

RETI: Architettura della connessione

Misurata dal costo, velocità ed affidabilità di invio dei messaggi da dove provengono a dove devono andare

Parametri:

- **latenza**
- **banda**
- connettività
- costo hw
- reliability
- funzionalità
(e.g. combinazione messaggi, frammentazione)

Obiettivo

MIGLIORE USO RISORSE E PERFORMANCE

Anche modelli complessi per tenere conto del tutto

LOGP considera anche i contributi di overhead dei diversi nodi intermedi ed estremi

teoria delle code

SCELTE

Topologia

statica e dinamica

Modi di operazione della architettura

sincrono ed asincrono

Switching

packet switching, circuit switching,
virtual cut-through (wormhole)

Strategia di Controllo

centralizzato o distribuito

Latenza vs banda

I due fattori o indicatori costituiscono la metrica fondamentale per considerare un buon uso delle risorse

Banda di trasmissione (throughput)

*quantità di dati trasmessi con successo (per secondo)
su un link o una connessione*

Ethernet 10Mbps (numero puro su intervallo)

10 Mbit al secondo -> 1 bit 10µs

Tempo di latenza di trasmissione

*tempo impiegato per trasmettere una unità
di informazione (bit)*

tempo di andata/ritorno (Round Trip Time o RTT)

rete locale 25µs

Linea intercontinentale 25ms

$$T_L = T_{prop} + T_{tx} + T_q$$

T_{prop} dipende dalla **velocità** della luce nel mezzo
 $T_{prop} \approx \text{Spazio} / \text{Velocità}$

T_{tx} dipende dal **messaggio** e dalla **banda**
 $T_{tx} \approx \text{Dimensione} / \text{Banda}$

T_q dipende dai **ritardi** dovuti all'accodamento in
diversi punti della connessione
(*endpoint o intermedi*)

Problemi e ottimizzazioni

crescita infinita Banda ma limite a Latenza

identificazione dei **colli di bottiglia** ossia di quale fattore domina e si deve considerare
la gestione delle risorse deve trattare soprattutto questo

se invio/ricezione di 1 byte
dominante la latenza

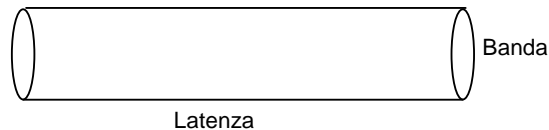
RTT banda non importante

se invio/ricezione di molti MegaByte 30MB
dominante la banda

throughput latenza non importante

IMPEGNO RISORSE

Prodotto **Latenza x Banda** risorsa canale dati



latenza 40ms e banda 10Mbps
prodotto 50 KBps (400 Kbps)

è necessario che il mittente invii **50KB** prima che il primo bit sia arrivato al destinatario
e prima di una risposta 100KB

BUFFERING

La infrastruttura potrebbe mantenere le pipe piene per con proprie risorse i tempi di risposta

Nel caso di ETHERNET

ci sono limiti alla dimensione del **frame**

almeno 64b (512 bit)
meno di 1500b

Se il frame fosse di **dimensione non limitata**
un nodo potrebbe tenere occupato il mezzo per sempre
(nessuna garanzia di **fairness**)

Se il frame fosse **troppo corto**, il messaggio non avrebbe tempo di propagarsi fino al punto più lontano (max 2500m) e il mittente potrebbe non identificare la collisione
Il mittente deve mandare almeno bit da saturare il mezzo

propagazione 12.5µs (considerando 2500m e ritardi)
25µs poi stando larghi **51,2µs** e
banda **10Mbps**

prodotto **512 bit**

ossia 64 byte, di cui
14 header frame MAC +
40 header TX e IP +
6 almeno applicazione +
4 footer MAC

In caso di mezzo occupato, si trasmette appena si libera
(**1-persistente**) senza aspettare

Se ancora collisione, attesa di un tempo con raddoppio intervallo (**backoff esponenziale**)

Primo tempo attesa tra 0 e 51,2 µs

Secondo tempo attesa tra 51,2 e 102,8 µs

...

Fino a 16 tentativi, poi eccezione

ORGANIZZAZIONI INTERNAZIONALI

settore Comunicazione pubblica ITU International Telecommunication Union

ITU definisce i modelli di comunicazione per servizi telefonici

modelli di **comunicazione** per il settore telefonico

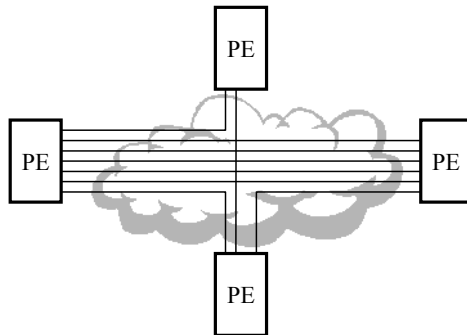
- DTE: interfaccia cliente
(data terminal equipment)
- DCE: interfaccia di rete
(data circuit terminating equipment)

modelli a **connessione**

- livelli L/N/T della pila OSI
con messaggi di setup e negoziazione

problemi di **controllo**

- scambio informazioni di monitor/controllo



ITU definisce tutti i protocolli tra DTE e DCE
V.24, X.21, X.25

Switching

impegno condiviso delle risorse per consentire di passare i dati in caso di nodi non in visibilità

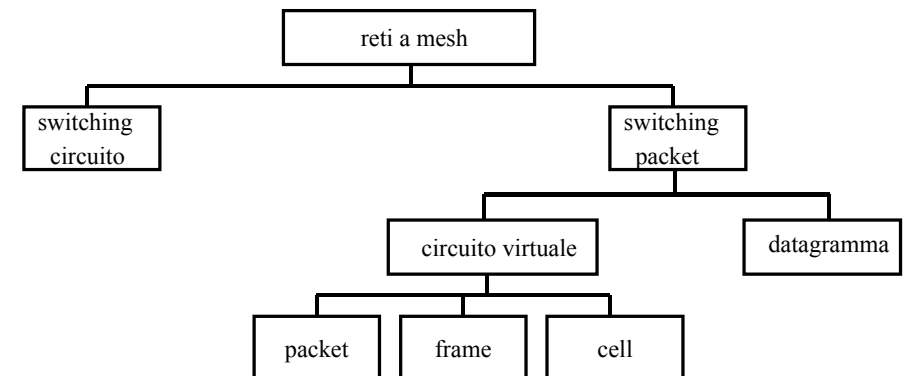
La comunicazione impegna:

- o collegamenti statici dedicati
- o collegamenti switch da stabilire e controllare
Permanent / Switched Virtual Circuit
- o usa risorse esistenti (ottimista)

Circuito impegno risorse intermedie

Messaggi singoli

con tecniche di **invio/spezzettamento/routing**
stream di byte/bit o pacchetti



Scambio di informazioni diverse di dimensioni determinate

- cella** insieme di 53 byte
- frame** unità del livello data link
- pacchetto** messaggio a livello network con dimensione da 100 a 1000 byte
- datagramma** associato alla comunicazione senza connessione
- messaggio** a livello applicativo

SWITCHING

come si impegnano le risorse circuito/datagramma

Switching di circuito (tecnica pessimista)

canale end-to-end per flusso di bit (anche circuiti multipli)

- impegno di risorse anche senza flusso di dati
- multiplexing inverso
combina N canali fisici per uno logico utente

schema molto statico e proattivo

Esempio: ISDN

Circuiti virtuali

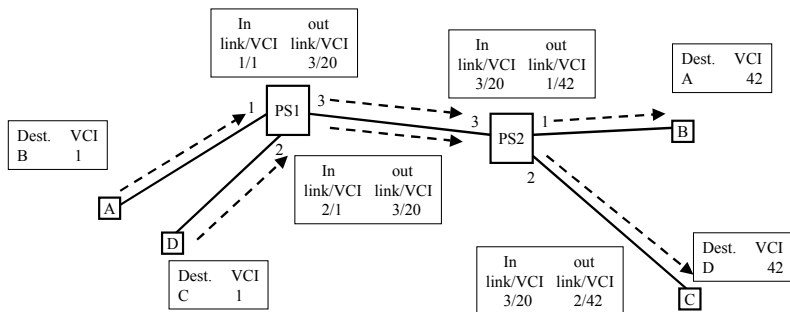
(condivisione ottimista)

connessioni end-to-end

trasmissione dati a pacchetti
anche con connessioni multiple

con virtual circuit identifier (VCI)

identificatori locali unici negli intermedi con eventuale
condivisione



SWITCHING OTTIMISTA (CONDIVISIONE)

Switching a datagrammi (tecnica ottimista)

nessun connessione end-to-end

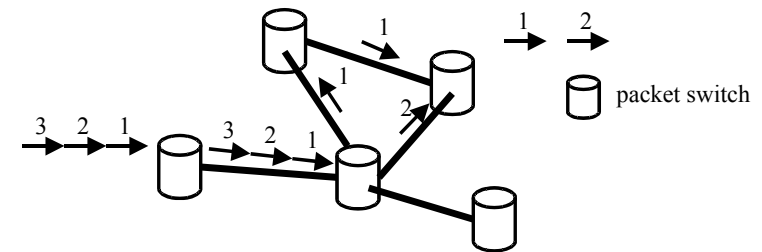
nessun controllo flusso, congestione, errore

nessuna garanzia di servizio

(no ordine, no QoS)

ogni pacchetto porta indirizzo

effetti di ritardo e jitter



con i datagrammi perdiamo le garanzie offerte dai canali

Si distingue

circuit switching
packet switching

impegno risorse

ad ogni nodo si attende il messaggio, lo si memorizza (store) e lo si reinvia (forward) -> **store-and-forward**

virtual cut-through

si impegna memoria ad ogni nodo
si manda il messaggio ma lo si muove solo o in caso di **successo completo** (non si impegna memoria intermedia) o di **interferenze** (uso memoria)

wormhole

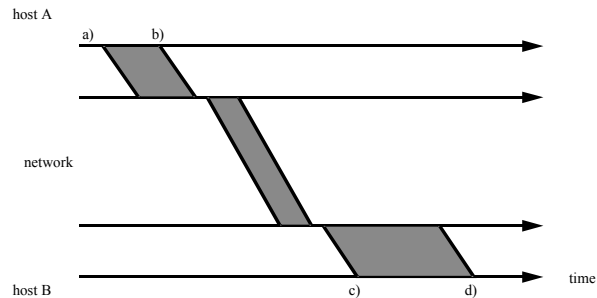
si manda **un solo pezzetto** del messaggio (flit) che attende la soluzione di eventuali conflitti ed interferenze

Pacchetto (tecnica ottimista)

pacchetto di dimensione fissa

con condivisione delle risorse e multiplexing del traffico
bande diverse e velocità diverse

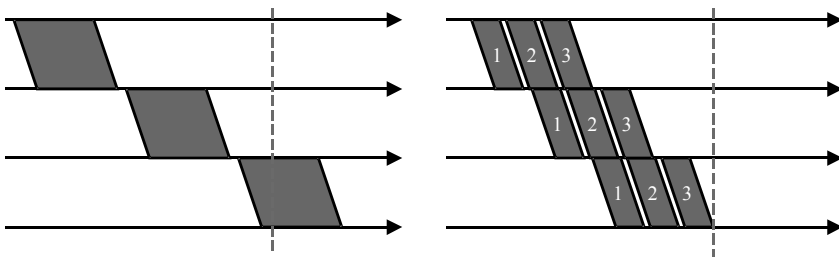
invio di un pacchetto alla volta



Messaggi (tecnica ottimista)

con messaggi di lunghezza diversa

Un solo messaggio può comportare molti pacchetti
usato a livello di applicazioni (es. e-mail)



Per ogni nodo store-and-forward

ritardo introdotto ad ogni switch

messaggi lunghi ritardano quelli corti

Tecniche di ottimizzazione

Switching a Frame (frame relay)

circuito virtuale di Network costruito sulla base di
un circuito virtuale a pacchetti

controllo errore e flusso ripetuto ai livelli 2 e 3

solo livello 2 link end-to-end

Eliminazione di overhead di garanzia QoS

- Frame relay

nessun controllo errore o flusso a livello 2

controlli lasciati a livello applicativo

Switching a Cella

pacchetti piccoli e fissati detti celle (ATM)

reattivo: si adatta al traffico

switch veloci e pacchetti fissi

non indirizzi globali ma VCI

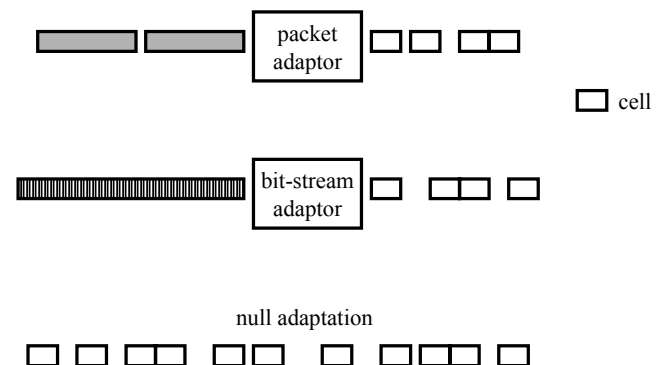
nessun controllo errore e flusso

celle di dimensioni fissate

multiplexing granulare dei servizi

- adattamento

pacchetti celle e bit-stream celle



Signalling controllo dell'operazione

Necessità di

- stabilire/chiudere una connessione
- requisiti QoS della connessione
- gestire le risorse da usare in connessione

Signalling (o Controllo)

