

3. Modello di riferimento dei protocolli ISO/OSI

Concetti di protocolli, interfacce e primitive

Livelli di comunicazione

Livello fisico

Livello Data Link

Livello di rete

Livelli di elaborazione

Livello di trasporto

Livello di sessione

Livello di presentazione

Livello di applicazione

1



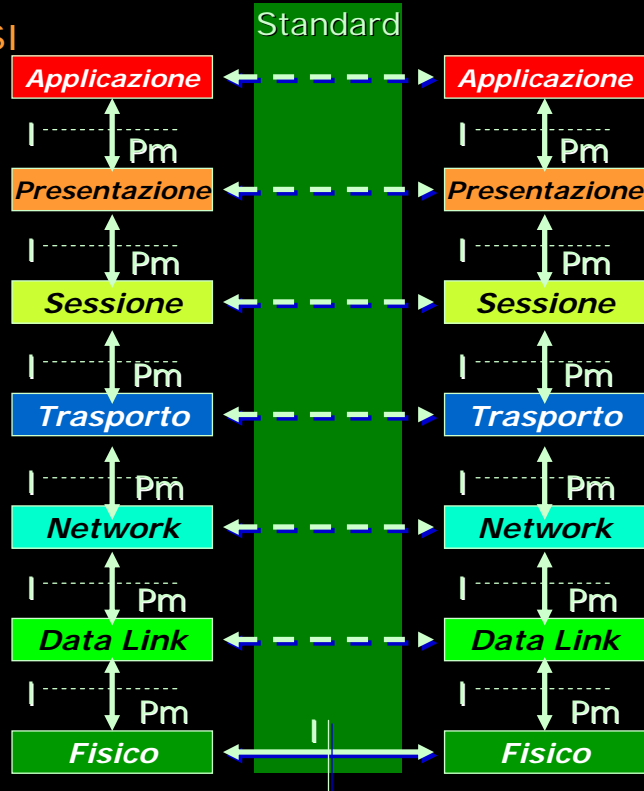
Concetti di protocolli, interfacce e primitive

Livello:
suddivisione dell'architettura OSI

Protocollo:
insieme di regole e formati che determina la comunicazione fra due livelli uguali

Interfaccia (I):
elemento di confine fra livelli adiacenti

Primitiva (Pm):
informazione scambiata tra 2 livelli adiacenti attraverso l'interfaccia



2



Il modello ISO/OSI è un modello di riferimento standardizzato nel 1988 dall'ISO (International Organization for Standardization) per lo sviluppo di standard di tipo aperto. E' chiamato modello di riferimento ISO-OSI (Open System Interconnection - interconnessione di sistemi aperti) perché si interessa di collegare sistemi aperti cioè, sistemi che sono aperti per comunicazioni con altri sistemi. Questo modello deve infatti essere inteso come riferimento per la realizzazione di protocolli standard, per metterli in relazione e per fornire una descrizione e una terminologia comune. Il modello OSI ha sette livelli. I principi che sono stati seguiti per arrivare a sette livelli sono i seguenti:

- Un livello dovrebbe essere creato ogni volta che viene richiesto un diverso livello di astrazione;
- Ogni livello dovrebbe realizzare una ben determinata funzione;
- La funzione di ciascun livello dovrebbe essere scelta con un occhio rivolto alla definizione di protocolli internazionali standardizzati;
- I limiti dei livelli dovrebbero essere scelti per minimizzare il flusso delle informazioni attraverso le interfacce;
- Il numero di livelli deve essere abbastanza ampio per permettere a funzioni distinte di non essere inserite forzatamente nel medesimo livello senza che sia necessario e abbastanza piccolo per permettere che le architetture non diventino pesanti e poco maneggevoli.

Si noti che il modello OSI da solo non è un'architettura di rete perché non specifica con precisione i servizi e i protocolli che devono essere usati in ogni livello. Esso dice solo che cosa dovrebbe fare ciascun livello. Tuttavia, ISO ha prodotto degli standard per tutti i livelli, anche se questi non fanno parte del modello di riferimento. Ognuno di questi è stato pubblicato come uno standard internazionale separato.

Il modello OSI suddivide le funzioni svolte dai vari livelli in 2 gruppi:

- 1) funzioni orientate alla rete, o di comunicazione;
- 2) funzioni orientate alle applicazioni o di elaborazione.

Ciascun gruppo funzionale è composto da un certo numero di protocolli: ciascun protocollo realizza funzioni tra loro complementari ai fini della comunicazione.

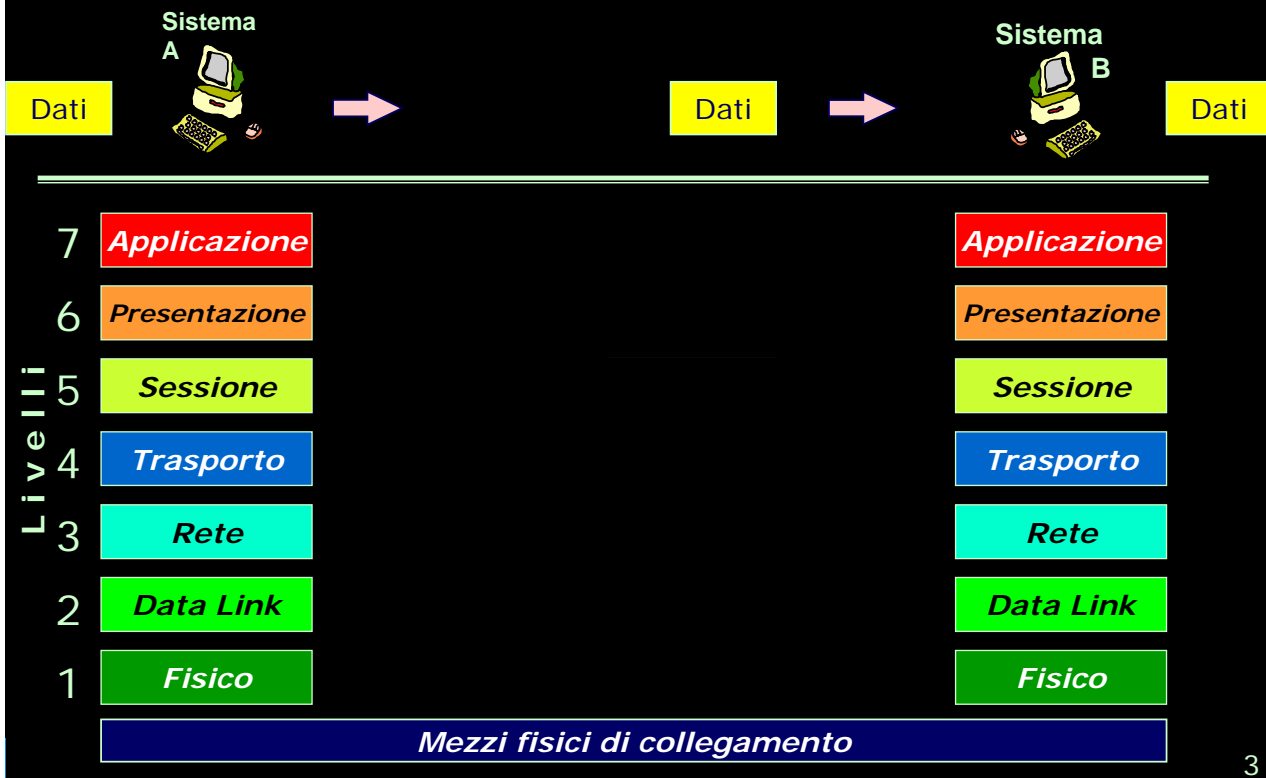
I protocolli sono costituiti da un insieme di regole che permettono il colloquio fra unità funzionali dello stesso livello (da 1 a 7). I protocolli necessitano di scambiare informazioni con i livelli adiacenti, facendo opportune richieste ai livelli inferiori per soddisfare le esigenze dei livelli superiori. Tali informazioni vanno sotto il nome di primitive, e il confine fra i livelli è l'interfaccia. Siccome tale situazione avviene interamente all'interno di uno stesso elaboratore, le primitive non sono soggette ad una modalità standard, ma vengono opportunamente implementate sulle specifiche esigenze del costruttore.

