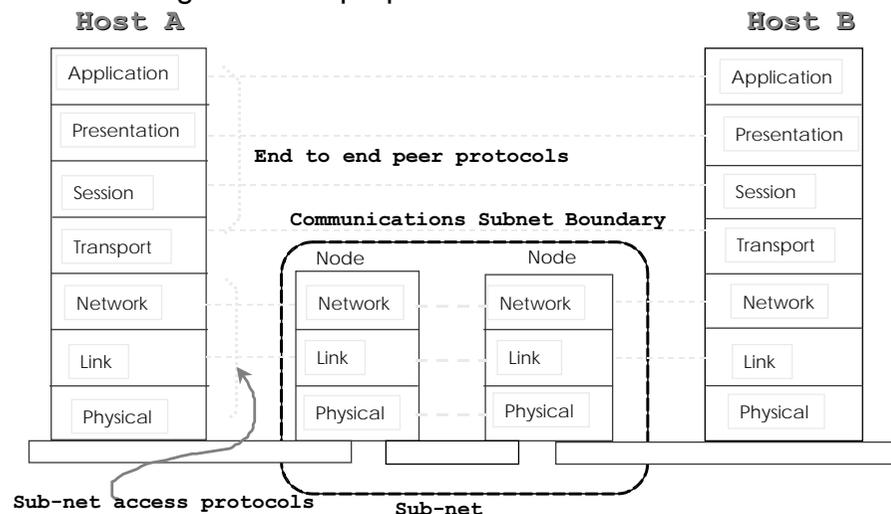


STANDARD di comunicazione

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**

Obbiettivi di INTEROPERABILITÀ

potere effettivamente comunicare ed operare tra sistemi eterogenei

OSI Gestione dei sistemi

(*Systems management Network management*)

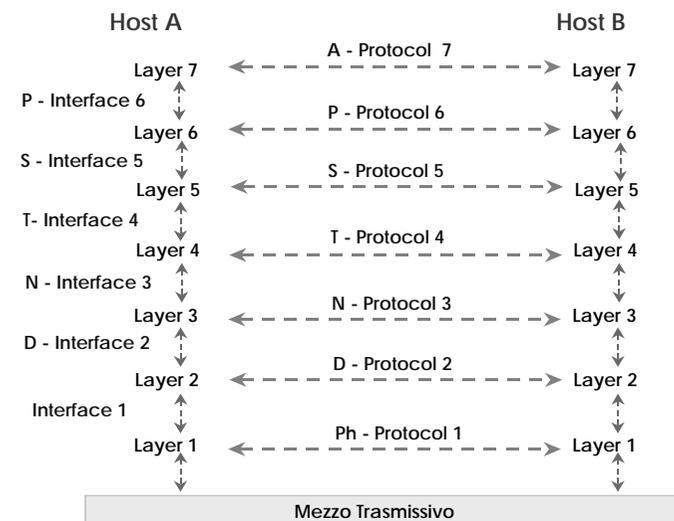
controllare, coordinare, monitorare sistemi interconnessi

VANTAGGI DEGLI STANDARD

Possibilità di ottenere **comunicazione in sistemi aperti**

Sistemi a livelli

Ogni livello ha protocollo ed interfaccia propri

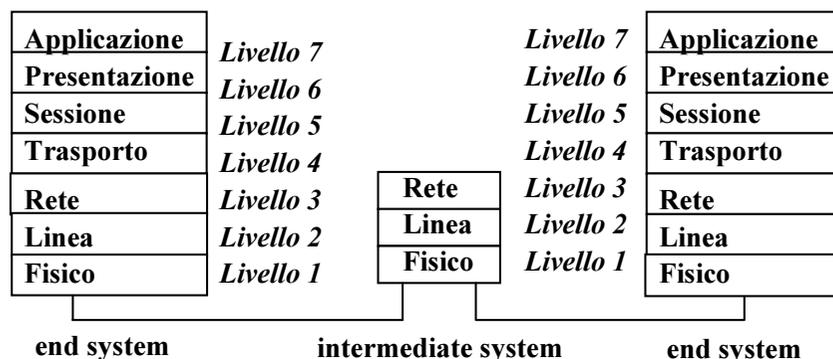


Il livello ha l'obiettivo di **comunicare** con il pari e realizza l'obiettivo tramite il protocollo, usando il servizio sottostanti. Ogni livello fornisce un **servizio specificato** al livello superiore.