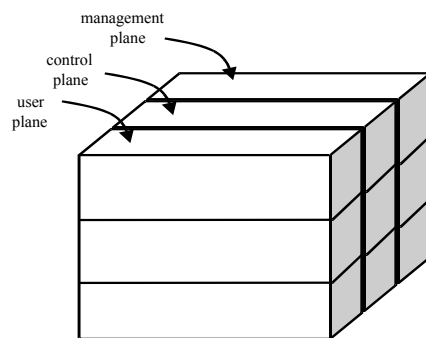
- **in-band**:
usando gli stessi cammini dei dati
- **out-of-band**: cammini separati controllo

Common Channel Signalling (CCS)

segnalazione out-of-band per tutti i canali

Signalling in piani di specifica

- **Piano User**
protocolli utente
- **Piano di Controllo**
controllo operazioni di connessione
user-to-network e network-to-network signalling
- **Piano di Management**
 - network management and monitoring
 - fault identification



Modi di comunicazione

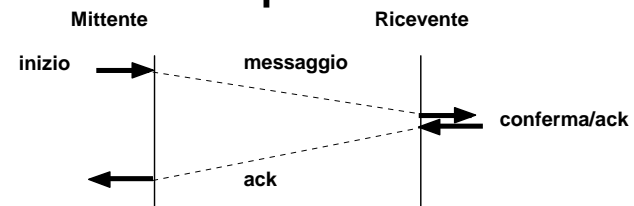
dobbiamo considerare due dimensioni

- affidabilità**
- asincronismo**

per svincolare il servizio e
consentire asincronismi

Automatic Repeat reQuest (ARQ)

Stop and wait



casi di *ritrasmissione con time-out*

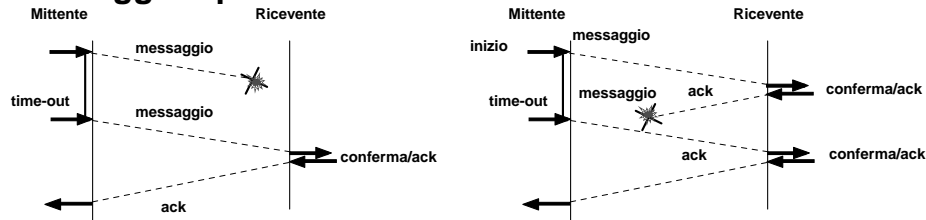
Naturalmente dobbiamo considerare la possibilità di avere anche **errori**:

- sia **perdite di messaggi**
- sia **messaggi compromessi**

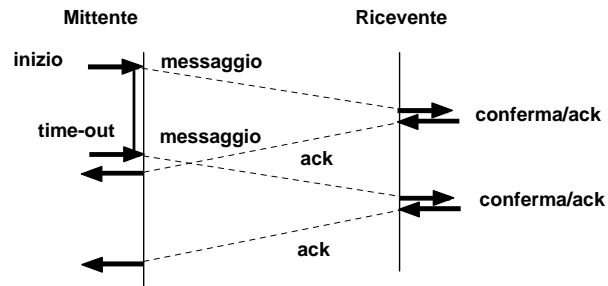
Si intende ottenere il massimo della **asincronicità** e della **autonomia** di decisione tra i pari che devono comunicare

diversi casi

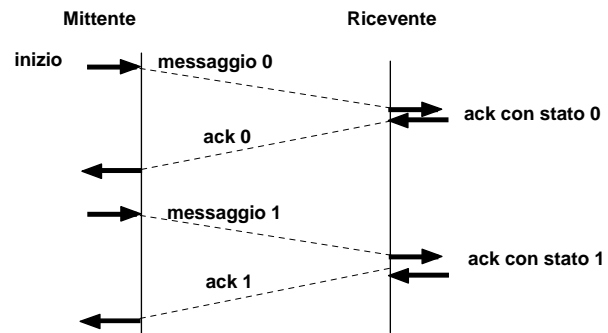
messaggio ripetuto



ack ripetuti per lo stesso messaggio



Uso di identificazione del messaggio



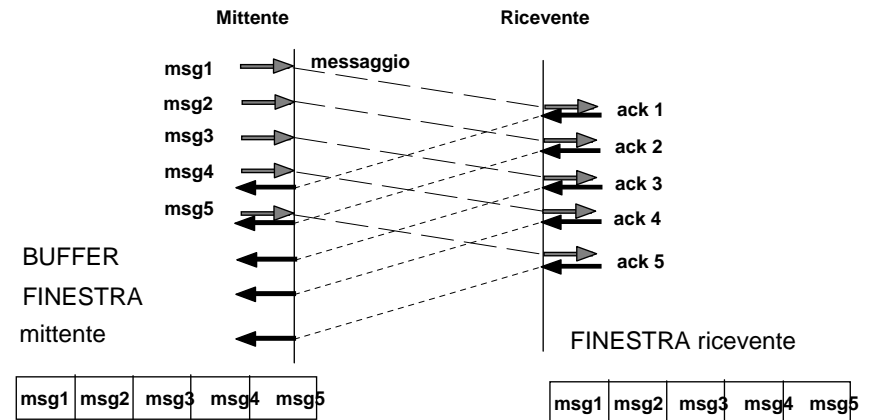
Continuous Request

Si mandano messaggi che sono bufferizzati fino a saturare la risorsa memoria (**finestra buffer**) disponibile

Il mittente scorre la finestra solo all'**acknowledgement**

Attesa del mittente solo a **finestra piena**

La dimensione della finestra **imposta da chi ?**



In caso di full-duplex, gli ack sono mandati in **piggybacking** sul traffico nello stesso senso

Cosa succede in caso di errore o di messaggio non arrivato mentre altri messaggi inviati dopo sono arrivati?

due modi:

GO-BACK-N

attesa del messaggio (timeout al mittente) e scarto degli altri **successivi** già arrivati non in sequenza

SELECTIVE RETRANSMISSION

attesa del messaggio tenendo conto dei successivi non in sequenza cui si da ack (timeout al ricevente)

go-back confonde messaggi non in ordine con perdite

(HDLC usa go-back-N e selective retx)

(TCP usa go-back-N e ack cumulativi)

Finestra scorrevole (sliding window)

Usata da TCP per ogni direzione
per un verso di trasmissione

il Mittente (TX)

numera ogni messaggio con **NumSeq**

decide una dimensione della finestra **TXWinSize**

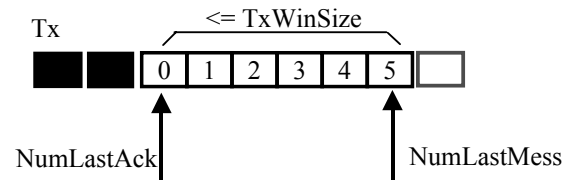
mantiene il valore dell'ultimo messaggio inviato

NumLastMess

mantiene il valore dell'ultimo ack ricevuto **NumLastAck**

Obiettivo è mantenere

$$\text{NumLastMess} - \text{NumLastAck} + 1 \leq \text{TXWinSize}$$



il Ricevente (RX) tiene

una dimensione della finestra **RXWinSize**

il numero del prossimo messaggio atteso

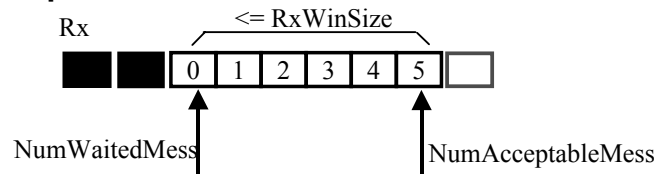
NumWaitedMess

il numero dell'ultimo messaggio confermabile

NumAcceptableMess

Il Ricevente deve mantenere

$$\text{NumAcceptableMess} - \text{NumWaitedMess} + 1 \leq \text{RXWinSize}$$



Molte politiche e decisioni differenti

Si noti che può anche succedere che le informazioni siano non consistenti tra i due pari

INTERCONNESSIONE

Statica (reti dirette)

Dinamica (reti indirette)

INTERCONNESSIONE GLOBALE

Libera e non Vincolata

interconnessione completa

unico crossbar

interconnessioni irregolari replicate

INTERCONNESSIONE RISTRETTA

Vincolata

bus unico

reti dirette statiche

anelli - ring

interconnessione nel piano - mesh

alberi (binari, ternari, etc.) - tree

ipercubi

cube connected cycles (CCC)

reti a stage dinamiche

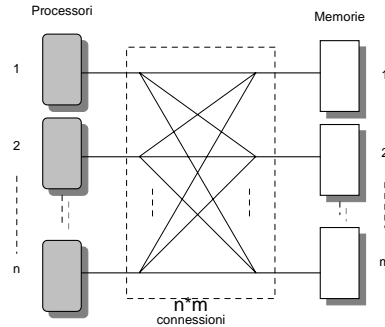
omega network

banyan network

Benes network

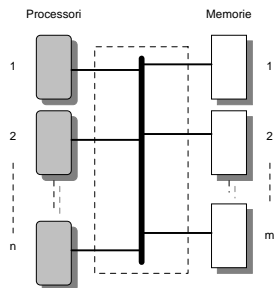
INTERCONNESSIONE

Connessione completa

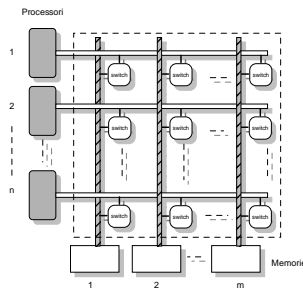


Ogni processore collegato con tutti i banchi di memoria
 $n \cdot m$ connessioni
 In un sistema ideale è realizzato attraverso
un bus o **con switch**

Numero modesto di processori (ordine della decina)



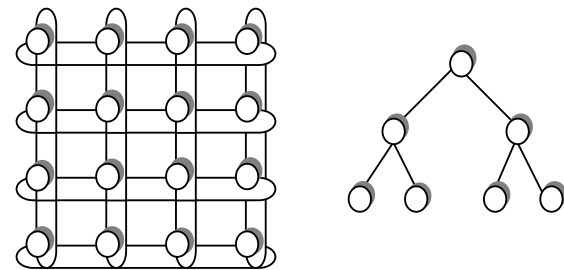
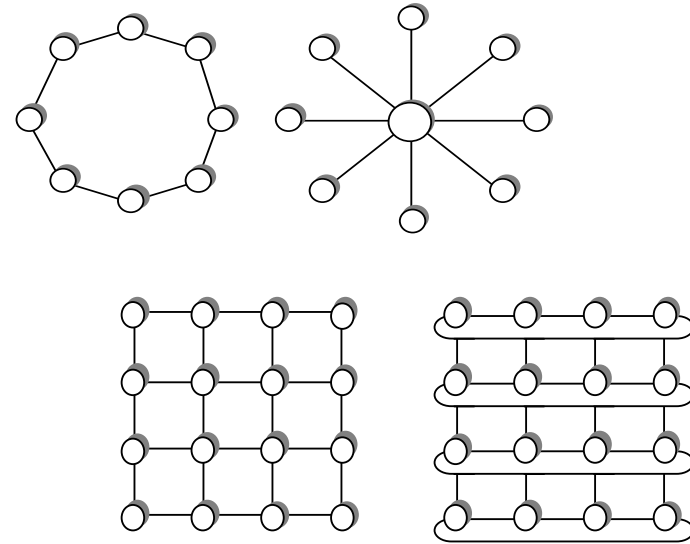
Sistema shared bus



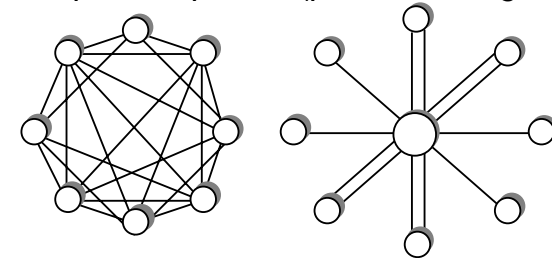
Sistema crossbar switch

Problemi di scalabilità risolti con ipotesi di località
 limiti sul numero dei partecipanti
 Eventuale aggiunta di interconnessioni ulteriori
connessioni multiple replicate

RETI dirette

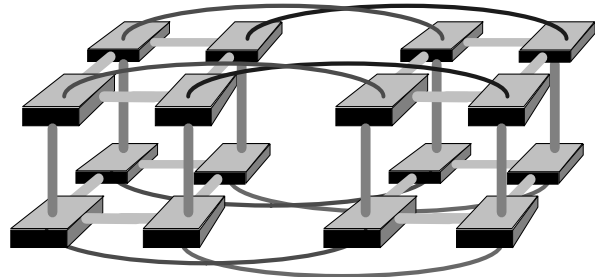


Fino a strutture più complesse (per tollerare guasti)



anche con ridondanze parziali
 Possibile imposizione del verso sui link

Ipercubi



ipercubo di dimensione 4 (due nodi per dimensione)

ring => con due nodi

mesh => con due nodi per dimensione

In genere, nodi totali 2^n di un ipercubo

un ipercubo ha due nodi per ogni dimensione

Si estende a numeri superiori di nodi sulla stessa dimensione => se $K = k$ per ogni dimensione