Due livelli, appartenenti a sistemi elaborativi diversi, comunicano tra di loro mediante l'intermediazione dei livelli inferiori. Ogni livello deve svolgere dei compiti ben precisi vincolato dagli standard internazionali di riferimento.

Solo le interfacce di livello 1 sono realizzate fisicamente, quelle altre si possono considerare "astratte" o virtuali.

Scambio delle informazioni tra livelli: overhead

Standard ISO/IEC 7498-1, Racc. ITU-T X.200



3



Un protocollo come già affermato è un insieme di regole che governa la comunicazione tra due entità di pari livello. Per scambiarsi le informazioni sulla modalità di processamento delle unità informative necessitano di byte di overhead.

La figura mostra un esempio di come possono essere trasmessi i dati usando il modello OSI. Il processo mittente vuole spedire alcuni dati a un processo destinatario. Esso passa i dati al livello applicazione, che attacca l'intestazione AOH (Application Overhead, che potrebbe anche essere nulla), all'inizio dei dati stessi e che passa il blocco di dati ottenuto al livello presentazione. Il livello presentazione può trasformare questo blocco di dati in diversi modi e forse aggiunge un'intestazione all'inizio, passando poi il risultato al livello sessione. E' importante rendersi conto che il livello presentazione non è a conoscenza di quale porzione di dati costituisca AOH, se questa è presente, o quali siano i veri dati dell'utente.

Supponiamo che il processo attivo giunto al livello 5 dai livelli superiori (6, 7) generi un messaggio da spedire all'interlocutore di pari livello. Il messaggio in questione verrà passato dal livello 5 al livello 4. Il livello 4 aggiunge un header al messaggio passatogli dal livello 5. L'header contiene in genere informazioni di controllo che servono a gestire il protocollo. Fatto ciò il livello 4 passa il messaggio risultante al livello 3, il quale come il livello precedente gli aggiunge un header. Infine a differenza degli altri, il livello 2 aggiunge al messaggio passatogli sia un header che un trailer e passa il tutto al livello 1 che fisicamente si preoccupa di spedire il messaggio. Questo processo viene cioè ripetuto fino a quando i dati raggiungono il livello fisico, dove sono finalmente trasmessi alla macchina destinazione. A questo punto il messaggio arriverà al livello 1 dell'host di destinazione, il quale passerà quanto ricevuto al livello superiore che rimuoverà l'header ed il trailer e passerà la rimanente porzione al livello superiore, e così via. In sostanza all'host ricevente ogni livello si occuperà di eliminare l'header che il livello corrispondente aveva introdotto e passare quindi il risultato al livello superiore. Cioè su tale macchina le diverse intestazioni sono eliminate una per una mentre il messaggio si propaga lungo i vari livelli fino ad arrivare al processo destinatario.

In questo modo i livelli hanno effettivamente la sensazione che la comunicazione stia avvenendo in orizzontale, come mostrato dalle linee tratteggiate nella slide precedente. L'idea chiave è che sebbene i dati vengano in realtà trasmessi verticalmente, ogni livello è programmato come se la comunicazione avvenisse in modo orizzontale. Quando il livello trasporto mittente, per esempio, riceve un messaggio dal livello sessione, esso vi attacca la propria intestazione e passa il tutto al relativo livello trasporto destinatario. Dal suo punto di vista, il fatto che debba passare il messaggio al livello rete della propria macchina è un aspetto tecnico trascurabile. Analogamente, quando un diplomatico parla alle Nazioni Unite nella propria lingua, egli pensa di parlare agli altri diplomatici. Il fatto che stia parlando solamente al proprio interprete è solo un dettaglio tecnico.

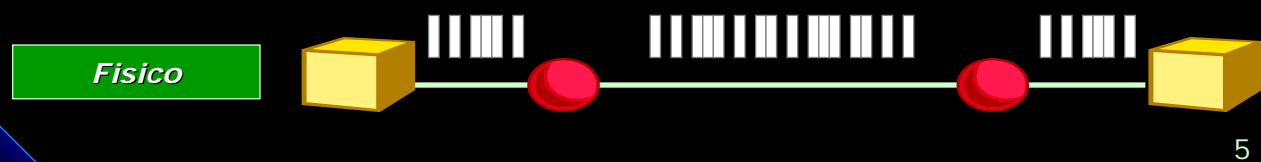
Livelli di comunicazione



Funzioni del livello fisico

- ↖ Trasmissione dei bit sui portanti trasmissivi
- ↖ Sincronizzazione degli apparati di trasmissione
- ↖ Multiplazione di tributari su flussi numerici a capacità maggiore
- ↖ Gestione dei sistemi trasmissivi
- ↖ Deve fornire ai livelli superiori un flusso di bit di "buona" qualità

Nelle reti pubbliche la valutazione della Qualità Trasmissiva viene riportata nelle Raccomandazioni ITU-T G.821, G.826, M.2100



5



Il livello fisico fa riferimento alla trasmissione dei bit lungo un canale di comunicazione. Lo scopo primario di tale livello è la definizione delle interfacce meccaniche ed elettriche, in base al mezzo trasmissivo che sta sotto di esso. Ad esempio bisogna specificare che tipo di connettore usare, il significato di ogni pin del connettore, che forma d'onda è assegnata al bit "1" e quale al bit "0" ecc.