OSI CARATTERISTICHE

OPEN SYSTEM INTERCONNECTION

sistema a livelli



livelli OSI

Si considerano **macchine host** e anche nodi intermedi di **rete**

NOMI DEI LIVELLI

lettera iniziale maiuscola **(N)-layer**

(N-1)-layer

(N+1)-layer

(N)-protocol

S-protocol per il livello di Sessione

definizione di un **servizio** *semantica del servizio*

interfaccia del servizio usata dal livello sovrastante

implementazione *specifica del protocollo*

protocollo realizzato dal livello e non visibile

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
- **specifiche del protocollo** (descrizione orizzontale)
specifica 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

(N)-SAP (Service Access Point)

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

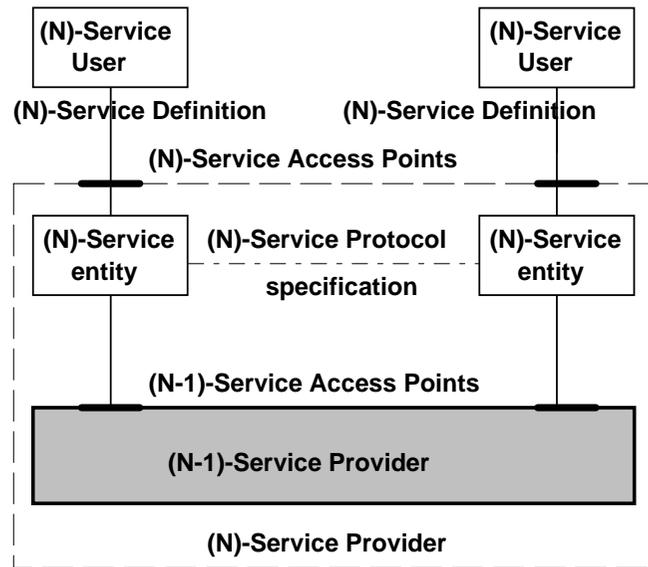
(N)-SAP address ==> un indirizzo unico che identifica un singolo (N)-SAP

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

SAP entità

API le funzionalità disponibili

(N)-SAP (Service Access Point)



Servizio del generico livello OSI

Ogni **SAP** deve avere un **nome unico** per essere identificata e ritrovata

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**

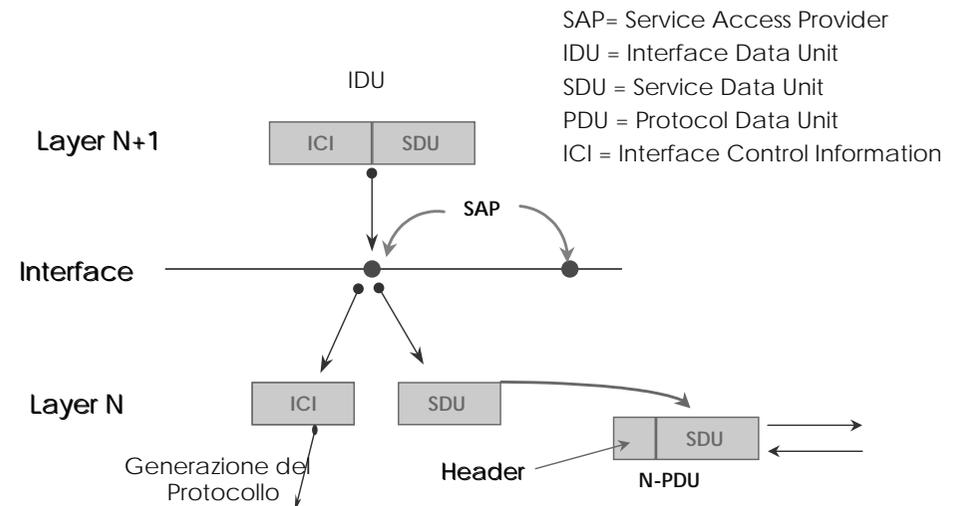
ICI (Interface Control Information) coordina operazioni