K-ary n-cube nodi totali k^n

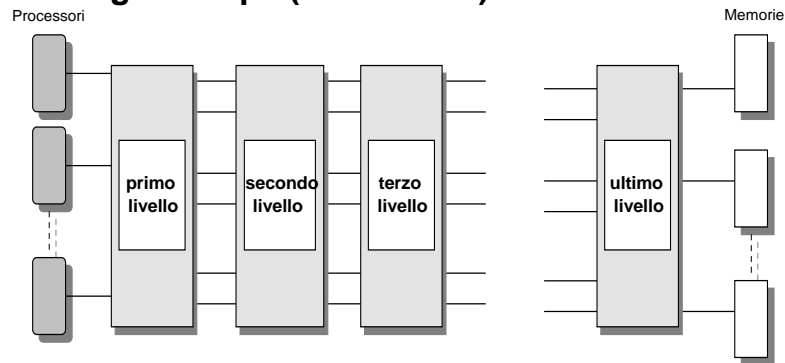
un ipercubo di dimensione n con un numero di nodi

Grado di interconnessione **non scalabile**

Reti Dinamiche

Ogni stage può connettere all'input una di diverse uscite

RETI a stage multipli (INDIRETTE)



Conflitti e ritardi di **setting** e attraversamento

Reti locali

si possono analizzare

- **topologie**
- **controllo di accesso**
- **mezzo trasmissivo**
- **applicazioni**
- **standard**

In rete locale =>

facilità di broadcast

bassa probabilità di errori

alta velocità ed ampia banda di trasmissione

Notate che il grosso interesse degli ultimi anni non è tanto per la singola rete, quanto per la **interconnessione di reti**

A livello di industria e finanza pubblica

Interconnessione di reti

LAN (Local Area Network)

MAN (Metropolitan Area Network)

WAN (Wide Area Network)

Si possono avere reti statiche e anche riorganizzate dinamicamente

Inoltre, le **reti fisse** possono anche prevedere appendici mobili (con reti varie)

WLAN (Wireless Local Area Network)

WPAN (Wireless Personal Area Network)

topologie usate per LAN

In reti generiche (Wide Area Network) molte e diverse
mesh Public Switch Telephone Network (PSTN)

In LAN topologie semplici

stella

Private Automatic Branch Exchange (PABX)

Private Digital Exchange (PDX)

bus anche un insieme di bus interconnessi

ring

hub

un **bus** inglobato in una unica unità centrale di connessione ==> simile ad una stella

realizzazione poco costosa a unico ring interno

evoluzione verso sistemi con connessioni dinamiche ad alte prestazioni

mezzi trasmissivi

doppino schermato o non schermato e cavo coassiale

	10BaseT	100BaseT	1000BaseX
banda	10Mb/s	100Mb/s	1000Mb/s
CAT5 UTP	100m+	100m	100m(?)
STP/coax	500m	100m	25m
fibra multi-mode	2Km	412m (hd) 2Km (fd)	500m
fibra single-mode	25Km	20Km	3Km (5Km?)

Controllo di accesso

Carrier Sense Multiple Access/Collision Detection
(CSMA/CD) (Ethernet)

token (**control token**)

anello a slot (**slotted ring**)

CSMA/CD accesso ottimistico

sistema dinamico di impegno del mezzo

collisioni ==>

recovery con ritrasmissione ad intervallo random

control token accesso pessimistico

sistema statico:

un solo possessore del diritto di trasmettere (token)

passaggio del token da un vicino ad un altro

slotted ring accesso pessimistico

controllo statico di accesso ad un anello:

anello come insieme di contenitori di messaggi circolanti (slot)

Standard IEEE 802

CSMA/CD (802.3)

token ring (802.5) token bus (802.4)

I comitati stanno estendendo al wireless

802.11 802.11x

(802.11b WiFi 802.11g)

Diffusione di Ethernet

sistema **reattivo** e non **proattivo**

Performance

Fino ad occupazione media del mezzo (50%)
prestazioni equivalenti
con throughput più alto per messaggi più lunghi
Ad alto traffico, qualche differenza

Interconnessione reti

necessità di nuove tecnologie o mezzi trasmissivi

con una possibilità di trasmettere informazioni
velocemente in modo dinamico
Occupando risorse solo in **caso di utilizzo**
garantendo determinati **livelli di servizio o QoS**
(Quality Of Service)

Ethernet (10Mbps) su coassiale thick o thin
cavo giallo **10base5** (lunghezza segmento 500m)
cavo sottile **10base2** (lunghezza segmento 200m)
twisted pair **10baseT** (lunghezza segmento 100m)
fibra **10baseF** (lunghezza segmento 1500m)

Fast Ethernet (100Mbps) su doppino **100baseT**
CAT5 UTP Unshielded Twisted Pair

Non solo reti ad alta velocità
Nuove tecnologie: **FDDI, ATM**

FDDI

fibre distributed data interface
Standardizzazione ANSI compatibile IEEE 802
banda 100Mb/s ma indirizzi MAC a 48 bit

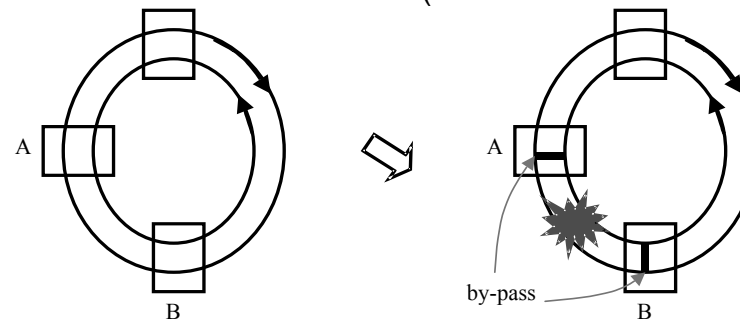
- FDDI-1: packet switching
- FDDI-2: circuit switching

si definiscono il numero massimo di stazioni, massima lunghezza tratta, massima lunghezza frame, etc.

Due anelli, gestiti in verso opposto
uno con funzioni di **back-up**
Si agganciano le stazioni ad entrambi o ad uno solo

Elevato costo di installazione
uso come dorsale di interconnessione con concentratori

Due tipi di traffico **sincrono ed asincrono**
uno in real-time, l'altro con minore qualità
Uso di token che ruota tra i nodi (**Token Rotation Time**)



traffico sincrono da garantire
traffico asincrono se c'è tempo

Ogni nodo mantiene il token per un periodo **THT** (token holding time) per la trasmissione dati sincroni

Accesso al ring con token temporizzato

RING gestito in modo reattivo

ogni stazione mantiene un tempo concordato come obiettivo ideale: **target token rotation time TTRT** confrontato con **misura** del **token rotation time TRT**

$$TRT < \text{numeroNodi} \cdot THT + T_{\text{LatenzaRing}}$$

Se il **TRT** misurato supera **TTRT**, si invia il solo traffico sincrono per l'intervallo prefissato, e si passa subito il token, altrimenti, lo si trattiene per trasmettere il traffico asincrono

Dettaglio

dati sincroni e asincroni

Meccanismi per il calcolo della somma **SAT < TTRT**

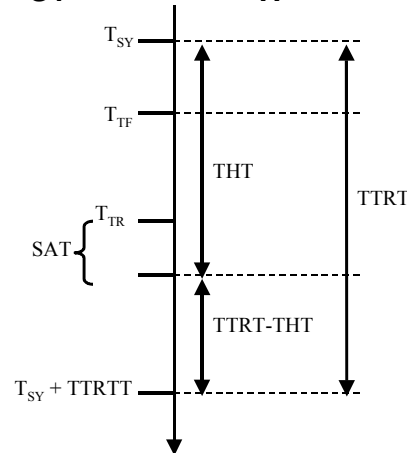
Inviati **sincroni** per ogni acquisizione di token per un tempo **TTRT - TRT**

Il tempo per dati asincroni è **TTRT - THT**

considerato il tempo di *token holding time* **THT**

if $(TTRT - THT) == 0$, forward token senza inviare dati

$$T_{SY} + TTRT < T_{TF} + TTRT$$



ISDN Integrated Service Digital Network

circuit switching

adatto per informazioni di tipo multimediale

si forniscono **bit stream end-to-end**

Narrowband N-ISDN

canale di 64Kbit/s B-channel

canale di 616Kbit/s D-channel

per la segnalazione

Basic Rate-ISDN 2B+D

Primary Rate 30B+D

gerarchia di interfacce per ottenere lo stabilirsi delle connessioni

La diffusione dei servizi ISDN nasce dalla integrazione dei servizi offerti da un unico fornitore di servizi, per ambiti molto differenziati come:

- telefono
- musica
- audio
- video
- internet

e sfruttando le stesse risorse

Asynchronous Transfer Mode (ATM)

multiplexing asincrono in tempo

adatto per informazioni di tipo multimediale
e supportato come **B-ISDN**

(Broadband Integrated Service Digital Network)

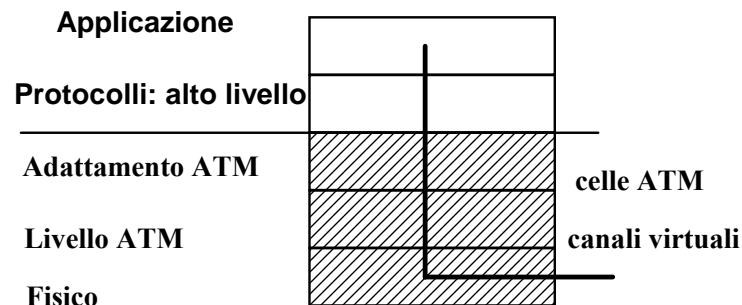
packet-switching molto veloce basato su **celle** in grado di scambiare messaggi di dimensione fissa senza imporre controllo di flusso e error check
ogni cella specifica le informazioni di instradamento solo via **NSAP** (Network Service Access Point)

connessione virtuale alimentata da celle diverse che prelevano messaggi di tipo diverso

trasporto di **blocchi** di 53 byte
5 header e 48 di informazioni utili

header contiene informazioni di routing e priorità

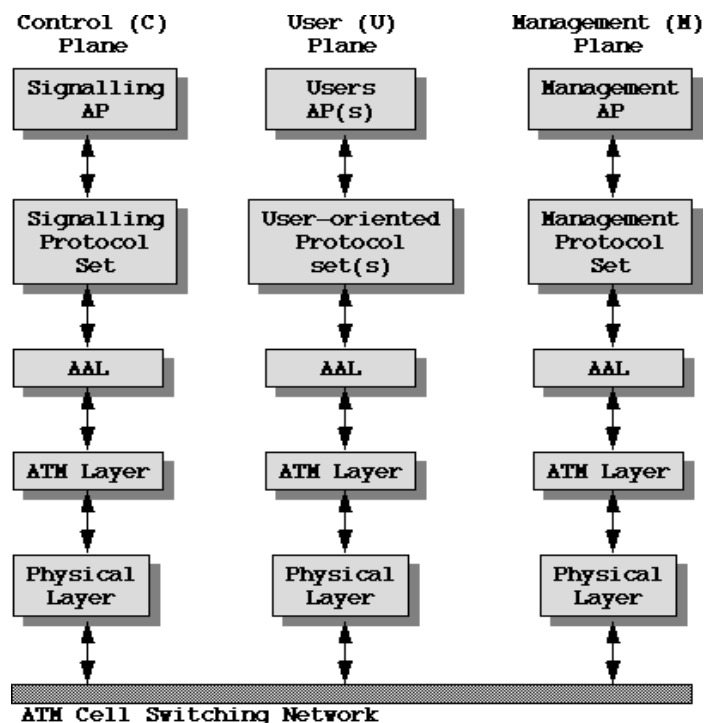
Struttura a livelli



ATM

Livello OSI	Livello ATM	Sublayer ATM	Funzionalità
APPLIC.	applicazioni che in genere non utilizzano direttamente le celle ma passano ai livelli sottostanti blocchi di informazioni più grandi		
PROT. ALTO LIV.	protocolli che usano ancora blocchi più grandi delle celle		
3/4	AAL (ATM adaptation layer)	CS sublayer convergence	offre diversi tipi di servizio, con diverse necessità di trattamento informazioni (errori, ecc.)
		SAR segmentation & reassembly	suddivide i pacchetti in celle e viceversa
2/3	ATM		controllo flusso, controllo congestione, multiplexing/demultiplexing, generazione/estrazione header celle, gestione circuito virtuale,
2	Fisico	TC transmission convergence	conversione stream di bit in celle e viceversa (frames, checksum, ecc)
1		PMD phys. medium dependent	Gestione mezzo tramissivo

piani diversi ATM



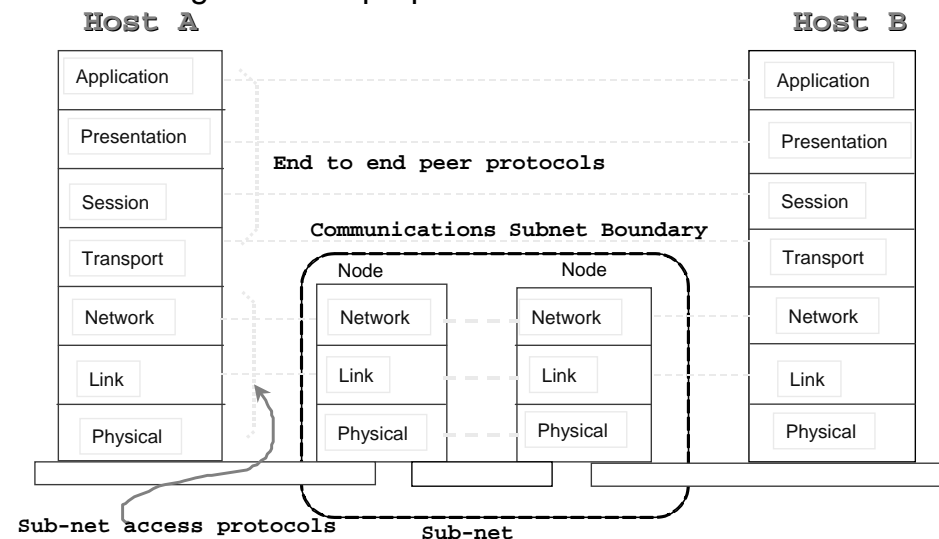
Distinzione tra i piani
di **management**
di **monitoraggio / controllo**
di **utente**