Gli aspetti di progettazione hanno a che fare con l'assicurarsi che quando una estremità invia un bit 1 esso venga ricevuto dall'altra come un 1, e non come uno 0. Problemi tipici a questo livello sono quanti volt devono essere utilizzati per rappresentare un 1 e quanti per uno 0, quanti microsecondi richiede un bit, se la trasmissione può procedere simultaneamente in entrambe le direzioni, come si stabilisce la connessione iniziale e come la si termina quando a entrambe le estremità la comunicazione è finita, quanti connettori devono essere presenti sul cavo di rete e come ogni connessione deve essere utilizzata. Gli aspetti di progettazione hanno spesso a che fare con interfacce meccaniche, elettriche e procedurali, e il mezzo di trasmissione fisica, che si trova al di sotto del livello fisico.

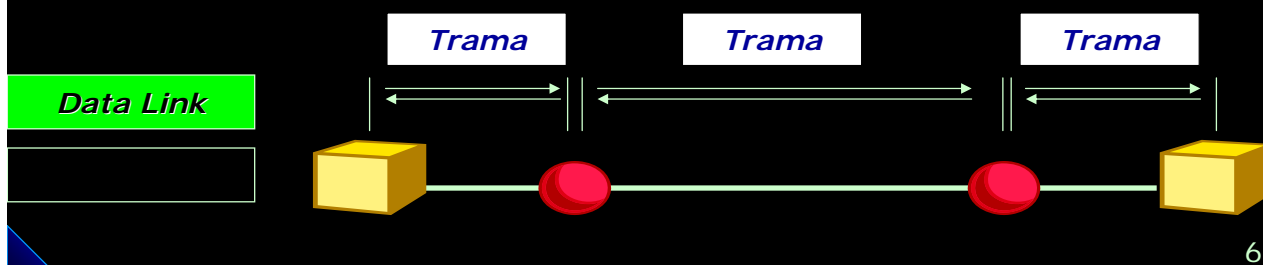
Al di sopra di tale livello, si avrà quindi a disposizione un servizio per il trasferimento di bit.

Funzioni del livello Data Link

- Definizione del formato delle trame
- Sincronizzazione delle trame
- Trasparenza dei dati
- Rilevamento dell'errore

Opzionali (connection Oriented):

- Attivazione e chiusura della connessione
- Gestione del trasferimento
- Controllo della sequenza
- Controllo del flusso
- Recupero dell'errore



Tale livello trasforma una trasmissione grezza in una linea in modo che appaia libera da errori di trasmissione non segnalati al livello superiore. Il compito principale del livello di Data Link (DLL - Data Link Layer) è quindi quello di mettere a disposizione un canale di comunicazione affidabile e privo di errori. Cioè questo livello, sfruttando quanto fornito dal livello fisico (e cioè la possibilità di trasferire bit su un mezzo trasmissivo) realizza, tramite il protocollo di Data Link, una comunicazione affidabile, servizio messo a disposizione al livello superiore.

Il livello Data Link riesce a realizzare una comunicazione affidabile suddividendo i dati provenienti in strutture dette trame o Data Frame (la cui lunghezza può variare dalle centinaia alle migliaia di byte) inoltrandoli sequenzialmente, attendendo poi il messaggio di avvenuta ricezione (ack) del pacchetto inviato dal ricevente. La correttezza dei dati viene verificata mediante algoritmi di checksum dal terminale ricevente il quale comunica l'esito al terminale trasmittente. Il Data Link layer deve anche prevedere meccanismi di ritrasmissione dei frame persi o corrotti. Visto che il livello fisico accetta e trasmette sequenze di bit senza far riferimento al loro significato o alla loro struttura, è compito del livello data link creare e riconoscere i limiti delle trame. Questo può essere realizzato attaccando dei bit speciali all'inizio e alla fine del "pacchetto". Se questi bit si presentano in modo accidentale nei dati, bisogna fare particolare attenzione per evitare che vengano interpretati erroneamente come delimitatori di pacchetto. Un'interferenza sulla linea potrebbe completamente distruggere un pacchetto. In questo caso, il software del livello data link sulla macchina dei mittente può ritrasmettere il pacchetto stesso. Tuttavia, trasmissioni multiple del medesimo pacchetto introducono la possibilità di pacchetti duplicati. Un pacchetto duplicato potrebbe essere spedito se il messaggio di avvenuto ricevimento è andato perso. E' compito di questo livello risolvere i problemi causati dal danneggiamento, perdita o duplicazione di pacchetti. Il livello data link può offrire diverse classi di servizi al livello superiore, ognuno di diversa qualità e avente un diverso costo. Un altro aspetto importante al livello data link (e anche in molti altri livelli) è come mantenere veloce una trasmissione evitando di sommergere un ricevitore di dati lento. Deve essere stabilito qualche meccanismo di regolazione del traffico per permettere al trasmettitore di conoscere quanto spazio ha a disposizione il ricevente per memorizzare i propri messaggi. Spesso, la regolazione di questo flusso e la gestione degli errori sono integrati.

Un'altra importante funzione svolta da questo livello è quindi il flow control per impedire ad un trasmettitore molto veloce di congestionare un ricevente più lento che, esauriti i buffer in cui memorizzare i dati in ingresso, sarebbe costretto a scartare delle frame ricevute correttamente.

Se la linea può essere utilizzata per trasmettere dati in entrambe le direzioni, questo introduce una nuova complicazione che il software del livello data link deve gestire. Il problema è che il messaggio di avvenuto ricevimento dei pacchetti che viaggiano da A a B compete per l'utilizzo della linea con i pacchetti che viaggiano da B ad A. E' stata proposta una soluzione efficiente (piggybacking) verrà presentata in dettaglio nel seguito.

Inoltre le reti broadcast hanno un'ulteriore particolarità per quanto riguarda il livello data-link: occorre controllare l'accesso al canale condiviso. Uno speciale sottolivello per il livello data link, detto MAC Medium Access Control (sottolivello di accesso al mezzo di comunicazione), ha il compito di gestire questo tipo di problema.

Sincronizzazione dei protocolli di linea

Protocollo ASINCRONO:

Non sono previsti legami temporali tra comandi e risposte



Protocollo SINCRONO:

Le trame sono tra loro legati temporalmente



7



I protocolli asincroni non prevedono alcun legame temporale tra le trame. Ai comandi, quindi, non è previsto nessun time out legato ai responsi. Questa tecnica fu inizialmente adottata per fare fronte alle carenze dei primi terminali e successivamente abbandonata grazie alla capacità di calcolo sempre maggiore degli Host.

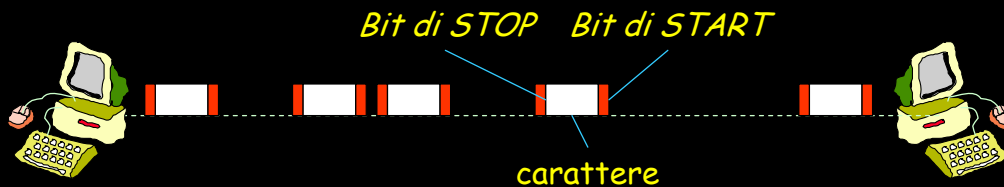
I protocolli sincroni sono invece caratterizzati da una relazione temporale fra comandi e responsi. Oltre un certo time out l'assenza di risposta viene considerata come un errore.