PDU di un certo livello costruito aggiungendo al

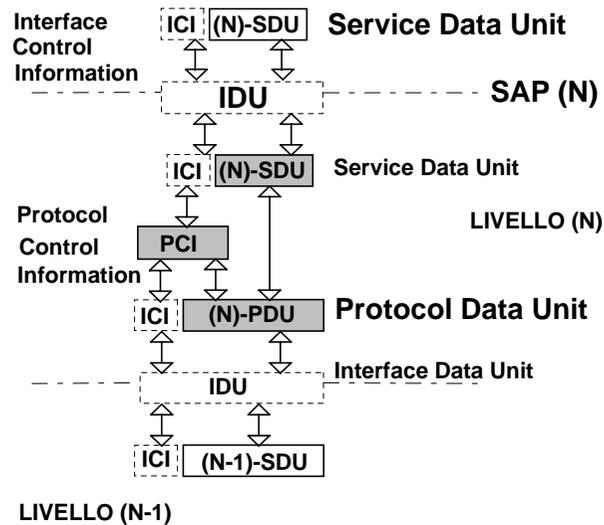
SDU del livello sopra (detta **Service Data Unit**)

informazioni di protocollo

tipicamente attraverso l'uso di servizi sottostanti



Per ogni livello

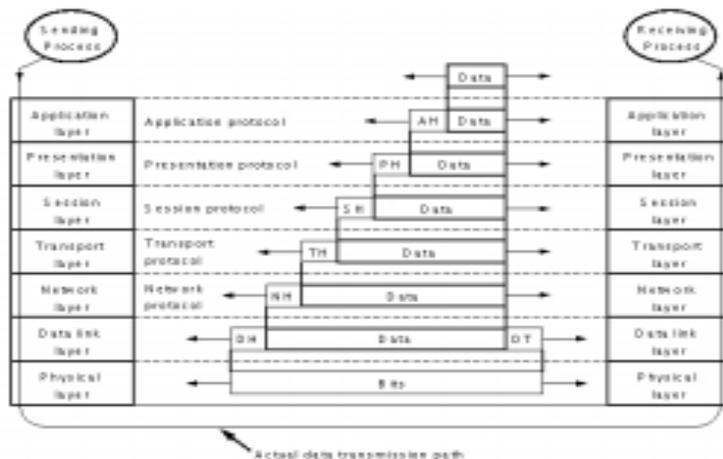


ICI determina il protocollo e diventa PCI

PCI Protocol Control Information

Il PDU viene formato aggiungendo informazioni al dato passato da sopra

Al ricevente azioni complementari



OSI 7

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**

La connessione non significa impegno di **risorse** su eventuali nodi intermedi necessariamente

- **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)**

OSI 8

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*

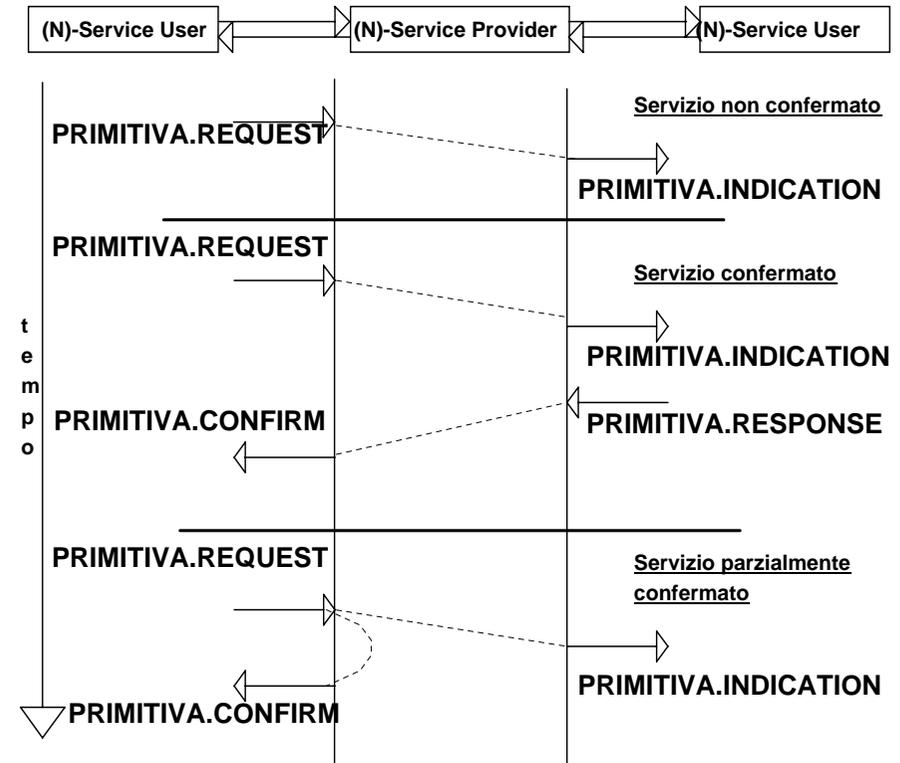
- **Request**
il service user richiede un servizio (una *azione*)
- **Indication**
il service provider indica al service user che è stato richiesto un servizio (segnalazione di *evento*)
- **Response**
il service user specifica la risposta alla richiesta di servizio (una *azione*)
- **Confirm**
il service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

senza conferma

Primitiva sincrona

con conferma al cliente e azione al servitore

Primitiva asincrona bloccante

solo conferma al cliente

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

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**

permanente o switched (**multiplexing**)