Garanzia di qualità di servizio diverse
classe A con banda costante
altre classi con banda variabile

OSI

STANDARD di comunicazione

Obiettivi di **Intercomunicazione**
tenendo conto di reti e sottoreti
e di tecnologie diverse proprietarie



I produttori forniscono architetture che affrontano i problemi
della intercomunicazione in modo completo (a molti livelli)

IBM SNA, DEC DNA, XEROX XNS

Necessità di andare verso sistemi **aperti**

NON sistemi **chiusi**, ossia riconosciuti da un produttore e
confinati a una architettura e un sistema specifico

ESEMPI di sistemi APERTI

UNIX che non lega ad un produttore, con software free
(**open source**)

protocollo di comunicazione **X.25**

Standard OSI

per ogni livello si specifica

- **definizione del servizio** (descrizione verticale)
definizione astratta dei servizi del livello corrente disponibili al livello immediatamente superiore
in altre parole, l'**interfaccia** offerta dal livello stesso
accesso al servizio
- **specificazione del protocollo** (descrizione orizzontale)
specificazione dettagliata di come il livello fornisce il servizio tramite scambio di dati ed informazioni tra le due realizzazioni dei sistemi comunicanti
realizzazione del servizio

Ogni elemento attivo in un livello è una **entità**

Service Access Point

Interfaccia logica tra

una (N-1)-**entity** ed una (N)-**entity** detta

SAP entità **API** le funzionalità disponibili

(N)-SAP (Service Access Point)

un **(N)-address** identifica un insieme di SAP al confine tra il livello (N) ed il livello (N+1)

(N)-SAP address ==> un indirizzo unico che identifica un singolo (N)-SAP tra i più possibili

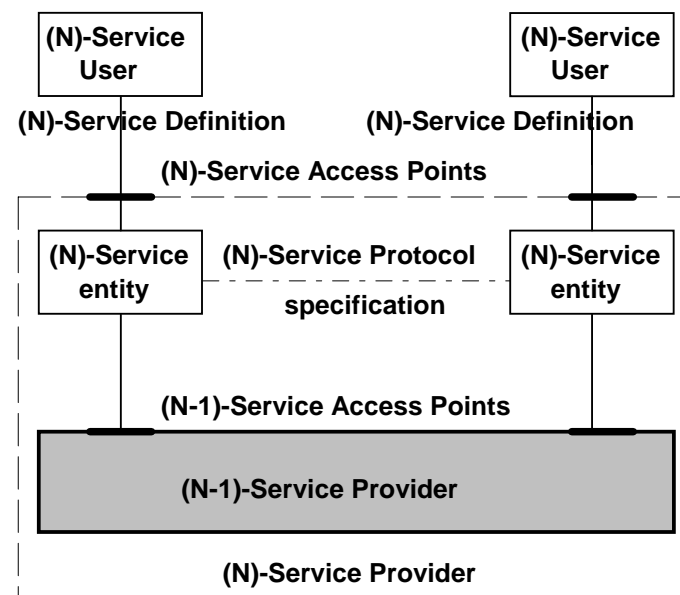
NOMI di pari in OSI

per identificare una **entità completa** si devono specificare tutti i SAP di livello successivo

si devono indicare tutti i SAP della pila

(spesso ci si ferma al livello di *Network*)

(N)-SAP (Service Access Point)



Servizio del generico livello OSI

Ogni **SAP** deve avere un **nome unico** per essere identificata e ritrovata per il suo livello
(concatenazione dei SAP dei veri livelli)

La organizzazione è molto astratta e potrebbe essere applicata alla modellizzazione anche di molti sistemi OSI definisce le sole specifiche di comunicazione nessuna specifica a livello **locale** (procedure/processi/ etc.)

per ogni livello, sono possibili e riconosciute
implementazioni a procedure
implementazioni a processi
ancora più **parallele**

Passaggio Informazioni tra livelli

IDU (Interface Data Unit)

PDU (Protocol Data Unit)

Incapsulamento di informazioni

al passaggio da un livello a quello inferiore

All'interfaccia con il livello sottostante **IDU** contiene la

parte di **specifici di protocollo**

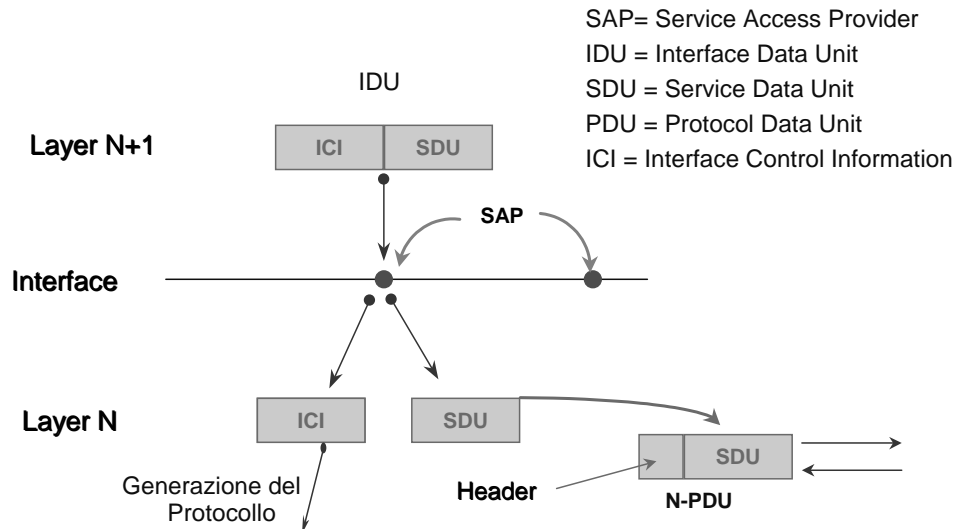
ICI (Interface Control Information) coordina operazioni

PDU di un certo livello costruito aggiungendo al

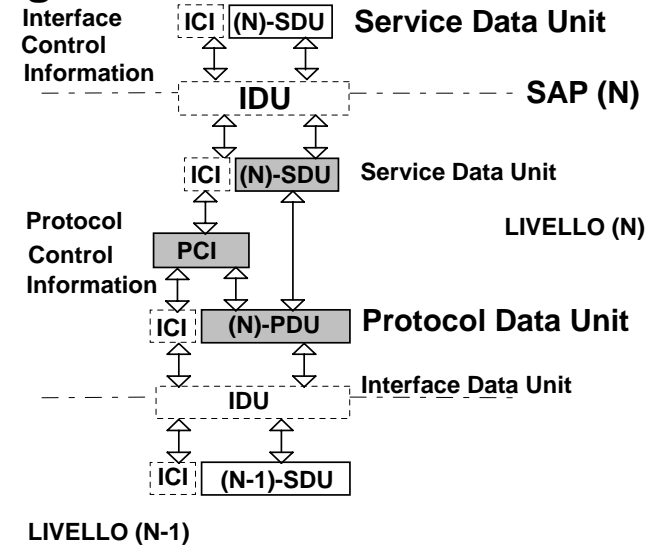
SDU del livello sopra (detta **Service Data Unit**)

informazioni di protocollo

tipicamente attraverso l'uso di servizi sottostanti



Per ogni livello

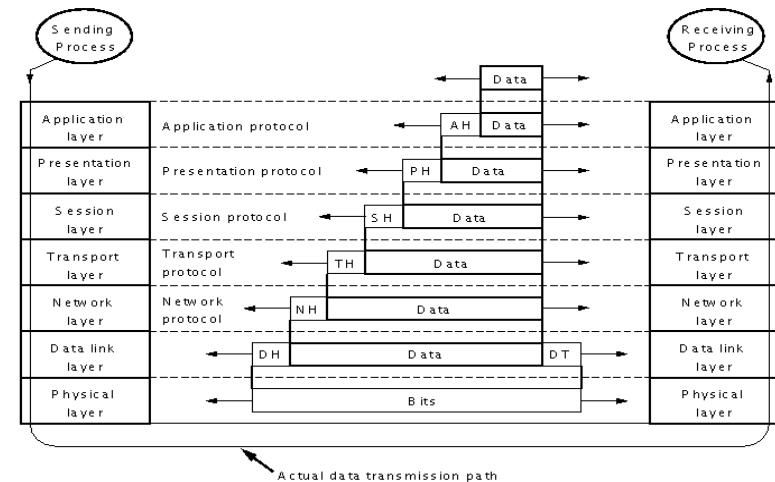


ICI determina il protocollo e diventa PCI

PCI Protocol Control Information

Il PDU viene formato aggiungendo informazioni al dato passato da sopra

Al ricevente azioni complementari



Due modalità

- **CONNECTION ORIENTED.** Si stabilisce una *connessione* tra entità pari che devono comunicare
Le caratteristiche della connessione sono negoziate durante la fase iniziale

In modalità connection oriented la comunicazione tra due utenti di pari livello avviene in tre fasi:

1. **apertura della connessione**
2. **trasferimento di dati sulla connessione**
3. **terminazione della connessione**

Il servizio connection-oriented di un livello deve fornire le opportune funzionalità per le tre fasi

Si considera la **qualità del servizio**

In linea di principio, la connessione non significa impegno di **risorse** su eventuali nodi intermedi necessariamente

CONNESSIONE OSI tipicamente **impegna risorse** sul cammino dal Mittente al Destinatario

- **CONNECTIONLESS.** No negoziazione e valutazione
Ogni unità di dati è trasferita in modo indipendente dalle altre unità ed è autocontenuta per arrivare all'utente desiderato (senza ordinamento)

Lo scambio di informazioni tra i due pari avviene senza storia e senza nessun concetto di negoziazione

Diversa Qualità di SERVIZIO

OSI considera il servizio come caratterizzato da attributi che costituiscono la **Qualità di Servizio (QoS)**

PROTOCOLLI

Le due entità sugli end system devono cooperare tra di loro per implementare le funzionalità del livello cui appartengono

Protocollo è l'interfaccia tra le entità di pari livello

Quattro possibili *tipi per una primitiva*

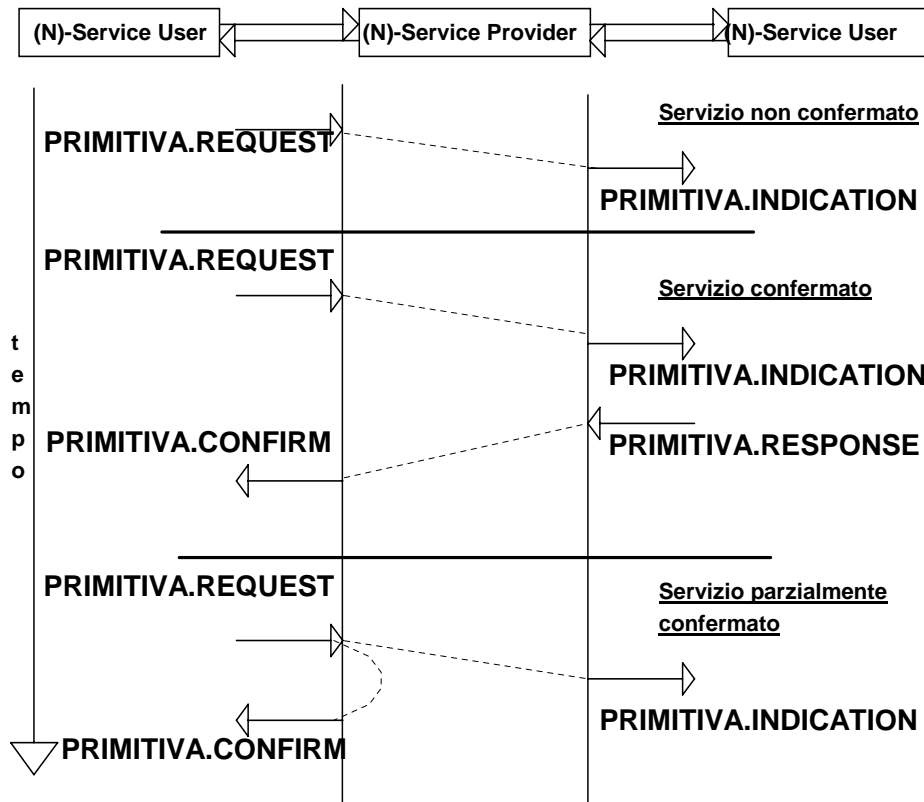
- **Request**
il service user richiede un servizio (una *azione*)
- **Indication**
il service provider indica al service user che è stato richiesto un servizio (segnalazione di *evento*)
- **Response**
il service user specifica la risposta alla richiesta di servizio (una *azione*)
- **Confirm**
il service provider segnala la risposta alla richiesta di servizio (segnalazione di *evento*)

dialogo tra utenti

comporta di utilizzare primitive ai diversi livelli
per ciascuna sono possibili i tipi visti sopra

S-CONNECT . response

Nome primitiva punto Tipo primitiva



Primitiva asincrona (nessun coordinamento)
senza conferma

Primitiva sincrona (coordinamento)
con conferma al mittente (e azione al destinatario)

Primitiva asincrona bloccante
solo conferma al mittente

Si noti l'idea di **evento**, non richiesto dall'utente ma stimolato dalla comunicazione (**indication**)

Scelta della QoS

Ogni servizio deve fare i conti con la qualità logicamente richiesta e le possibilità reali

Esempi di Servizi

affidabilità della comunicazione

garanzie di sequenzializzazione di messaggi e flussi di dati

connessione	affidabile non affidabile
non connessione	affidabile non affidabile

non connessione non affidabile **datagramma**

connessioni affidabili

sequenze di messaggi

flussi di byte (byte stream)

Electronic junk mail **NC** datagrammi (non affidabili)

Mail importante **NC** datagrammi confermati

Database query **NC** protocollo request-response

Voce Digitale **C** connessione non affidabile

Remote login **C** stream affidabile di byte

Immagini animate **C** stream non affidabile

Sequenza di pagine **C** stream affidabile di pagine

C connessione **NC** senza connessione

Se un certa QoS non si può garantire, si può ridimensionare la richiesta

Se la banda è limitata, possiamo anche trasmettere immagini con minore definizione

Monitoraggio di sistema

OSI vs TCP/IP

connessione OSI => con **qualità di servizio - QoS**

connessione TCP/IP => **best effort**

- al meglio, senza troppe garanzie per evitare overhead
- delle condizioni di operazione delle risorse
- dei crash, delle congestioni, ecc.