Sincronizzazione dei terminali

Metodo ASINCRONO:

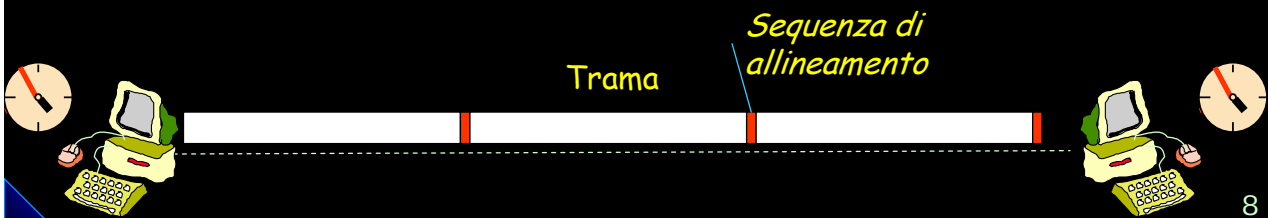
inserisce dei bit di start e di stop fra un blocco di bit di dati.

La stazione ricevente utilizza il bit di start per adeguare la propria velocità a quella della stazione trasmittente.



Metodo SINCRONO:

mantiene in sincronismo le due stazioni terminali con una piccola ridondanza inserita nei blocchi di bit informativi.



8



Storicamente la trasmissione asincrona deriva dalla necessità di collegamento delle telescriventi. I caratteri provengono dalla tastiera a intervalli casuali, dipendenti dalla pressione delle dita dell'operatore sui tasti.

In generale, considerando il fatto che ad una rete possono essere collegati diversi dispositivi è facile comprendere come questi lavorino a velocità differenti. I computer stessi hanno velocità di clock che possono variare di sistema in sistema. Quindi è necessario che le stazioni si concordino su una determinata velocità per evitare perdite di dati. I metodi sono due: asincrono e sincrono.

Nel metodo ASINCRONO, prima di ogni blocco di informazioni vengono inseriti dei bit di partenza o avvio e al termine bit di arresto. Nel momento in cui alla stazione ricevente giunge un bit di partenza questa provvederà ad adeguare la propria velocità a quella della stazione trasmittente.

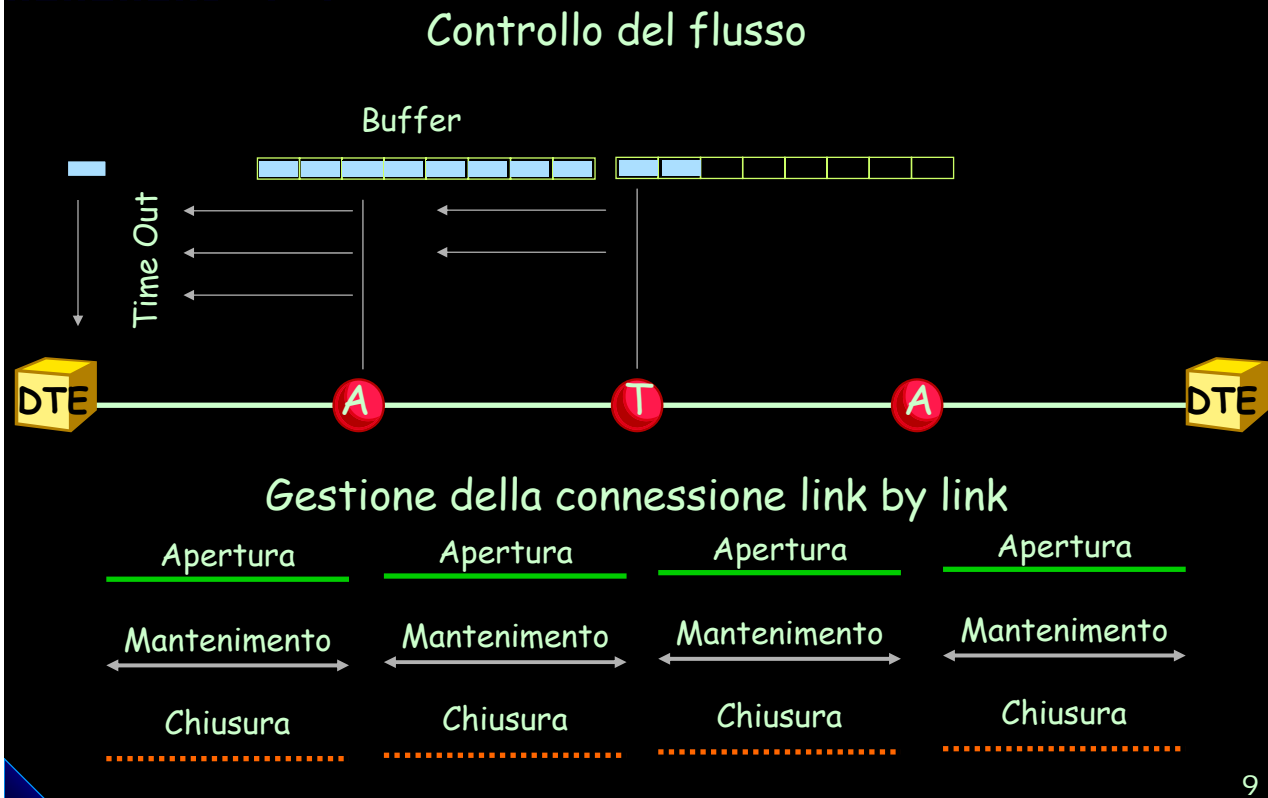
Questo metodo ha il grande vantaggio che in reti complesse (private) possono essere utilizzati apparecchi di diverso tipo che si concorderanno sulla velocità più adeguata ad ogni collegamento. Questo adeguamento continuo prima di ogni blocco comporta però un rallentamento nel trasferimento in quanto ai dati vengono aggiunte molte informazioni di protocollo che aumentano il volume dei dati e del tempo necessario al completamento dell'operazione di trasmissione.

Il metodo SINCRONO invece implica una sincronizzazione globale delle apparecchiature ad una determinata velocità che viene governata da byte aggiuntivi di sincronismo contenuti nelle trame che includono anche i dati. Il metodo sincrono implica quindi una sincronizzazione globale degli apparecchi collegati ad una determinata velocità rendendo omogeneo il trasferimento tra due stazioni.

Per essere precisi, l'entità ricevente per interpretare correttamente il messaggio speditogli deve essere in grado di determinare:

- L'inizio di ogni bit (bit or clock synchronization).
- L'inizio e la fine di ogni carattere (character or byte synchronization).
- L'inizio e la fine della trama (block or frame synchronization).