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

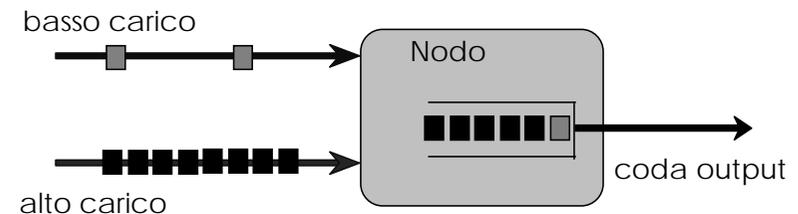
*Ogni livello può avere (anche per gli intermedi) compiti di **controllo di errore e ritrasmissione***

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



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

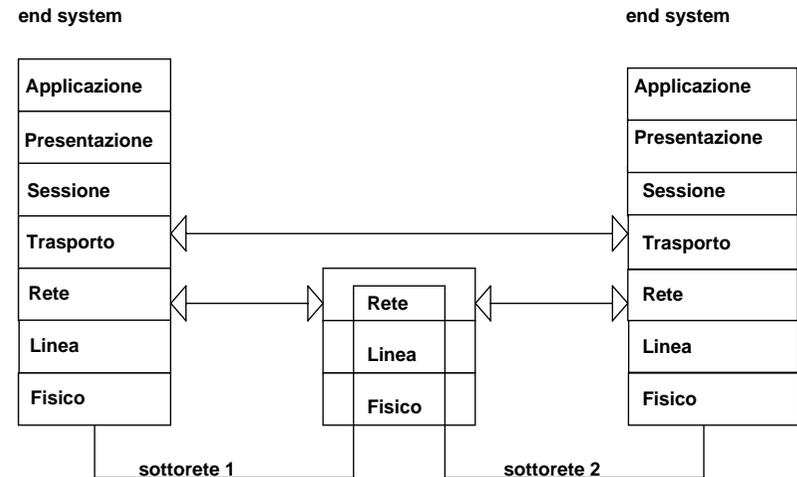
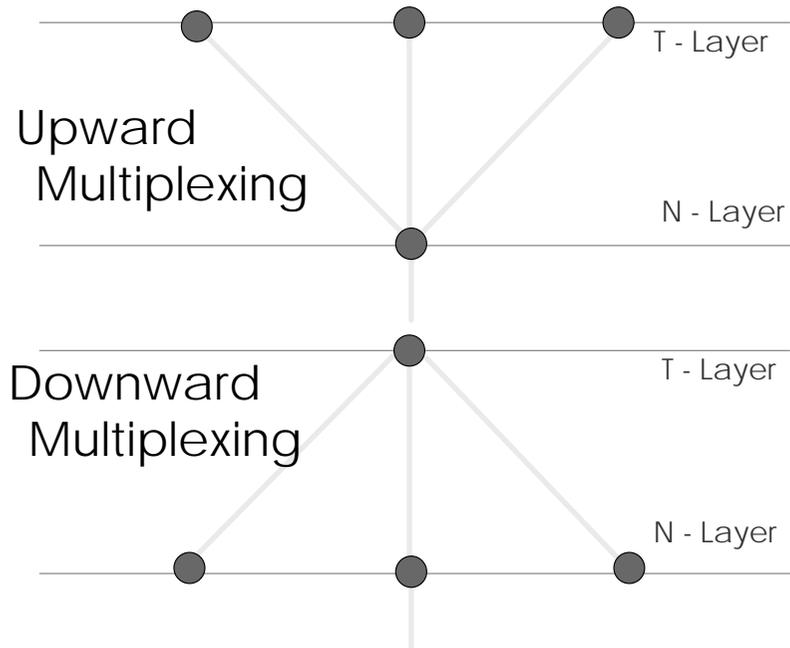
NON negoziazione vera e propria (?)