Livelli inferiori OSI

I livelli inferiori intermedio

Fisico, Data Link, Rete e Trasporto

forniscono un meccanismo trasparente per il trasporto dei dati tra end system

le funzioni base dei quattro livelli includono:

- **controllo degli errori** dovuti al rumore o altra causa
- *controllo di flusso dei dati*
- **modelli di indirizzamento** per identificare end system (servizi di naming)
- **strategie di routing** per trasferire i dati

Internetworking ==> nelle comunicazioni OSI fondamentale l'**indirizzamento**

Esempio **livello fisico** ripetitore
protocollo RS232

Esempio **livello data link** bridge
protocolli
Ethernet, HDLC, PPP

Livello di RETE o Network

Necessità di avere nodi intermedi tra due endsystem

Impossibilità di controllare direttamente il cammino da un qualunque mittente ad un qualunque destinatario

Il livello di network si occupa dei modi diversi di routing tra reti diverse

obiettivo

**passaggio delle informazioni interferendo
meno possibile sul comportamento locale**

Principio di separazione

I nodi intermedi devono potere interagire solo per le funzionalità necessarie e non essere toccati ai livelli applicativi

modalità di network OSI

- **protocolli con connessione** tra end-system (CONS)
X.25 uso di canali virtuali
- **protocolli senza connessione** (CLNS)
- standardizzazione dell'**internetworking**
I protocolli di routing stabiliscono come passare dagli end system ai diversi **intermediate system**

standardizzazione di router (**network**)
gateway (**applicazione**)

N level OSI X.25

Comunicazione tra

DTE Data Terminal Equipment

DCE Data Circuit terminating Equipment
a livello basso X.21 e HDLC e LAPB

X.25 con forme di **connessione sui canali fisici**
permanente o switched (**multiplexing**)

Pacchetti di 128 byte che usano canali virtuali e vengono consegnati in ordine e senza errori

Ogni livello può avere (anche per gli intermedi) compiti di **controllo di errore e ritrasmissione**

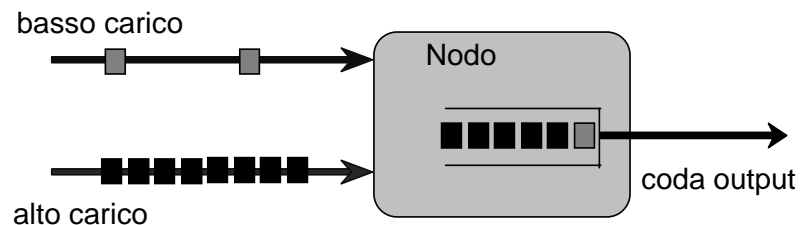
COMPITI

1. Indirizzamento (vedi IP)
2. Controllo di flusso

Controllo di **flusso** tra due pari

Controllo di **congestione** nel sistema intero

Per migliorare efficienza,
evitare ingiustizia, deadlock



Verso il controllo di flusso nel routing

metodi **statici** e **dinamici**

Preallocazione della memoria

- uso di buffer predefiniti
- per circuiti virtuali

scarto di pacchetti

identificata la congestione (localmente)

si scartano con giudizio datagrammi

non si scartano ACK

pacchetti CHOKE

identificata la congestione (localmente)

si inviano messaggi di choke (vedi IP)

finestre scorrevoli

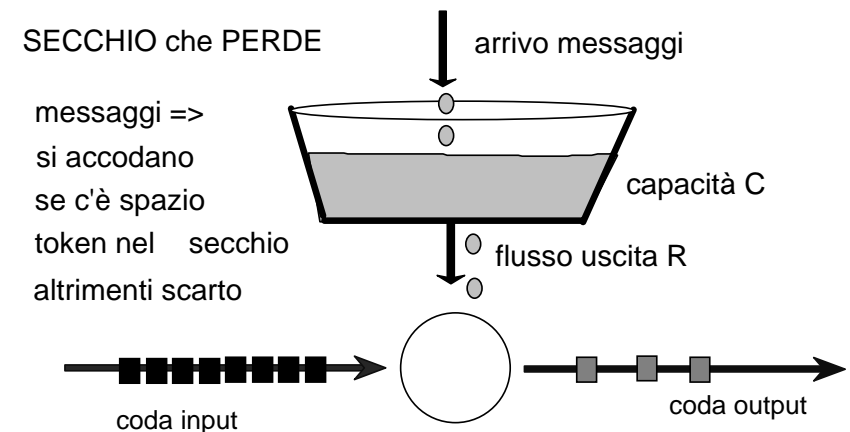
algoritmi leaky bucket

in caso di congestione

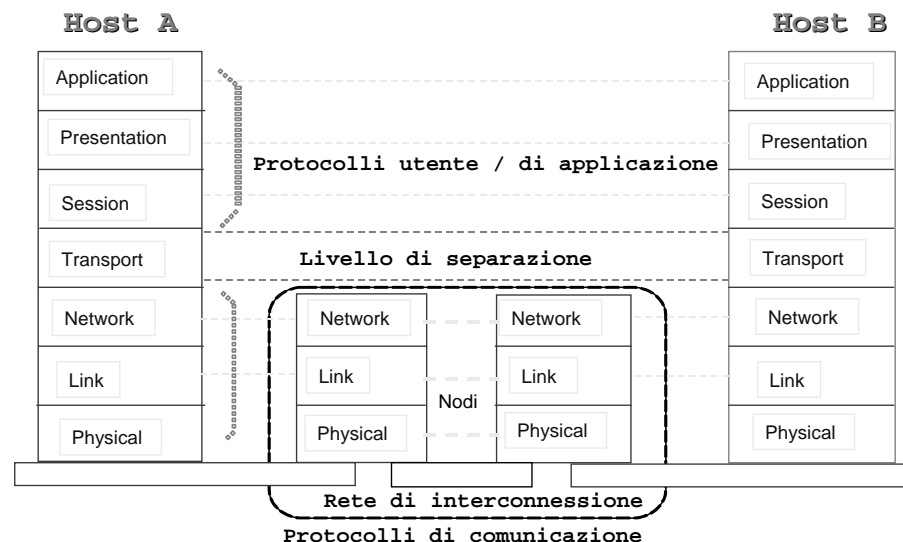
si scartano datagrammi in base allo stato del secchio:

C capacità del router, **R** velocità di smistamento in out

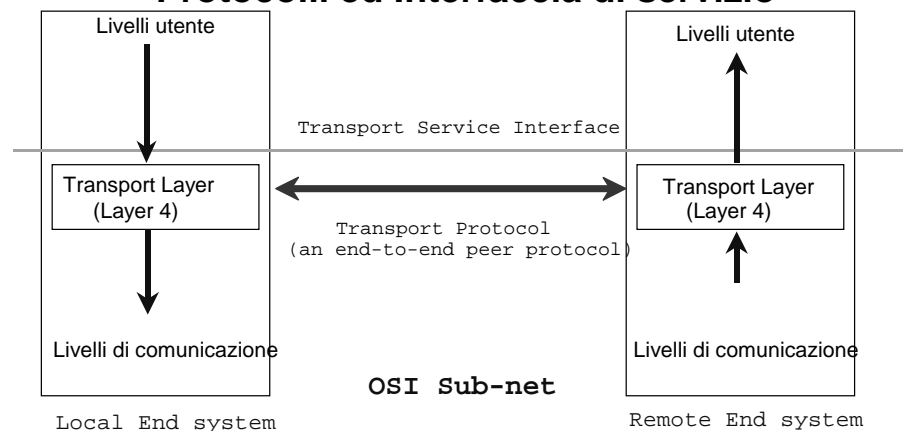
T impegno di risorse del messaggio (token associato)



Comunicazione i livelli fino sotto il trasporto
Applicazione i livelli sopra il trasporto



Protocolli ed Interfaccia di servizio



Livello di Trasporto

Trasporto il livello end-to-end

svincolato dai problemi di connessione fisica dei livelli inferiori

Il livello di trasporto separa i livelli relativi alla comunicazione da quelli più vicini alla applicazione

obiettivo

spedizione di dati sul canale di connessione con correttezza e con certi tempi di risposta con una certa qualità di servizio

indipendentemente da come questo è stato ottenuto

Si trattano le richieste che arrivano dal livello S superiore

modalità connection oriented

- **apertura e terminazione** di una connessione
- trasferimento di **dati normali e privilegiati** (expedited)
 I dati expedited sono soggetti ad un controllo di flusso separato che permette l'invio di messaggi di controllo anche se il servizio per i dati normali è bloccato

In generale, ogni primitiva di servizio per un dato livello prevede un certo insieme di **parametri**

sono possibili modi non connessi

In caso di connessione, OSI impegna risorse intermedie

Qualità di Servizio

Molti indicatori per garantire diversi livelli di qualità del servizio

Ritardo nel set-up di connessione e probabilità di errore

ritardo tra il tempo di richiesta e il tempo di completamento (e tempo massimo accettato)

Throughput

numero di byte trasferiti per secondo

Tempo di ritardo

ritardo tra l'istante di invio e la possibile disponibilità per il pari (massimo ritardo ammissibile detto **jitter**)

(Probabilità o Rapporto di) Errore residuo

rapporto tra il numero di messaggi non corretti e quelli buoni

Priorità

importanza relativa di alcuni messaggi

Errore del trasporto (resilience)

probabilità della terminazione della connessione

La fase in cui si stabilisce una connessione implica la negoziazione tra i pari per la qualità richiesta della connessione.

È possibile che il pari (o uno dei due) non possa fornire quanto richiesto, e allora

o si adegua la **QoS richiesta**

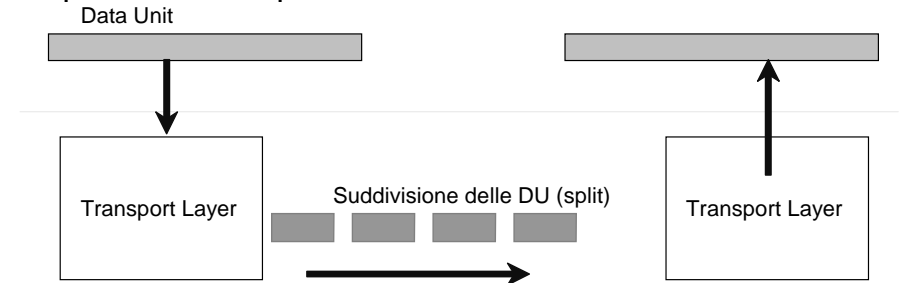
o **non** si fa la connessione

Offerta accettata o rifiutata

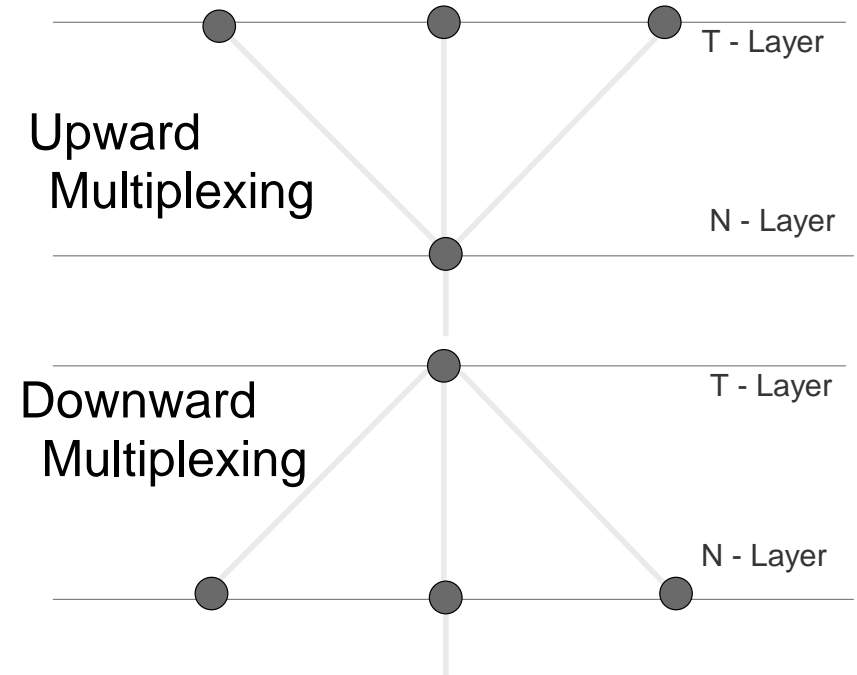
NON negoziazione vera e propria (?)