Controllo del flusso e gestione del collegamento



L'overhead introdotto a livello data link contiene dei campi che forniscono l'indirizzo del destinatario, oltre ad una serie di informazioni di controllo utilizzate per il controllo di sequenza e di flusso.

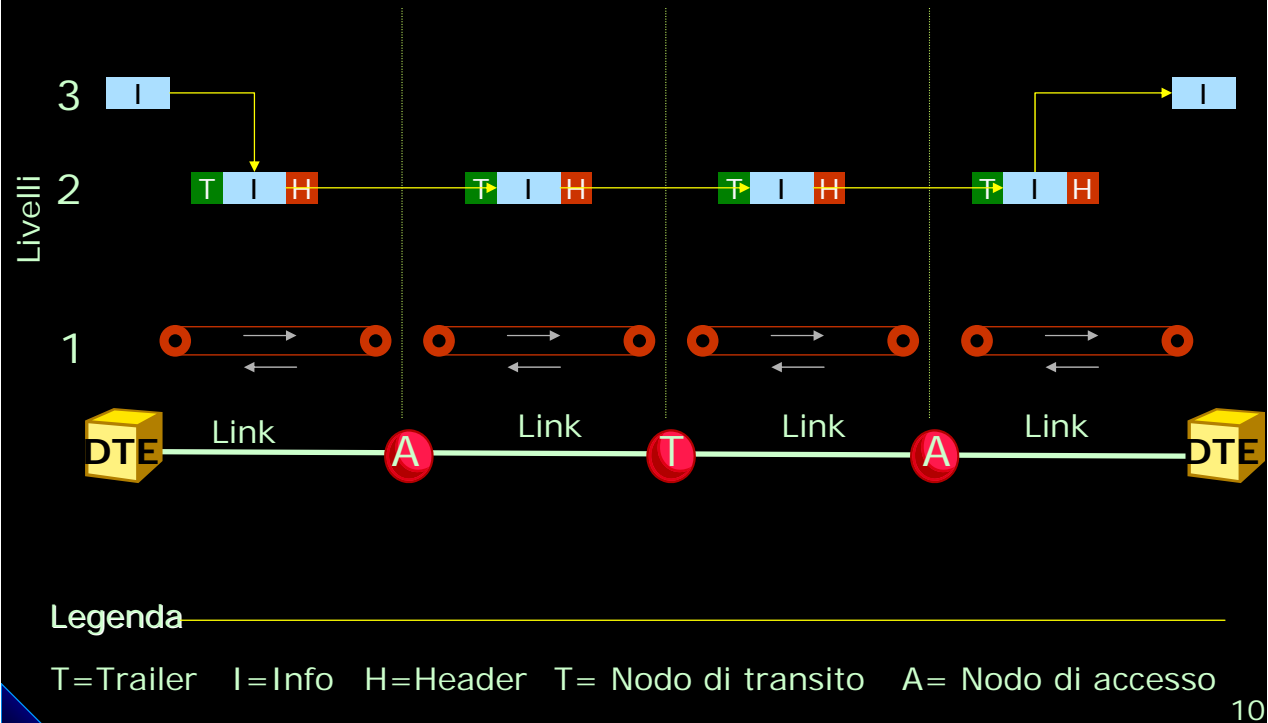
Il controllo di flusso (Flow Control) consiste nell'impedire che un nodo o una stazione trasmetta ad un ritmo medio più alto rispetto a quello che l'altro nodo adiacente (ricevitore) può, o vuole, accettare. L'eccessiva velocità dei dati ricevuti potrebbe determinare l'esaurimento della capacità di bufferizzazione con conseguente perdita delle informazioni.

La tecnica tipicamente usata per realizzare questo tipo di controllo è nota con il nome di Sliding Window (Finestra Scorrevole). Essenzialmente questa tecnica consiste nel porre un limite massimo al numero di Frame che possono essere spedite, senza ricevere conferma. Questo schema ha una azione auto-regolante, infatti nel caso in cui il ricevitore non invia le trame di conferma entro un tempo max (time out) significa che non riesce a mantenere il passo con il trasmettitore. Di conseguenza il trasmettitore provvede a diminuire la velocità.

Sin ora abbiamo visto come sia possibile scambiare frame tra due entità connesse da un link. Affinché questo scambio possa avvenire, le entità che desiderano colloquiare devono poter instaurare una connessione, ed analogamente, quando le due entità hanno concluso il loro colloquio devono poter chiudere tale connessione. Quindi uno dei requisiti dei protocolli di linea è quello di essere capaci di guidare il sistema da una fase all'altra seguendo una serie di procedure dipendenti dal particolare protocollo.

Il livello di rete

Trasferimento affidabile dei dati sui singoli link



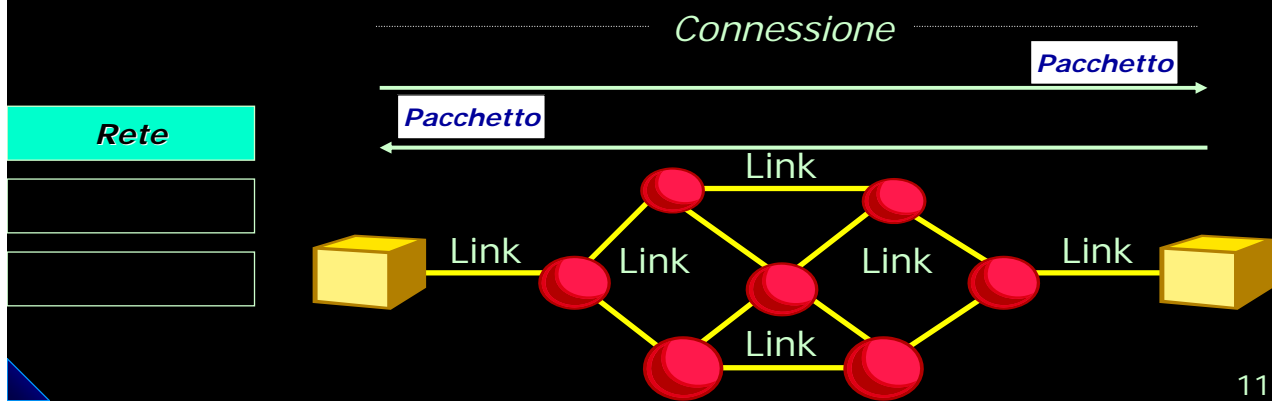
10



Note:

Funzioni del livello 3

- Routing;
- Gestione delle connessioni end to end;
- Rilevamento degli errori;
- Recupero degli errori;
- Controllo della sequenzialità;
- Controllo di flusso;
- Mapping fra indirizzi di rete e indirizzi di data-link;
- Mapping tra modi di trasferimento data link orientati alla connessione o connectionless.