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**)



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

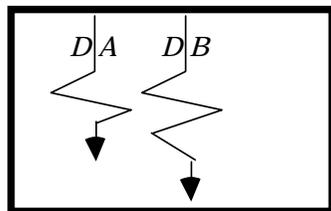
CONSIDERANDO IL DIALOGO TRA ENTITÀ

bisogna tenere in conto le possibilità tra due pari che comunicano

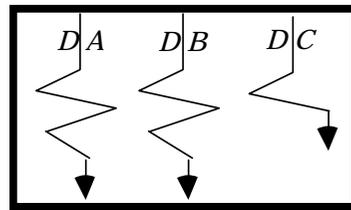
Con il trasporto arriviamo da nodo a nodo dalla sessione

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 sincro
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*

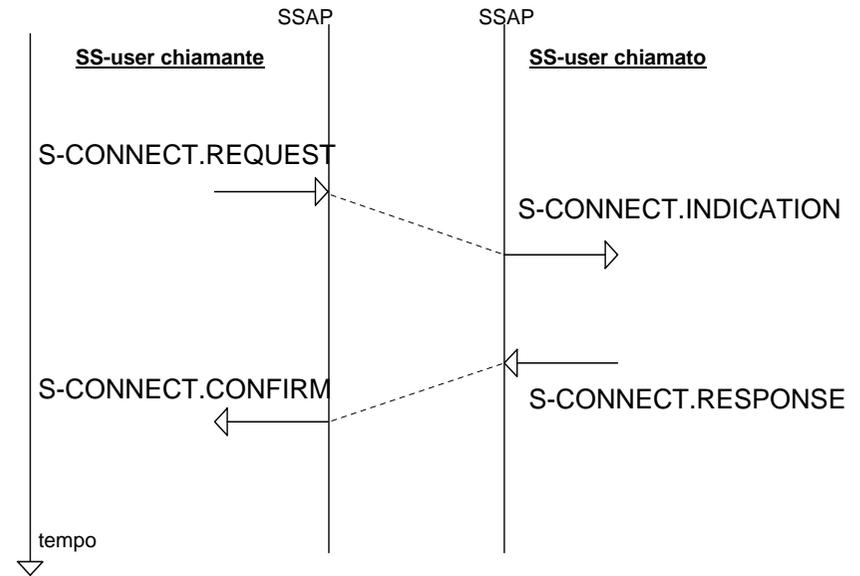
L'intersezione dei due insiemi di unità funzionali determina le unità funzionali attive nella connessione di Sessione

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

Necessità di accordarsi e 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

Negoziazione 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 (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 uno concreto 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 = "deca"
address.destination = "sunb"
length = 3
data = 'x', 'y', 'z'
    
```



```

Primitive types    BER
BOOLEAN           02
INTEGER           04
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