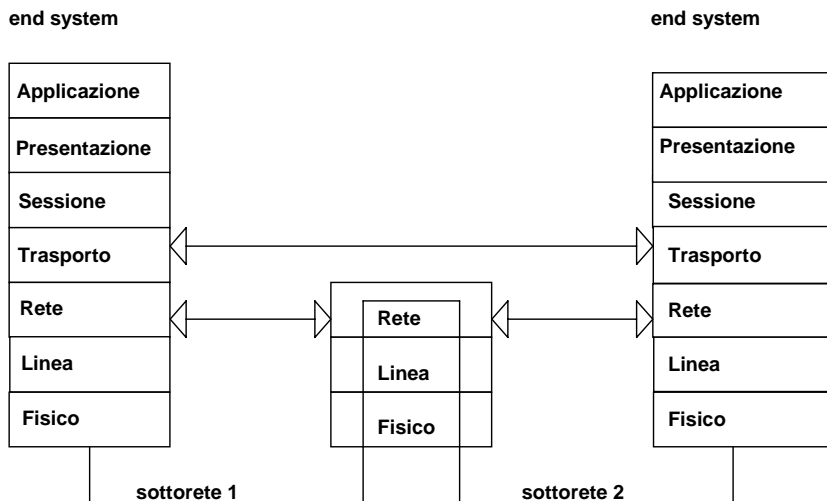
Livello T - Funzioni possibili

Il trasporto può **spezzare** il dato e **ricomporlo** dopo averlo trasportato fino al pari suddiviso



Il trasporto può lavorare **unendo** o **decomponendo** flussi di trasporto rispetto a quelli di rete (**Multiplexing**)





Intermediate System

Livello di Trasporto agisce su base end-to-end

<u>primitiva</u>	<u>tipo di servizio</u>	<u>parametri di servizio</u>
T-CONNECT	servizio confermato	indirizzo del chiamante e del chiamato, opzione per l'uso di dati privilegiati, qualità di servizio e dati d'utente.
T-DATA	servizio non confermato	dati di utente
T-EXPEDITED-DATA	servizio non confermato	dati di utente
T-DISCONNECT	servizio non confermato	ragione della terminazione, dati d'utente

Primitive del servizio di Trasporto

Livelli superiori OSI

I livelli superiori o *livelli applicativi* forniscono servizi più astratti di quelli dei livelli sottostanti
servizi connection oriented

Necessità di supporto al dialogo

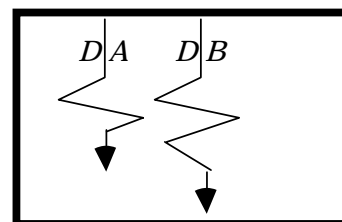
CONSIDERANDO IL DIALOGO TRA ENTITÀ

bisogna tenere in conto le possibilità tra due pari che comunicano

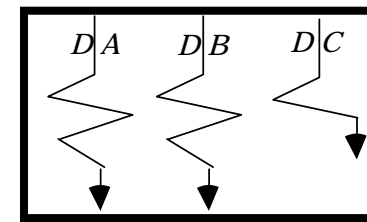
Con il trasporto arriviamo da nodo a nodo dalla sessione

la interazione può

- essere **bidirezionale**
- considerare le **risorse** impegnate
- deve avere garanzie di **correttezza** e **affidabilità**
- essere strutturata a diverse **attività**



Applic A



Applic B

Ogni pari struttura con ricchezza espressiva la interazione

Livello di Sessione

Il livello di Sessione coordina il dialogo tra gli utenti

basandosi sul servizio offerto dal livello di Trasporto

Servizio di Sessione

58 primitive e 13 unità funzionali

Il livello di Sessione offre servizi analoghi a quelli del livello di Trasporto, quindi:

- **apertura della connessione e sua terminazione**
- **trasferimento dati** (si possono avere fino a quattro tipi di dato)

Servizi **aggiunti e specializzati** per:

- **gestione dell'interazione**
modalità di dialogo half-duplex, full-duplex o simplex
- **sincronizzazione**
inserimento dei punti di sincronizzazione (**checkpoint**)
La successione dei punti di sincronizzazione individua una successione di *stati della comunicazione*

*Trasmissione di un file da due ore bloccata dopo un'ora
Si riprende dal risultato del trasferimento precedente*

Ogni livello superiore è partizionato in *unità funzionali*, ognuna legata ad un insieme di primitive e parametri

Il numero delle unità funzionali

- cresce per i livelli servo l'applicazione
- è negoziato tra i due pari dei diversi livelli

CONTROLLO del DIALOGO

Possibilità di intervenire sul dialogo tra pari

Se si verificano errori nella comunicazione => **roll-back**

*Si pensi al trasferimento di informazioni di molti MByte ,
se crash, si ricomincia (?)*

Due tipi di **punti di sincronizzazione maggiori e minori**

si può ritornare ad uno stato definito e concordato dalle due entità di Presentazione (cioè gli SS-user) tramite un'analisi dei punti di sincronizzazione determinati

1) punti di sincronizzazione maggiore

necessaria la segnalazione alla ricezione

(garanzia di ricezione ma tempo di attesa)

Chi ha inviato tale punto aspetta in modo sincrono bloccante la conferma del punto stesso dall'altro utente

1) punti di sincronizzazione minore

punto di sincronizzazione minore con primitiva

S-SYNC-MINOR.REQUEST (**conferma** o meno)

L'utente alla ricezione di un punto minore non è obbligato a segnalare al pari (anche se la conferma è richiesta)

Si può continuare a spedire dati o punti di sincronizzazione anche senza conferma

Con la conferma di un punto di sincronizzazione minore, si confermano anche tutti i punti precedenti

controllo dialogo

Tipicamente, si può negoziare il **numero di punti** di sincronizzazione minore che possono rimanere in attesa di conferma determinando dimensione della **finestra di scorrimento** (*sliding window*)

La risincronizzazione verso uno stato definito della comunicazione può comportare

In caso di RECOVERY

- 1) **abbandono**: reset della comunicazione corrente
L'utente può decidere di provare a ripeterla
- 2) **ripristino**: la comunicazione è riportata nello stato precedente l'ultimo punto di sincronizzazione maggiore confermato
Si possono ritrasmettere i dati successivi
- 3) **diretto**: la comunicazione è riportata in uno stato arbitrario senza controllo delle conferme "mancanti" di punti di sincronizzazione

dialogo

Si possono anche controllare

- **attività**: è possibile iniziare, terminare, interrompere, riprendere e cancellare parti di lavoro dette *attività*
- **eccezioni**: è possibile notificare eccezioni al servizio corrispondente

Connessione con QoS negoziata

le connessioni a questo livello hanno una semantica sempre più complessa

Si attuano negoziazioni attraverso la presenza di package di servizi detti **Unità funzionali**
Quelle basi sempre presenti le altre negoziate

Unità Funzionali

Gruppi di funzioni di Sessione in sottoinsiemi:

- **Kernel**: composto dalla sola unità funzionale Kernel
- **Basic Combined**: composto dal sottoinsieme Kernel e dall'unità funzionale Half Duplex o Duplex
- **Basic Synchronized**: composto dal sottoinsieme Basic Combined e dalle unità funzionali Minor, Major Synchronize, Typed Data, Negotiated Release e Resynchronize
- **Basic Activity**: composto da Basic Synchronized (modalità half-duplex e punti di sincronizzazione minore) e dalle unità Activity Management ed Exception Report

L'utente seleziona l'unità funzionale più adatta alle proprie esigenze

Unità funzionali di Sessione

Kernel	Supporta i servizi di base per lo stabilimento di una connessione di Sessione, il trasferimento di dati e la terminazione di una connessione.
Negotiated Release	Permette una negoziazione della terminazione della connessione.
Half Duplex	Permette il controllo del diritto a spedire i dati utilizzando il data token.
Duplex	Permette ad ambedue gli SS-users di spedire dati, non vi sono token per regolare tale diritto.
Expedited Data	Permette l'invio di dati privilegiati se è disponibile il corrispondente canale di Trasporto.
Typed Data	Permette l'invio di dati anche quando non è il turno dell'utente che li vuole spedire. Viene ignorato il data token.
Minor Synchronize	Nel flusso dei dati vengono inseriti dei numeri seriali che servono da punti di (minore) sincronizzazione. Tali punti costituiscono il modo con cui i provider tengono traccia della divisione (logica) del flusso di dati a livello di Sessione. Due utenti sono sincronizzati se l'ultimo punto di sincronizzazione spedito da uno di essi è anche l'ultimo punto di sincronizzazione ricevuto dall'altro utente.
Major Synchronize	Ancora vengono inseriti dei numeri seriali nel flusso di dati come punti di (maggiore) sincronizzazione. L'utente che riceve un punto di sincronizzazione deve indicare questo fatto all'altro utente che nel frattempo non può intraprendere alcuna altra azione (deve attendere un confirmed major synchronize point).
Symmetric Synchronize	In modalità di comunicazione full duplex viene inserito un numero seriale in ognuno dei due possibili flussi di dati.
Resynchronize	Permette agli SS-users di negoziare il recupero della comunicazione verso uno stato concordato dopo un errore od una mancanza di risposta da parte del provider o di un utente. I numeri di sincronizzazione marcano i diversi stati raggiunti durante una comunicazione e possono essere sia di maggiore che di minore sincronizzazione.
Exceptions	Permette di segnalare ad un utente errori rilevati dal provider o dallo altro utente senza dover necessariamente terminare la connessione.
Activity Management	I punti di sincronizzazione dividono un flusso di dati in unità di dialogo. Una attività è costituita da una o più unità di dialogo tra loro logicamente correlate. Queste unità permettono la gestione e la risincronizzazione delle attività ed in particolare per una attività sono previste primitive indicanti l'inizio, la fine, la sospensione temporanea e la cancellazione dell'attività stessa. È utilizzabile solo in modalità di dialogo half-duplex.
Capability Data	Permette, se selezionata l'unità precedente, l'invio di dati anche se nessuna attività è in corso.

Controllo Dialogo - protocollo

Strutturazione e sincronizzazione del dialogo attraverso **oggetti astratti detti *token***

un solo utente possiede il **token** in ogni momento ed ha il diritto di uso di un insieme di servizi di Sessione

La primitiva S-CONNECT che stabilisce la connessione tra due SS-user consente di negoziare anche i token

Ad esempio, *l'utente che richiede la connessione e l'utente che la accetta indicano le unità funzionali desiderate*

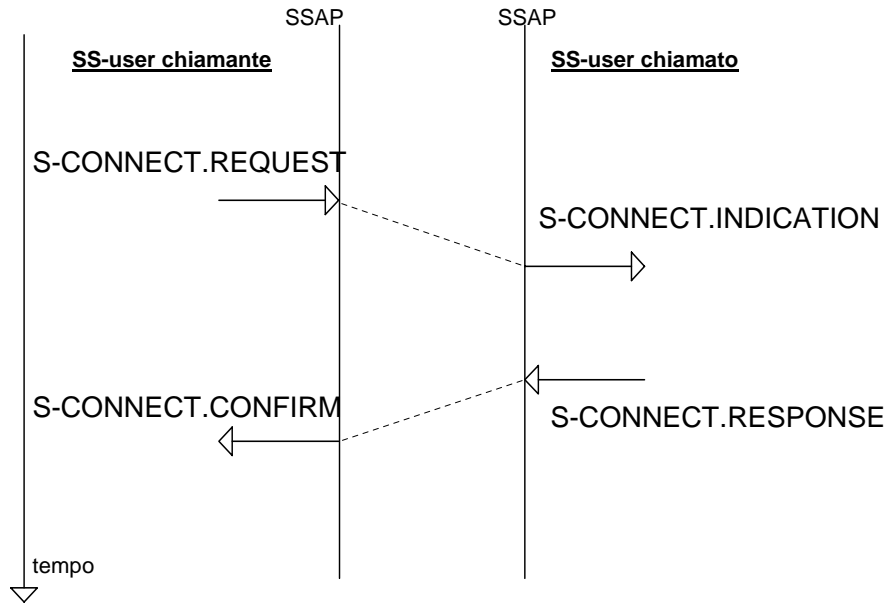
L'intersezione dei due insiemi di unità funzionali determina le unità funzionali attive nella connessione di Sessione

unità funzionale Kernel ==> sempre attiva
indipendentemente dalle decisioni dei due utenti

A secondo delle unità funzionali selezionate

tipi di token distinti:

- **data token:** per spedire i dati con l'**uf Half Duplex**
- **release token:** si può richiedere la terminazione della connessione con l'**uf Negotiated Release**
- **synchronize minor token:** chi lo possiede può creare *punto di sincronizzazione minore* con l'**uf Minor Synchronize**
- **synchronize major token:** chi lo possiede può creare punti maggiori con l'**uf Major Synchronization** o l'unità Activity Management



parametri di S-CONNECT.REQUEST

- identificatore di connessione di Sessione
- indirizzi del chiamante e del chiamato
- qualità proposta per il servizio
- unità funzionali proposte dal chiamante
- numero seriale iniziale per i punti di sincronizzazione; obbligatorio solo se si utilizza l'unità Minor o Major Synchronize
- proprietà iniziale dei tokens disponibili con le unità funzionali proposte
- dati di utente (opzionali)

parametri di S-CONNECT.RESPONSE

- identificatore di connessione di Sessione
- indirizzo del chiamato
- qualità di servizio
- unità funzionali proposte dal chiamato
- numero seriale iniziale
- proprietà iniziale dei tokens; il chiamato deve accettare le scelte del chiamante
- dati di utente
- ragione dell'eventuale rifiuto o accettazione della connessione

Apertura di una connessione di Sessione

Livello di Presentazione

La **codifica** delle informazioni **non univoca**

Ogni **pari** può usare **codifiche diverse**

Il livello di Presentazione offre

tutti i servizi offerti dal livello di Sessione

per trasformare la codifica dei dati ricevuti dai vicini

NECESSITÀ di codifiche diverse:

- differenze naturali tra i sistemi che comunicano
- migliorare la comunicazione (**efficienza e sicurezza**)
- uso di compressione dei dati (**efficienza**)
- crittografia dei dati (**sicurezza**)

I dati devono essere scambiati dopo un accordo tra i pari che spera gli eventuali problemi di eterogeneità

linguaggi di programmazione diversi **C e ADA**

sistemi operativi diversi

UNIX e VMS

architetture diverse

ALPHA e RISC

Se non ci sono problemi non si fanno trasformazioni
Altrimenti bisogna progettare correttamente

Progetto generale:

caso senza conoscenze pregresse

Necessità di accordarsi. Bisogna definire un

contesto di comunicazione

- il soggetto della comunicazione
- la semantica delle informazioni
- le informazioni vere e proprie

Il livello di presentazione stabilisce come

negoziare e definire una base comune

Negoziare a molte fasi (durata?)