11



Il Network Layer (NL) svolge un importante compito che è quello del routing. Il DLL mette a disposizione un canale di comunicazione affidabile per inviare dati tra computer collegati con un link fisico. Il NL deve quindi risolvere il problema di trovare un percorso per far giungere i dati all'HOST di destinazione. I cammini dei pacchetti possono essere basati su tabelle statiche che sono inserite all'interno della rete che vengono cambiate raramente. Essi possono essere determinati anche all'inizio di ogni conversazione, per esempio una sessione di terminale. Infine, possono essere determinati in modo dinamico per ciascun pacchetto, per prendere in considerazione il corrente carico di lavoro.

Se troppi pacchetti sono presenti nello stesso istante nella sottorete, daranno vita in ogni caso a qualche forma di congestione. Il controllo di tale congestione è responsabilità del livello rete.

Il network layer inoltre si occupa del set-up della connessione prima del trasferimento dei dati e del rilascio della stessa alla fine della comunicazione.

Infine, è compito del Network Layer gestire l'internetworking cioè, se un pacchetto deve attraversare reti di tipo diverso (per andare dalla sorgente alla destinazione) potrebbero insorgere alcuni problemi, quali l'indirizzamento e la lunghezza dei messaggi. Visto che gli operatori di sottorete possono avere bisogno di una numerazione dei pacchetti per realizzare i propri scopi, c'è spesso qualche funzione di numerazione inserita all'interno del livello rete. Nel dettaglio, il software deve contare quanti pacchetti, o caratteri o bit vengono inviati da ogni utente, per produrre le informazioni di riepilogo. Quando un pacchetto attraversa un confine nazionale, con diverse velocità fra le due parti, il conteggio può diventare complicato. Quando un pacchetto deve viaggiare da una rete a un'altra per raggiungere la propria destinazione, possono nascere molti problemi. L'indirizzamento utilizzato nella seconda rete può essere diverso da quello utilizzato nella prima. La seconda potrebbe non accettare il pacchetto perché troppo lungo. I protocolli potrebbero essere differenti, e così via. E' compito del livello rete superare tutti questi tipi di problemi per permettere il collegamento di reti eterogenee.

Ovviamente in semplici reti di tipo broadcast (reti in cui tutti i nodi collegati ricevono il dato trasmesso, e solo il destinatario del messaggio processa il messaggio stesso) in cui il problema del cammino da scegliere per far viaggiare i pacchetti (problema del routing) è semplice, il livello rete è spesso sottile o assente.

Servizi forniti ai livelli di elaborazione

Connessione tra user;

Indirizzi degli user;

Trasferimento dei messaggi;

Parametri di Qualità del Servizio;

Rilevamento dell'errore.



Elaborazione dei terminali



12



Il Transport Layer (TL) è il primo livello End-to-End, cioè esso non è presente nei nodi intermedi della rete, quindi si deve far carico di tutte le problematiche relative al trasferimento dei dati dalla sorgente alla destinazione. La funzione di base del livello di trasporto è di accettare dati dal livello superiore, spezzarli in piccole unità se è necessario, passare queste al livello rete, e assicurarsi che tutti i frammenti giungano correttamente a destinazione. Inoltre, tutto questo deve avvenire in modo efficiente, e in modo tale da isolare i livelli superiori da inevitabili cambiamenti nella tecnologia hardware. Le funzioni svolte da tale layer sono quindi frammentazione, controllo d'errore e controllo di flusso End-to-End.

Solitamente, il livello di trasporto crea una nuova connessione di rete (livello sottostante) per ogni richiesta di connessione di trasporto effettuata dal Session layer. Nel caso in cui venga richiesto un throughput (quantità di dati da trasferire) elevato per una particolare connessione di trasporto, in corrispondenza di essa potrebbero essere aperte due connessioni di rete, dividendo i dati tra esse per aumentare la capacità di trasmissione. Analogamente se vi sono due o più connessioni di trasporto che richiedono un basso throughput, esse possono essere mappate in un'unica connessione di rete. In ogni caso, il livello trasporto deve rendere la condivisione trasparente al livello superiore. Inoltre, per far condividere lo stesso canale a più sequenze di messaggi, il livello trasporto deve fare attenzione a stabilire ed eliminare connessioni lungo la rete. Questo richiede un meccanismo di denominazione, in modo tale che i processi su una macchina abbiano un modo per descrivere con chi desiderano conversare.

Il livello trasporto determina inoltre il tipo di servizi da fornire ai livelli superiori. Il più popolare tipo di connessione a livello trasporto è un canale punto-a-punto libero da errori che consegna messaggi o byte nell'ordine in cui sono stati inviati. Tuttavia, altri tipi di servizi a questo livello sono il trasporto di messaggi isolati senza garanzia sull'ordine di ricezione, o la diffusione del medesimo messaggio a destinazioni multiple. Il tipo di servizio è determinato quando la connessione viene stabilita.

Il livello trasporto è davvero punto-a-punto, dalla sorgente alla destinazione. In altre parole, un programma sulla macchina sorgente mantiene una conversazione con un programma simile sulla macchina destinazione, usando le intestazioni dei messaggi e i messaggi di controllo. Ai livelli inferiori, i protocolli sono situati fra ogni macchina e l'immediato vicino, e non fra la prima macchina sorgente e l'ultima macchina destinazione, che possono essere separati da diversi router. Questo è ciò che differenzia i livelli da 1 a 3, che sono collegati, dai livelli da 4 a 7, che sono in realtà punto-a-punto.

Ci deve inoltre essere un meccanismo per regolare il flusso di informazioni, in modo tale che un host veloce non sovraccarichi uno lento. Questo meccanismo è chiamato controllo del flusso e gioca un ruolo chiave nel livello di trasporto (come in altri livelli). Il controllo del flusso fra host è distinto dal controllo del flusso fra router, tuttavia si vedrà più avanti che principi simili si applicano a entrambi.

Per concludere, il livello di trasporto funge un po' da livellatore; in base alla qualità del servizio (QOS - quality of service) messa a disposizione dal network layer, esso ha 5 classi di servizio (classes of services). La classe 0 provvede solo le funzionalità base per creare una connessione e trasferire dati, mentre la classe 4 provvede una connessione affidabile e prevede delle procedure per il flow control.