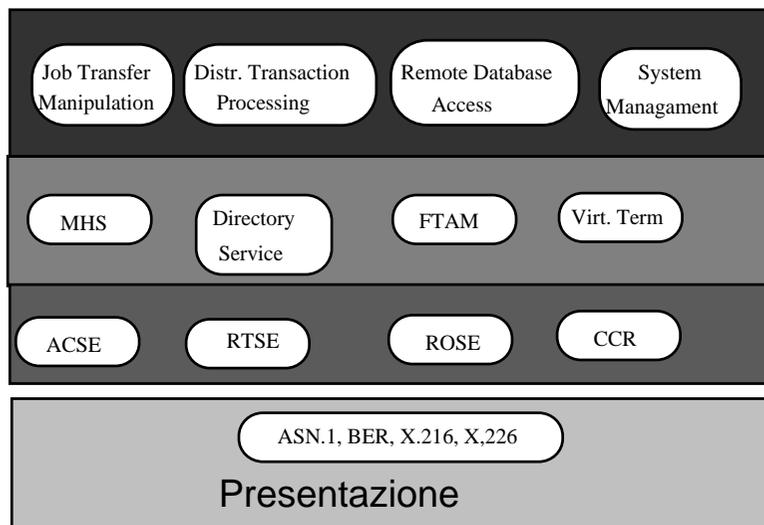
Distributed Transaction Processing

DTP

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

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

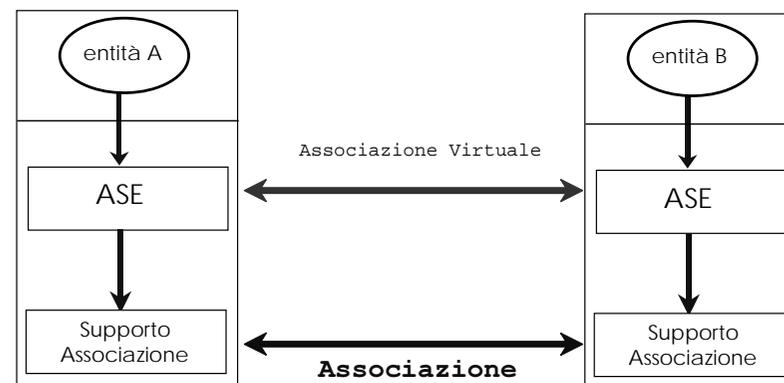
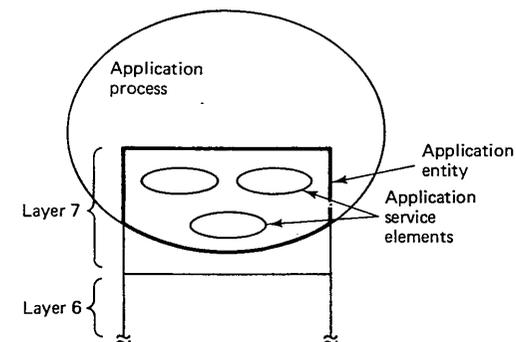
ROSE (Remote Operation Service Element)
per ottenere operazioni remote

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



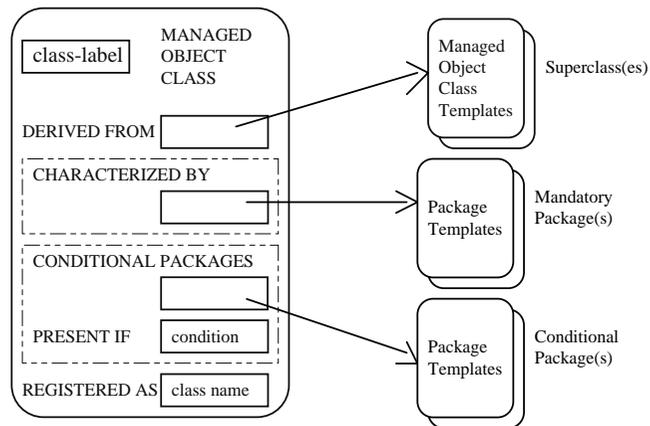
OSI - Descrizione ed Implementazione

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)

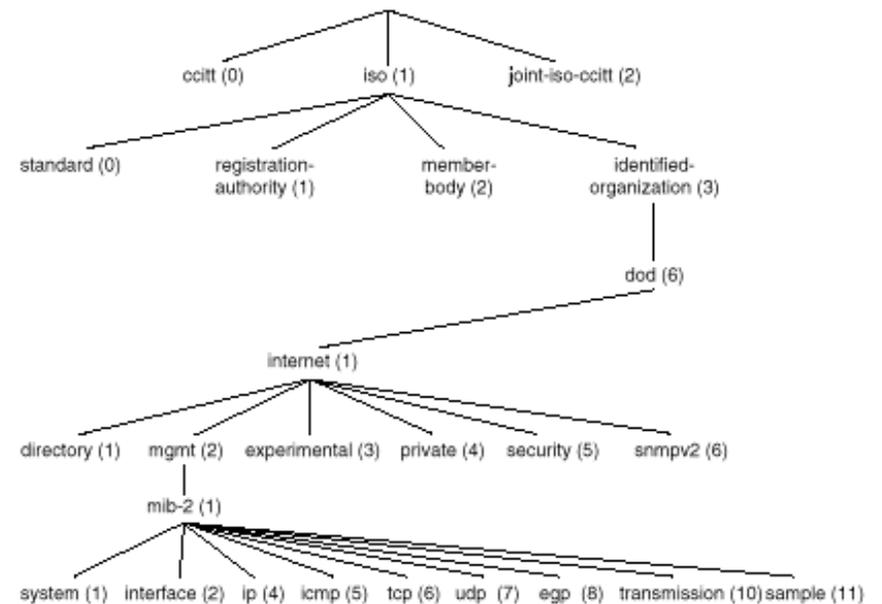
Si noti la **unicità dei nomi** come presupposto di base **NOMI UNICI** come servizio (X.500)



Servizi X.500

Il servizio di direttorio consente di collocare e classificare ogni dispositivo noto in un sistema gerarchico di conoscenza