Livello di Presentazione

Il livello ha responsabilità di

- distinguere informazioni in forma **astratta** e **concreta**
- specificare i dati e le informazioni di **controllo** in modo indipendente dalla forma concreta
- definire una **forma comune** per le informazioni **astratte** (anche diversa per connessioni diverse)
- definire una **forma comune di rappresentazione concreta** dei dati (non coincidente con una delle due coinvolte e richiedendo trasformazioni)

Si definisce un **linguaggio astratto** di specifica
(**ASN.1 Abstract Syntax Notation**)
e una **codifica concreta** di descrizione dei dati
(**BER Basic Encoding Rules**)

Se la **negoziazione iniziale** di contesto (ASN.1)
può fornire un accordo completo ==>
I dati vengono trasformati in accordo a questo
Si usa un formato standard **BER anche efficiente**

Se la **negoziazione** non lo consente ==>
I dati viaggiano insieme alla descrizione standard

☹ OVERHEAD ELEVATO

☺ ESTREMA FLESSIBILITÀ

Infatti ASN.1 consente non solo di passare dati non previsti inizialmente, ma anche codice e di creare accordo durante la comunicazione stessa
(vedi i limiti di linguaggi IDL come XDR, etc.)

Possibilità di passare anche codice

Anche con **ridondanze (reliability e security)**

Esempio di ASN.1 e BER

ASN.1

Primitive types

BOOLEAN
INTEGER
OCTETSTRING
IA5String

```
Address ::= SEQUENCE {  
    addr_src IA5String,  
    addr_dst IA5String  
}
```

Constructor types

SEQUENCE
SEQUENCE OF
SET
SET OF
CHOICE

```
Pdu ::= SEQUENCE {  
    pdu_ad Address,  
    pdu_len INTEGER,  
    pdu_data OCTETSTRING (SIZE 1024)  
}
```

BER

Triple Tag-Length-Value:

codifica a discesa ricorsiva

```
address.source = "Suna"  
address.destination = "Decb"  
length = 3  
data = 'x', 'y', 'z'
```



Primitive types BER

BOOLEAN	
INTEGER	02
OCTETSTRING	04
IA5String	16

Constructor types

SEQUENCE	30
SEQUENCE OF	
SET	
SET OF	
CHOICE	

```
30 22  
 30 12  
    16 04 'S' 'u' 'n' 'a'  
    16 04 'D' 'e' 'c' 'b'  
 02 01 03  
 04 03 01 02 03
```

In molti casi, si possono risparmiare informazioni

Livello di Applicazione

Il livello di Applicazione è il livello che si interfaccia con l'utente finale della comunicazione in base al modello OSI

Obiettivo **astrazione**

nascondere la complessità dei livelli sottostanti coordinando il dialogo tra le applicazioni distribuite.

Il livello applicativo OSI standard definisce un **insieme di servizi indipendenti dal sistema** e li fornisce a *programmi di utente o ad utenti*

Diversi standard (ISO 9545):

Message Handling System	MHS
Directory service	X.500
System Management	X.700
Common Management Information Service	CMISE
Protocol	CMIP
File Transfer, Access and Management	FTAM
Virtual Terminal Standard	VT
Distributed Transaction Processing	DTP

OSI adotta un approccio particolare basato sul modello ad *Oggetti* per la specifica delle applicazioni

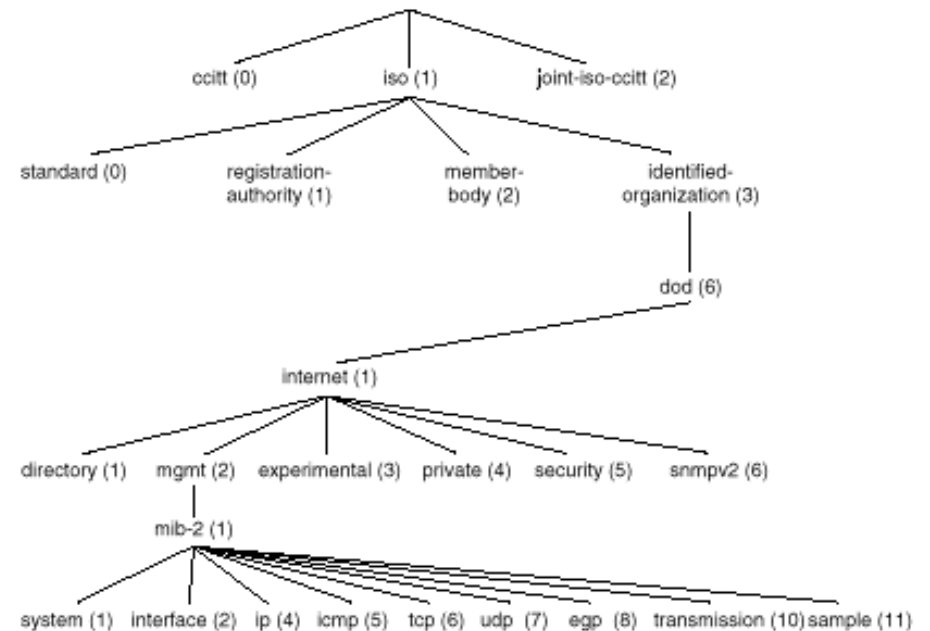
Servizi X.500

Il **servizio di direttorio** consente di collocare e classificare ogni dispositivo noto in un **sistemadi nomi** gerarchico

Directory per tutti gli apparati tipicamente usati

Nomi unici

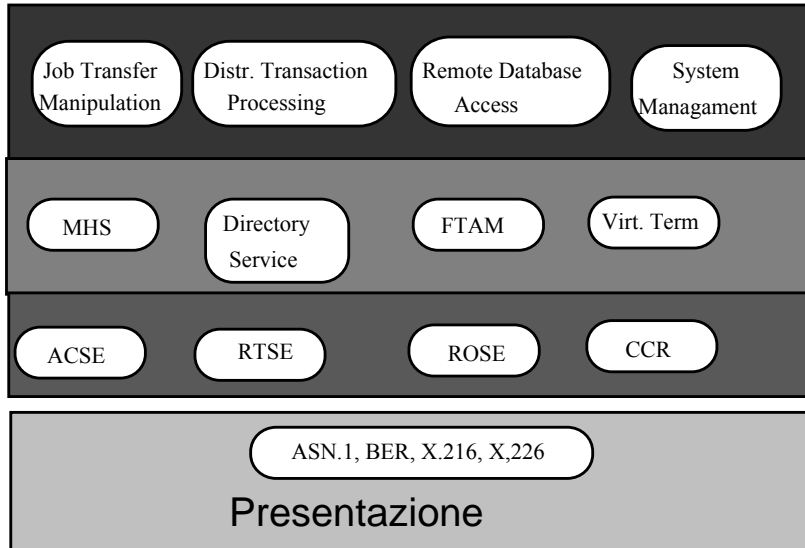
Spesso si riferisce una entità attraverso il suo identificatore specificato come sequenza di scelte



Negli altri casi, si negoziano le proprietà con ASN.1

APPLICAZIONE COME INSIEME DI LIVELLI E DI STRUMENTI

Applicazione



Alcuni strumenti sono a livello di base rispetto agli altri

ACSE (Association Control Service Element) di base per ogni servizio

RTSE (Reliable Transfer Service Element) per ottenere servizi affidabili

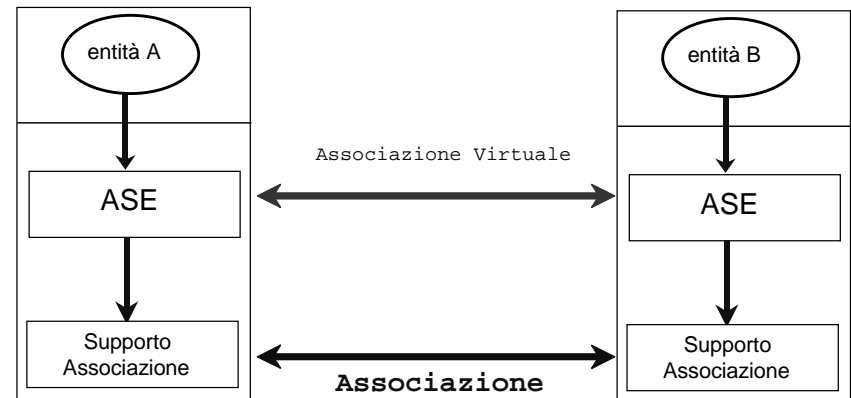
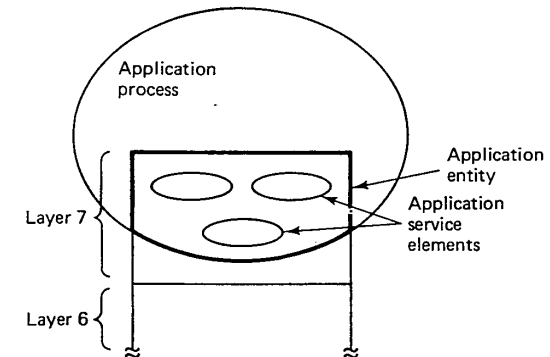
ROSE (Remote Operation Service Element) per ottenere operazioni remote

CCR (Commitment Concurrency and Recovery) protocollo per garantire azioni coordinate

APPLICAZIONE

Si devono considerare sempre connessioni a livello applicativo per ogni singolo servizio

uso di APPLICATION SERVICE ELEMENT



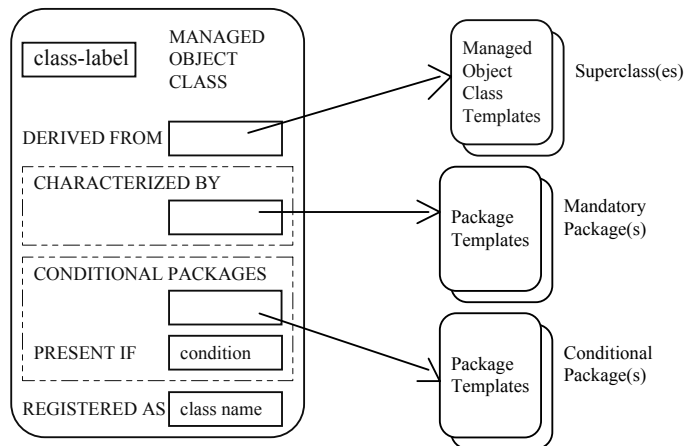
OSI - Descrizione ed Implementazione

Uso di template e package per definire gli oggetti

Pura **ereditarietà statica** tra astrazioni

Oggetti da manipolare come interfaccia
ed espressi attraverso l'uso di **package**
(anche condizionali)

Si noti la **unicità dei nomi** come presupposto di base
NOMI UNICI come servizio (X.500)



SISTEMI DI GESTIONE

OSI introduce la possibilità di

- gestire **risorse remote**
- tollerare **fallimenti** di risorse

Systems Management
Network Management

exec (OSI)
comm (TMN TINA)

Integrazione del **management di rete**
nel **sistema distribuito**

necessità di investire per ottenere il massimo beneficio
da sistemi con reti diverse, di venditori diversi e con
utenti diversi

Management per
controllare, coordinare, monitorare
l'intero sistema distribuito

per esempio, controllare equipaggiamento remoto,
monitorare i parametri di esercizio, ricevere allarmi e
comandare azioni

Management Standard OSI
Modello di **network management standard**
basato su **oggetti astratti**

Mappaggio da oggetti astratti a concreti

non è standardizzato

ad es. Le interfacce utente sono **non standard**

Uso di descrizione oggetti e standard azioni
Common Management Information Base

Distributed Management

OSI management process (MIS)

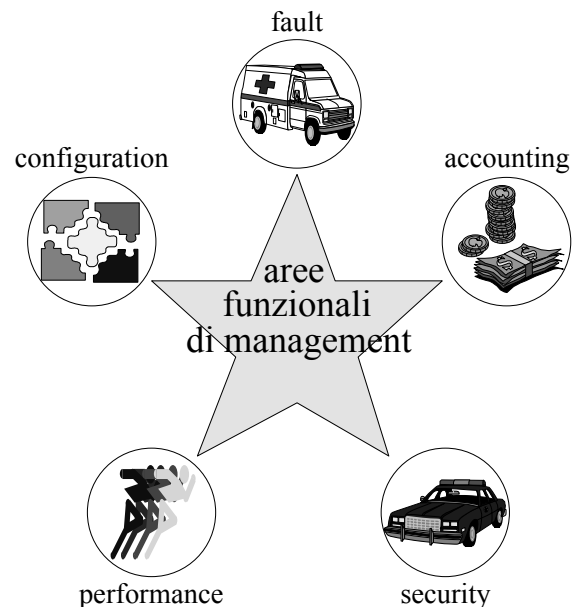
Management Information Service

Management unico delle informazioni (CMISE)

Common Management Information Service Element

Aree funzionali di Management

Fault	Management
Configuration	Management
Accounting	Management
Performance	Management
Security	Management



Standard de facto

OSI più sofisticato

si applica a qualunque sistema distribuito per la gestione distribuita del sistema

SNMP Simple Network Management Protocol

definizione di un semplice protocollo di management

con uso di **TCP/IP**

applicato ad ambienti **UNIX** e **LAN**

SNMP opera su un sottoinsieme di CMIP

incompatibile con lo standard CMIP

NECESSITÀ per SNMP

di ridefinizione e

di reingegnerizzazione

per tenere conto di esigenze di **sicurezza**

per tenere conto di modelli di **gestione flessibili**

per tenere conto di sistemi **legacy esistenti**

...