Livelli di elaborazione: Sessione

Non vengono processati dalla rete, ma solo dai terminali

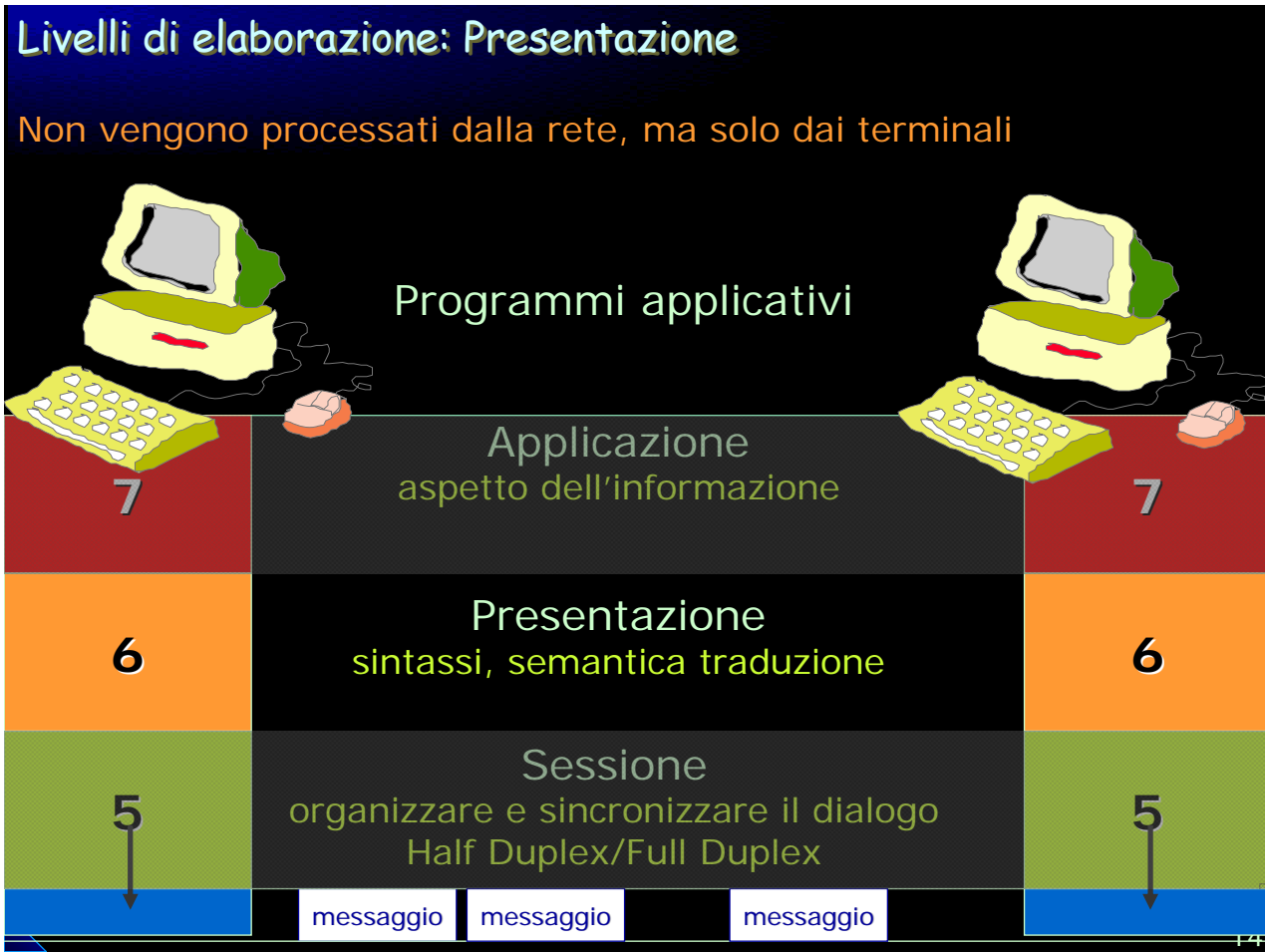


Session Layer

Nonostante con il Transport Layer vengono esaurite le problematiche del trasferimento affidabile dell'informazione, i servizi fino ad ora a disposizione non permettono a due qualunque processi applicativi di colloquiare. Il Session Layer (SL) organizza e sincronizza il dialogo delle entità di livello superiore consentendo ad utenti su macchine diverse di stabilire sessioni. Una sessione permette il trasporto ordinario di dati, realizzato dal livello trasporto, ma anche servizi avanzati utili a talune applicazioni. Una sessione potrebbe essere utilizzata per permettere a un utente di collegarsi a un sistema condiviso remoto o di trasferire archivi fra due macchine. Uno dei servizi del livello sessione è di gestire il controllo del dialogo. Le sessioni possono permettere al traffico di viaggiare in entrambe le direzioni allo stesso tempo, o solamente in una direzione alla volta. Infatti nel caso in cui la comunicazione può avvenire secondo la tecnica half-duplex (entrambe le entità possono trasmettere ma in modo alternato) il Session Layer gestisce i turni di utilizzo delle risorse: il livello sessione tiene traccia di chi è il turno attuale.

Un servizio legato al livello sessione è la gestione dei token. Per alcuni protocolli, è essenziale che entrambe le estremità non cerchino di effettuare la medesima operazione contemporaneamente. Per gestire queste attività, il livello sessione fornisce dei token che possono essere scambiati. Solamente l'estremità in possesso del token può eseguire l'operazione critica. Un altro servizio del livello sessione è la sincronizzazione.

Un'altra funzione utile svolta da tale livello è la sincronizzazione. In caso di crash, solo i dati trasferiti dopo l'ultimo punto di sincronizzazione dovranno essere ritrasferiti. Si consideri il problema che si può avere quando si cerca di effettuare un trasferimento di un archivio che impiega due ore fra due macchine che hanno un'ora come tempo medio fra guasti. Se un trasferimento si interrompe e fallisce, l'intero trasferimento dovrebbe iniziare daccapo e potrebbe a sua volta fallire. Per eliminare questo problema, il livello sessione fornisce un modo per inserire dei punti di controllo in una sequenza di dati, in modo tale che, dopo un guasto, solamente i dati trasferiti dopo l'ultimo punto di controllo debbano essere ritrasmessi.



