serie di pacchetti NCP per configurare lo strato di rete; 4) quando l'utente termina la connessione con ISP, NCP termina la connessione a livello di rete, liberando l'indirizzo di rete; 5) LCP chiude la connessione a livello di linea; 6) il modem chiude la connessione rilasciando il livello fisico.

Tecniche di accesso ad un canale condiviso: a) strategie di tipo statico: FDMA, TDMA; b) Strategie di tipo casuale: ALOHA, CSMA; c) Strategie di accesso su richiesta: POLLING, TOKEN-BASED. **Strategie di tipo deterministico**: **TDMA**: l'asse dei tempi è idealmente suddiviso in frame; ogni trama è costituita da M slot temporali, uno per ogni stazione, tutti di identica durata. La stazione i-esima può trasmettere solo nello slot i-esimo \rightarrow eliminaz delle contese per l'accesso alla risorsa condivisa; ripartizione rigida della risorsa tra le M stazioni; ciascuna staz trasmette al più R/M bit/s, se $R = \text{capac canale}$ \rightarrow penalizzazione di alcune stazioni se burstiness è elevata. **FDMA**: si ripartisce la banda in freq anziché la risorsa tempo \rightarrow ciascuna sottobanda ha un'ampiezza pari a $B_i = B_r/M$; in generale l'approccio statico risulta adeguato per i servizi di fonìa e per quei servizi di cui è nota a priori la banda; ok per GSM, GPRS. **Strategie di accesso casuale**: M utenti condividono ancora una risorsa, attraverso la quale si scambiano frame dati, ma dinamicamente. **Pure Aloha**: protocollo proposto negli anni 70, caratterizzato da una topologia a stella \rightarrow host centrale e diverse stazioni; 2 link impiegati: UPLINK, DOWNLINK. Nell'Aloha, se una staz ha un frame da trasmettere, lo invia brutalmente, senza preoccuparsi dello stato del canale e delle altre stazioni \rightarrow possibili collisioni: frame sovrapposti contemporaneamente sul canale \rightarrow frame persi che andranno ritrasmessi; assenza di info condivisa; la staz che ha colliso, deve ritrasmettere

dopo un ritardo aleatorio. Prestazioni: Hp) le M stazioni generano un traffico descrivibile attraverso un processo di poisson di parametro λ frame/s (frame ex novo che ritrasmessi); frame a dimensione costante, pari a l bit; il canale è ideale \rightarrow perdite dovute solo a collisioni. Allora si può definire il traffico offerto al canale: $G = \lambda \cdot t_f = \text{num medio di frame offerti al canale in un tempo di frame}$. Si definisce peraltro il periodo di vulnerabilità t_v , periodo nel quale può avvenire una collisione: da $t_0 - t_f + \epsilon$, a $t_0 + t_f - \epsilon \rightarrow t_v = t_0 - t_f + \epsilon - (t_0 + t_f - \epsilon) = 2t_f + 2\epsilon$. Si può dunque definire il Throughput $S = \text{num medio di frame trasmessi con successo sul canale di frame in un tempo di frame}$: $S = G \cdot \Pr\{\text{no arrivi di frame sul canale nel periodo di vulnerabilità}\}$; poiché gli arrivi appartengono ad un processo poissoniano $\rightarrow P_k(t) = ((\lambda t)^k / k!) e^{-\lambda t} \rightarrow S = G \cdot P_0(2t_f) = G \cdot e^{-\lambda(2t_f)}$, me essendo $G = \lambda \cdot t_f \rightarrow \lambda = G/t_f \rightarrow S = G \cdot e^{-2G}$. **Slotted Aloha**: migliora le prestazioni dell'Aloha; fa uso di slot \rightarrow l'asse dei tempi è suddiviso in slot di dimensione esattamente pari a t_f ; le staz possono iniziare a trasmettere solo all'inizio di ciascuno slot e sono sincronizzate \rightarrow le collisioni possono essere solo totali. Periodo di vulnerabilità: in questo caso si riduce a $t_f \rightarrow t_v = t_f \rightarrow$ Si può descriverne il throughput: $S = \text{num medio di frame trasmessi con successo sul canale di frame in un tempo di frame}$: $S = G \cdot \Pr\{\text{no arrivi di frame sul canale nel periodo di vulnerabilità}\} \rightarrow S = G \cdot P_0(t_f)$; ricordando che $P_k(t) = ((\lambda t)^k / k!) e^{-\lambda t} \rightarrow S = G \cdot P_0(t_f) = G \cdot e^{-\lambda t_f} = G \cdot e^{-G}$. La percentuale di slot vuoti è data da $P_0(t)|_{G=1} = e^{-\lambda t}|_{G=1} = e^{-G}|_{G=1} = e^{-1} = 0.37$; percentuale di slot occupati da collisioni $= 1 - 0.37 - 0.37 = 0.26$. **CSMA(carrier sense multiple access)**: questa strategia sfrutta l'info dello stato del canale \rightarrow riduce la prob di collisioni ma non la elimina (2 staz che testano contemporaneamente il canale con un certo ritardo di prop. Colliderano). 