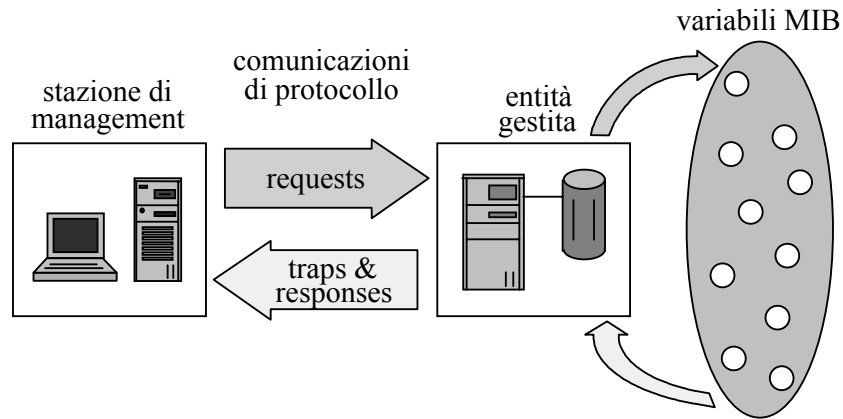
Inoltre, anche potere gestire non solo apparati,

ma **entità di qualunque tipo**

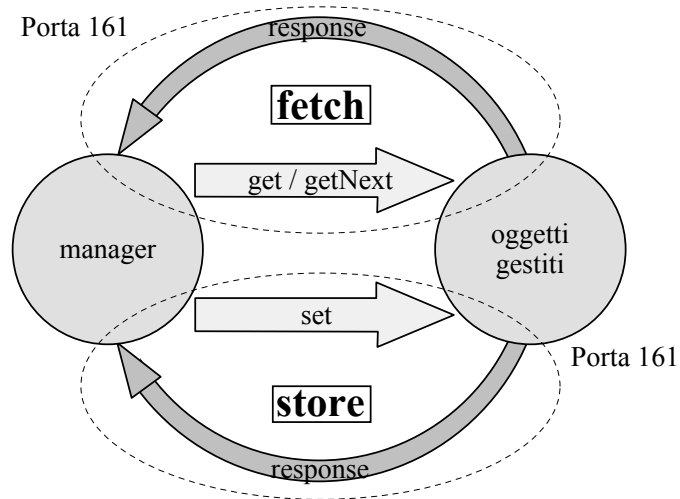
Protocollo SNMP

(Simple Network Management Protocol)

Estrema semplicità: solo alcune azioni elementari



Uso di messaggi molto semplici

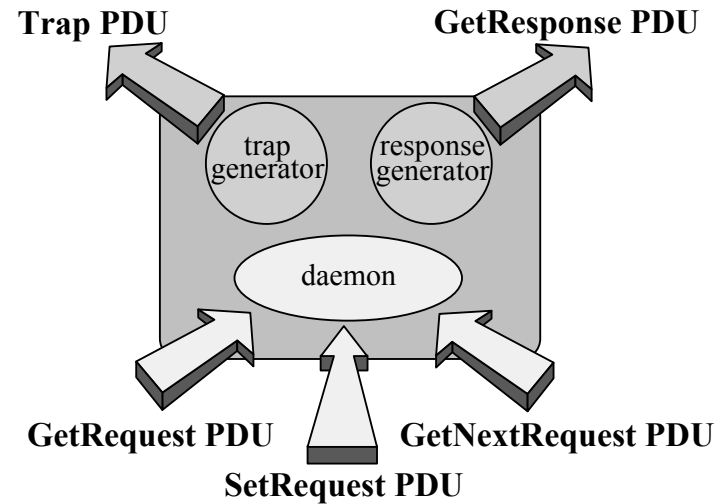


messaggi base: **Set, Get, Get_Next** (attributi multipli), **Trap** (porta 162 del manager)

Protocollo SNMP

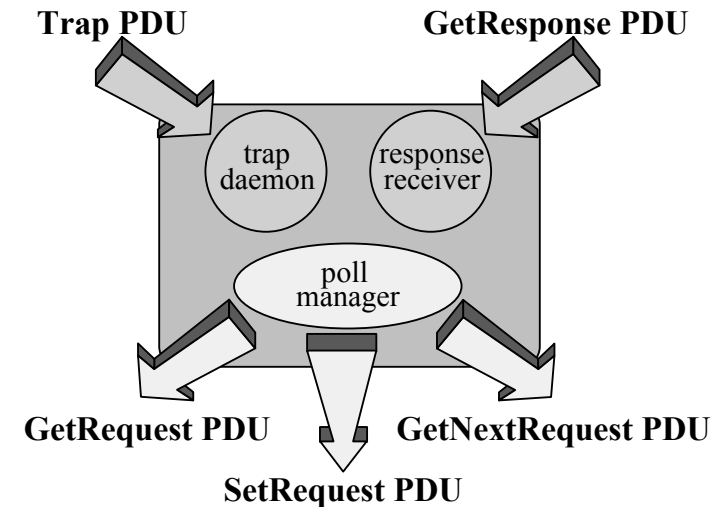
Struttura di un agente SNMP

eventi, trap, richieste di azioni



Struttura di un manager SNMP

eventi, comando di azioni, risposte



Problemi di SNMP

SNMPv1

Estrema semplicità Limitata espressività
Solo aree di **configuration** management (**fault**)
limitatprevisione dei trap (azioni iniziate dall'oggetto)

SNMPv2

Superamento del C/S
anche gerarchia di manager agent

SNMPv3

Introduzione della sicurezza **S-SNMP**

si trattano i problemi di

- integrità delle informazioni (anche stream)
 - masquerading
 - privacy (prevenire disclosure)
- non si trattano
- denial of service
 - analisi del traffico

E il traffico di rete? Remote **MON**itor

Introdotta le parti di supporto alla comunicazione
ed alle statistiche relative

RMON per aumentare la visibilità dell'utente sul traffico
come facciamo a monitorare la rete?

Introduzione di monitor e del protocollo di interazione tra
manager e monitor

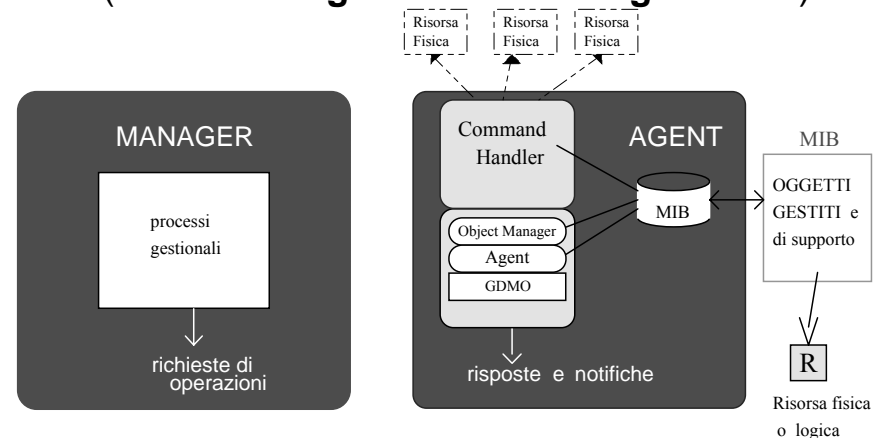
RMON1 sviluppi nel senso della azioni multiple e innestate

RMON2 e nel senso della garanzia di sicurezza

Modello Distributed Management

Modello basato su

entità attive (manager)
entità da controllare (oggetti)
entità intermedie (agenti)
(anche manager a loro volta in gerarchia)



Management Information Base (MIB)

Manager gestori realizzano le politiche di gestione sulla base di più agenti

Un manager *inserisce una risorsa* o la *toglie* dal sistema

Agenti i gestori usano **agenti** per ottenere le funzioni
servizi di *attuazione comandi*, *raccolta informazioni*
gli agenti possono anche essere manager

Managed Objects

risorse descritte in termini di oggetti

Un oggetto astrae una o più risorse nel sistema

risorse semplici, *un modem*,
o complesse, *più sistemi interconnessi*

Protocollo

La comunicazione tra il manager e gli agenti del management è standardizzata

**Common Management Information ServiceElement
CMISE**

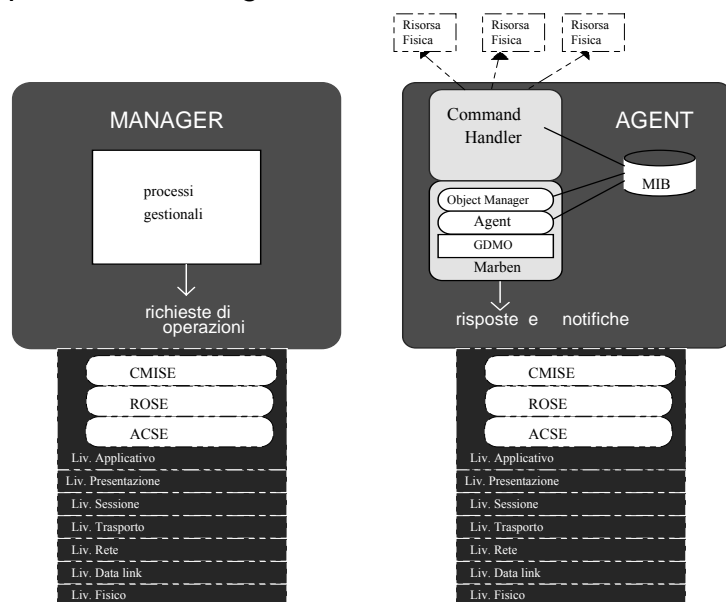
**Common Management Information Protocol
CMIP**

Si agisce con un linguaggio comune nel sistema distribuito

**Common Management Information
servizio di OSI per lo scambio reliable di informazioni**

Management Information Base (MIB)

MIB definisce un *modello concettuale dei dati*, parametri ed eventi del sistema da controllare e definisce le proprietà del management



Management Communication

servizi e protocolli contenuti in CMISE e CMIP
Uso di servizi applicativi ROSE, ACSE etc.

Management entity usano il protocollo **CMISE/P**

Insieme di operazioni remote per la comunicazione tra manager ed agenti

Set-Modify *stabilire, aggiungere o togliere un attributo ad un oggetto*

Get / Cancel Get *lettura di un attributo ad un oggetto (e revoca lettura)*

Action *azione su uno o più oggetti*

Create/ Delete *richiesta di una generazione/ distruzione ad un agente*

Event Report *invio di un evento notificato dall'agente al manager*
Si noti la aggiunta di attributi, azioni, agenti, e eventi

Livelli tipici di servizi e protocolli

GDMO	Fornisce la definizione object-oriented del modello informativo degli apparati; è quindi un supporto statico. In particolare ricordiamo che secondo la metodologia sviluppata dall'OSI-Management un managed object (MO) risulta definito da: Attributi, Operazioni, Notifiche e Comportamento.
CMISE	Supporto dinamico che specializza la RPC fornita da ROSE, al fine di gestire la rete, definendo i prototipi delle operazioni possibili sugli oggetti: ♦ creazione e distruzione, ♦ definizione e ridefinizione dei valori degli attributi , ♦ richiesta di operazioni , ♦ ricezione di rapporti dagli oggetti.
ROSE	Realizza la chiamata di procedura remota (RPC), ed è indispensabile per lo sviluppo del Software distribuito di gestione della rete.

**Guidelines for the Definition of Managed Object
Remote Operation Service Element**

Confronto OSI e TCP/IP

Aldilà del numero dei livelli

completezza di OSI

limiti TCP/IP

uso di **Object-Orientation** descrizione approssimata
negli RFC

interfaccia
implementazione

protocolli

progetto ben fatto

implementazione

interesse in standard

prodotti

banco di apertura

aggiunte di proprietà

Qualità di servizio
Connessione

IETF sta cominciando a definire nuove evoluzioni per fornire servizi di gestione adeguati alle nuove applicazioni e servizi

- di tipo **flussi real-time**

- di tipo **globale e commerciale integrati**

Differentiated services (DIFFSERV)

Integrated services (INTSERV)