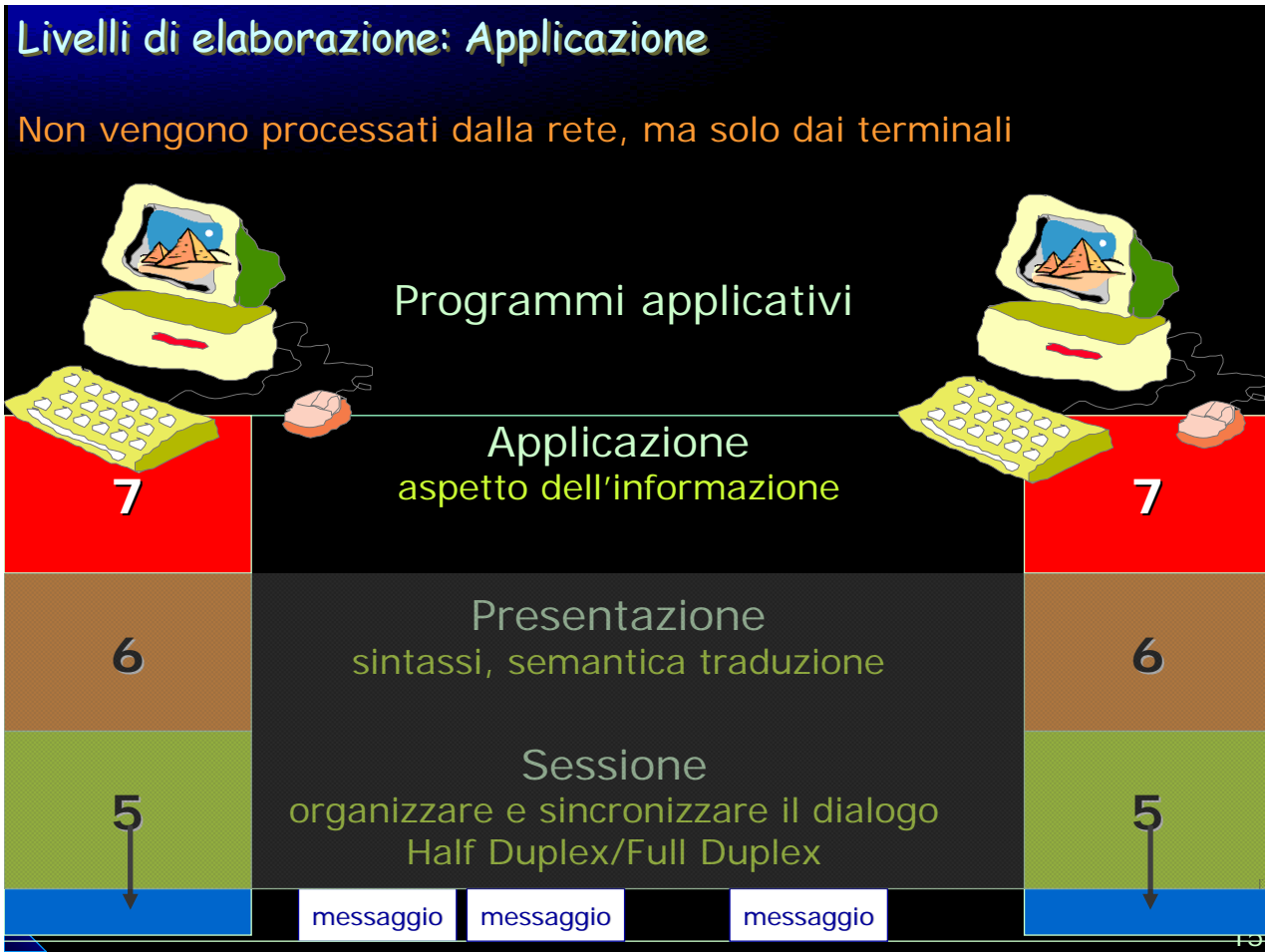
Presentation Layer

Il livello di presentazione realizza alcune funzioni che sono utilizzate abbastanza spesso da richiedere una soluzione generale, invece che lasciare a ogni utente il compito di risolvere tali problemi in modo autonomo. In particolare, diversamente dagli altri livelli sottostanti, che sono interessati solamente a muovere bit in modo affidabile da un punto a un altro, il livello presentazione si occupa della sintassi e della semantica delle informazioni da trasferire.

Un esempio tipico di un servizio del livello di presentazione è la codifica dei dati in un modo standard riconosciuto. La maggior parte degli utenti non scambia stringhe casuali di bit binari. Scambiano nomi di persone, date, quantità di denaro e fatture. Queste informazioni sono rappresentate da stringhe di caratteri, interi, numeri a virgola mobile e strutture dati composte di diversi componenti più semplici. Calcolatori differenti hanno codici differenti per rappresentare stringhe di caratteri (ad es. ASCII e Unicode), interi (ad es. complemento a 1 e complemento a 2), e così via. Per rendere possibile la comunicazione fra calcolatori con rappresentazioni differenti, le strutture dati che devono essere scambiate devono essere definite in modo astratto, in modo tale che una codifica standard possa essere utilizzata in modo dinamico. Il livello presentazione gestisce queste strutture di dati astratte e converte dalla rappresentazione usata all'interno del calcolatore alla rappresentazione standard della rete e viceversa.

Il livello di presentazione negozia quindi una sintassi di trasferimento da usare per il colloquio. Se tale sintassi è differente da quella locale, allora dovrà effettuare una conversione prima di trasmettere (o ricevere) i dati. Tale comportamento è molto simile a quello che succede quando due persone vogliono dialogare. Se una di esse parla solo l'italiano mentre l'altra solo il giapponese, sarà necessario il ricorso a degli interpreti. Bisogna però notare che è necessario un linguaggio universalmente compreso che permetta l'accordo sul linguaggio di trasferimento.

Un'altra funzione svolta dal presentation layer è il criptaggio dei dati.



Application Layer

Nonostante si ha già con il Presentation Layer una lingua comune con la quale colloquiare, affinché il dialogo sia costruttivo è necessario che vi sia condivisione dei contenuti del discorso comune. Continuando l'esempio precedente dei due interlocutori (italiano e giapponese), se l'italiano è un conoscitore dell'arte moderna mentre il giapponese è un scienziato esperto di esperimenti nucleari, sarà difficile che il dialogo fra i due interlocutori sia costruttivo.

L'Application layer è il livello più alto dell'architettura OSI, e il suo compito è dunque quello di mettere a disposizione dei programmi un insieme di funzionalità e procedure che in modo da far apparire il sistema di comunicazione come se fosse un dispositivo locale, rendendo il tutto trasparente all'utente.

Il livello di applicazione contiene una varietà di protocolli che sono normalmente necessari. Per esempio, ci sono centinaia di tipi di terminali non compatibili nel mondo. Si consideri la situazione limite di un editor a tutto schermo che deve lavorare su una rete con molti tipi di terminali differenti, ognuno con una diversa gestione dello schermo, diverse sequenze di escape per l'inserimento e la cancellazione di testo, diversi modi per muovere il cursore ecc. Un modo per risolvere questo problema è di definire l'astrazione di un terminale virtuale di rete. Per gestire ogni tipo di terminale, una parte di software deve essere scritta per realizzare il terminale virtuale sul terminale reale. Per esempio, quando l'editor muove il cursore del terminale virtuale nell'angolo superiore sinistro dello schermo, questo software deve inviare il comando appropriato al terminale reale per far spostare in quel punto il suo cursore. Tutto il software del terminale virtuale costituisce il livello applicazione.

Un'altra funzione del livello applicazione è il trasferimento dei file. File system differenti hanno differenti convenzioni per nominare gli archivi, diversi modi per rappresentare linee di testo, e così via. Il trasferimento di un file fra due differenti sistemi richiede di gestire queste e altre incompatibilità. Anche questo tipo di lavoro è a carico del livello applicazione, come la posta elettronica, la gestione remota di processi, la ricerca di directory, e molte altre funzioni di uso generale o specifico.