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



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

SISTEMI DI GESTIONE

OSI introduce la possibilità di

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

Systems Management **exec (OSI)**
Network Management **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)

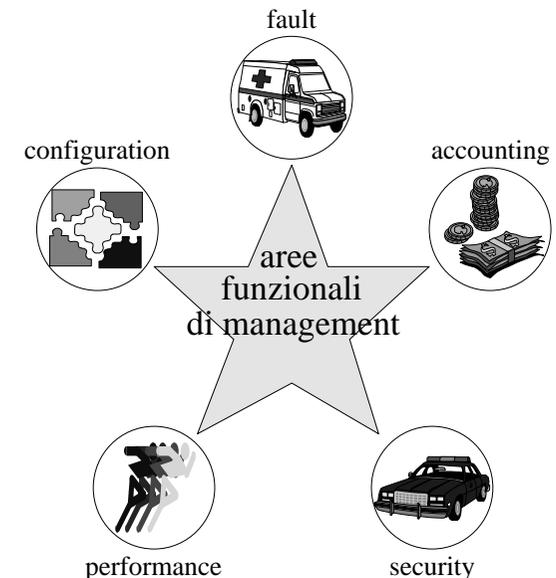
Management Information Service

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

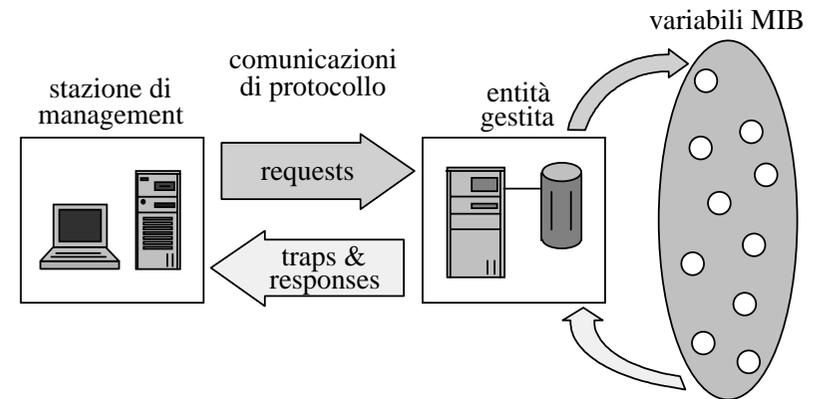
NECESSITÀ per SNMP
di ridefinizione e
di reingegnerizzazione

per tenere conto di esigenze di **sicurezza**
per tenere conto di modelli di **gestione flessibili**
per tenere conto di sistemi **legacy esistenti**

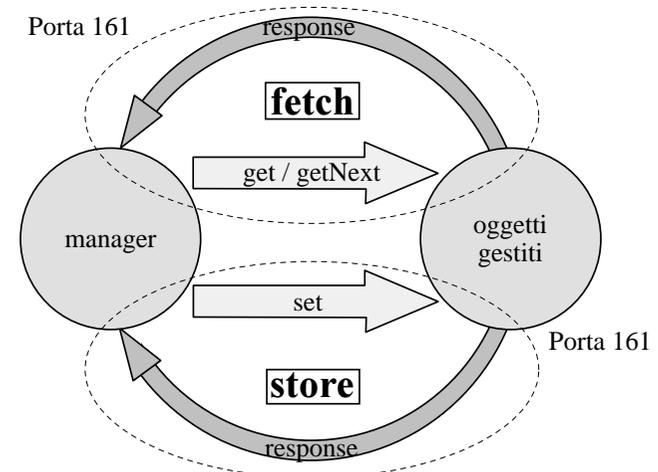
...
Inoltre, anche potere gestire non solo apparati,
ma **entità di qualunque tipo**

Protocollo SNMP

(Simple Network Management Protocol)
Estrema semplicità: solo alcune azioni elementari