1 persistent CSMA: la staz generica ha un frame da trasmettere; esegue il test del canale, se libero trasmette, se occupato continua ad interrogare il canale in modo persistente, fino a che non si libera. CSMA non persistente: la staz esegue il test su canale, se libero trasmette, altrimenti introduce un ritardo aleatorio prima di interrogare il canale di nuovo. CSMA non persistente slotted: dato un frame da trasmettere aspetta l'inizio dello slot successivo, poi esegue il test su canale; se libero trasmette, altrimenti introduce un ritardo e ritenta la trasmissione attendendo uno slot. CSMA p-persistent slotted: $0 \leq p \leq 1$ con p = parametro di persistenza del canale; la staz si allinea allo slot successivo \rightarrow test su canale; se occupato, riprova, altrimenti non trasmette con prob 1 ma la staz genera un num random $0 \leq r \leq 1$; se $r \leq p$ allora trasmette, altrimenti si riposiziona sullo slot successivo e ripete l'operazione. Collision detection: la staz che si accorge che il suo frame ha colliso, si ferma immediatamente. S (non persistent) $= 0.8$. Periodo di vulnerabilità di CSMA: sia τ_{ij} il ritardo di propag dalla stazione i alla j, dipendente dalla distanza tra le 2 staz d_{ij} . Se all'istanza t_0 la stazione i ha un frame da trasmettere, fa il sensing del canale; se a $t_1 < t_0 + \tau_{ij}$, anche la seconda staz vuole trasmetter, fa il sensing del canale e conclude erroneamente che il canale è libero \rightarrow collisione \rightarrow il periodo di vulnerabilità è il ritardo di propagazione τ_{ij} . **Determinazione del traffico-throughput per CSMA non-persistent**: Hp) processo degli arrivi dei frame poissoniano; ritardo di prop sia il max ritardo di prop. τ ; frame di lunghezza costante; canale ideale; sovrapposizione anche parziale di due frame sia una interferenza distruttiva; operazione di sensing in tempo nullo. Il frame 0 viene trasmesso a t_0 , occupa il canale per t_f secondi \rightarrow qualunque frame pronto può essere messo sul canale nel periodo $(t_0, t_0 + \tau)$, perché gli altri vedono il canale libero. Il frame n è l'ultimo ad arrivare, in un tempo $t_0 + Y < t_0 + \tau$. Se arrivano frame in tempi successivi questi non vengono trasmessi. All'istante $t_0 + Y + t_f$ termina la trasmissione dell'ultimo pacchetto, ma le staz si accorgono che il canale è idle solo dall'istante $t_0 + Y + t_f + \tau$. L'intervallo $(t_0, t_0 + Y + t_f + \tau)$ è detto periodo di busy, quello da $t_0 + Y + t_f + \tau$ all'istante della trasmissione di un nuovo frame è detto tempo di IDLE. In questo caso il periodo di busy è unsuccessful poiché durante questo periodo non è trasmesso alcun frame. Perché la trasmissione abbia successo, nel periodo τ nessun altro frame oltre lo 0 deve essere trasmesso \rightarrow per il non-persistent CSMA, il ritardo di propagazione è proprio τ . Le grandezze in gioco: U = tempo trascorso durante un ciclo a trasmettere con successo (assume valori 0 e t_f); B = durata del periodo di busy; I = durata del periodo di idle; I+U = ciclo. Si può determinare il throughput come segue: $S = E[U] / (E[B] + E[I])$. Determinazione di E[U]: $E[U] = 0 \cdot \Pr[\text{periodo di busy sia unsuccessful}] + t_f \cdot \Pr[\text{periodo di busy successful}] = t_f \cdot \Pr[\text{no collisioni}]$

nell'intervallo $(t_0, t_0 + \tau]$ $= t_1 \cdot P_0(\tau) = t_1 \cdot e^{-\lambda \tau}$, ma ricordando che $G = \lambda t_1 \rightarrow E[U] = t_1 \cdot e^{-G\tau/t_1}$;
Determinazione di E[B]: $E[B] = E[Y + t_1 + \tau] = E[Y] + t_1 + \tau$ (il valor medio di una costante è la costante stessa). Occorre determinare $E[Y] \rightarrow$ 1) valutare la funzione distribuzione $F_Y(y)$; 2) ottenere la funzione densità di probabilità; 3) calcolare $E[Y] = \int_0^\tau y \cdot f_Y(y) dy$; 1) Calcolo $F_Y(Y) = \Pr[Y \leq y] = \Pr$ [no arrivi sul canale nell'intervallo $(t_0 + y, t_0 + \tau]$] (altrimenti quello indicato con la Y non sarebbe l'ultimo) $= P_0(\tau - y) = e^{-\lambda(\tau - y)} = e^{-G(\tau - y)/t_1}$; 2) $f_Y(y) = (G/t_1) e^{-G(\tau - y)/t_1}$; \rightarrow 3) $E[Y] = \int_0^\tau y \cdot f_Y(y) dy = \int_0^\tau y (G/t_1) e^{-G(\tau - y)/t_1} dy = \tau \cdot (t_1/G) (1 - e^{-G\tau/t_1}) \rightarrow$ pertanto si può calcolare $E[B] = E[Y] + t_1 + \tau = \tau \cdot (t_1/G) (1 - e^{-G\tau/t_1}) + \tau + t_1 = 2\tau \cdot (t_1/G) (1 - e^{-G\tau/t_1}) + t_1$.