Uso di messaggi molto semplici

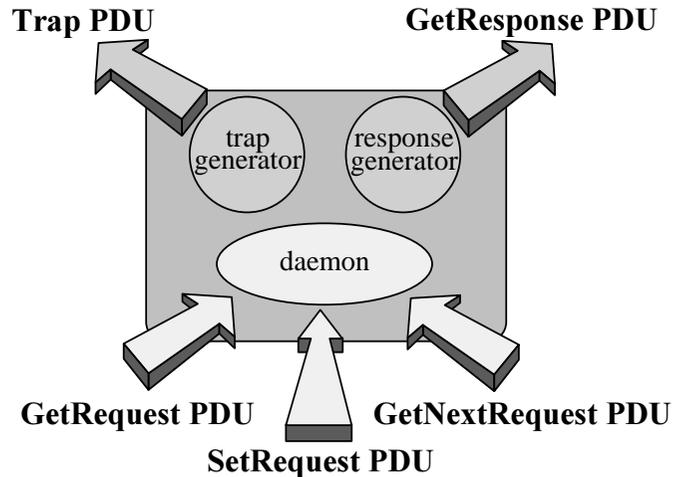


messaggi base: **Set, Get, Get_Next** (attributi multipli),
Trap (porta 162 del manager)

Protocollo SNMP

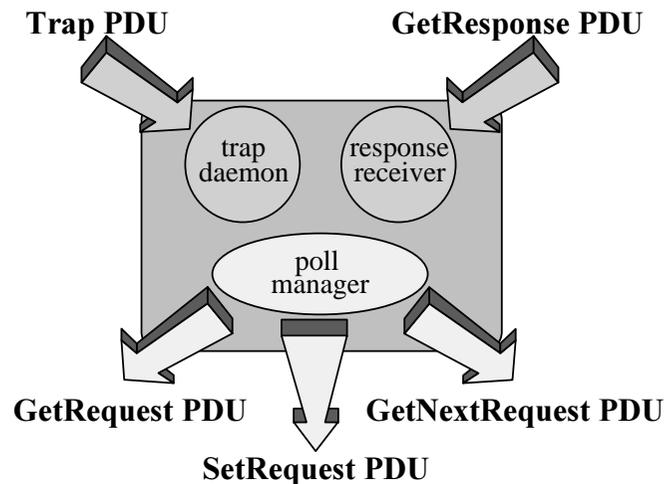
Struttura di un agente SNMP

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 MONitor

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

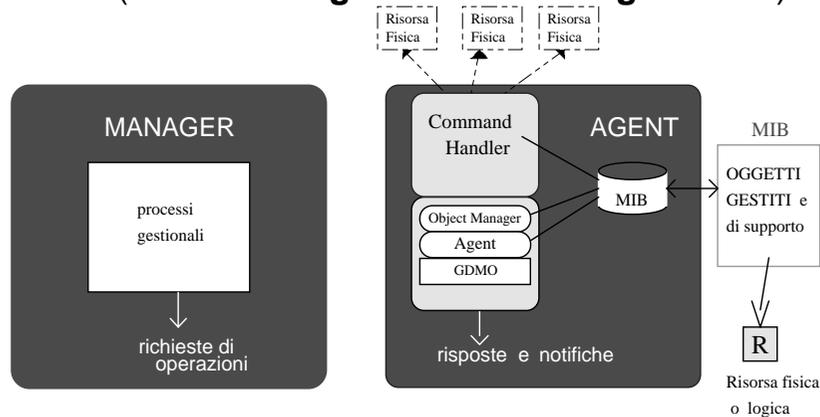
RMON1 sviluppi nel senso della azioni multiple e innestate

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

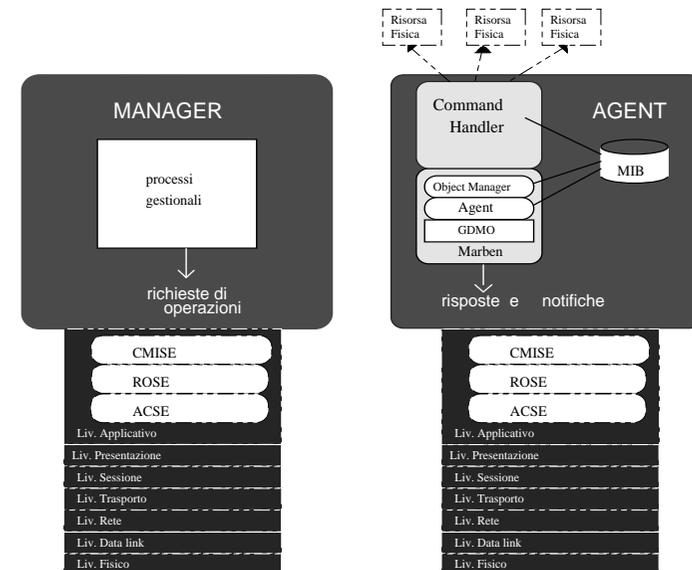
Common Management Information Service Element
CMISE

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
Usò 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)