Determinazione di E[I]: $E[I] = 1/\lambda$ (coincide con il tempo che mediamente intercorre tra un arrivo di un frame sul canale e il successivo, valor medio di una var aleatoria exp a color medio λ) $\rightarrow E[I] = 1/\lambda = t_1/G$.
Determinazione di S: ora sostituendo otteniamo che $S = E[U] / (E[B] + E[I]) = (t_1 \cdot e^{-G\tau/t_1}) / (2\tau \cdot (t_1/G) \cdot (1 - e^{-G\tau/t_1}) + t_1 + t_1/G) \rightarrow$ introducendo il parametro $a = \tau/t_1$ e dividendo tutto per t_1 , $\rightarrow S = e^{-aG} / (2a \cdot (1/G) \cdot (1 - e^{-aG}) + 1 + 1/G) = G \cdot e^{-aG} / (2aG - 1 + e^{-aG} + G + 1) = G \cdot e^{-aG} / (G(2a + 1) + e^{-aG})$.
Osservazioni: Per $a = 0$, abbiamo il caso limite: $S = G/(G + 1)$; in questo caso particolare, se $G \rightarrow \infty$, allora $S \rightarrow 1$. Se $a = 10^{-2} \rightarrow S = 0.8$ (interessante rispetto ad ALOHA) Si ricorda che $a = \tau/t_1 = \tau/(1 + I)/R$, con R = capacità trasmissiva (bitrate). Vogliamo lavorare con reti con estensione geografica contenuta per avere τ contenuti (strategia di accesso per LAN, non per MAN o WAN). I frame convergono lunghi e velocità trasmissive non particolarmente elevate.
Confronto su capacità di canale: S_{max} per Aloha è indipendente da a , mentre per csma non persistent dipende molto da $a \rightarrow$ può scendere sotto S_{max} aloha; $>a \rightarrow >\tau$. Per Csma non persistent, ci si svincola dall'instabilità con 1 ritardo di back off.
Collision detection: Trovata 1 collisione il sistema interrompe bruscamente la trasmissione di frame collisi \rightarrow non spreco capacità trasmissiva \rightarrow miglio S. Proprietà desiderabili: 1) quando una sola stazione è attiva voglio che impieghi tutta la banda disponibile (vero per Aloha e Csma, falso per TDMA, FDMA); 2) quando M stazioni sono attive, voglio che ciascuna possa inviare R/M bit/s (vero per TDMA/FDMA, falso per Aloha e CSMA perché il canale è suddiviso dinamicamente, quindi ci sono collisioni).
Tecniche di accesso su richiesta: si basano su classi di accessi che prevedono turni: 1) accessi che fanno uso di Polling: una stazione designata come master interroga ciascuna stazione client in round robin; invia un msg ad una staz dicendo che può inviare un num max di frame; terminata la staz i, il master esegue il poll della staz i+1; si ripete \rightarrow no collisioni, ma c'è polling delay (il master interroga tutte le staz anche se 1 sola deve trasmettere, inevitabile). 2) Accessi token-passing: non c'è master, una sola staz alla volta è abilitata a trasmettere, quella che detiene un frame speciale detto token, per un intervallo di tempo e che poi deve passare; accesso decentralizzato, efficienza elevata, necessita di manutenzione: (token perduto?); std 802.5 Token ring.