Gestione di Reti

Federico Matteoni

A.A. 2019/20

Indice

Introduzione

Perché bisogna studiare la gestione? La situazione corrente comprende: un aumento delle risorse strategiche informative, le reti di computer che da strumento di supporto sono diventate elemento chiave delle organizzazioni, l'aumento esponenziale dei dispositivi interconnessi e aumento anche della complessità e delle funzionalità.
C'è quindi richiesta di servizi di rete permanenti e di qualità ottimale, oltre alla necessità di ridurre i costi per le infrastrutture di rete di un'azienda.

Necessità Gestione di reti eterogenee con l'aiuto dei computer.

Terminologia e concetti fondamentali

Managed Objects Il controllo, la coordinazione e il monitoraggio delle risorse avviene tramite la manipolazione dei cosiddetti **managed objects**: un MO è una visione astratta di una risorsa che presenta le proprietà dal punto di vista della gestione. Sono **rappresentazioni astratte di risorse reali**.
I confini di un MO specificano quali dettagli sono accessibili ai sistemi di monitoraggio e quali sono schermati (**black box**)

Management-System ↔ Managed Object ↔ Real Object

Caratteristiche

Attributi: descrivono lo stato/condizione dell'MO, possono cambiare quando cambia lo stato dell'oggetto reale e possono essere manipolati attraverso operazioni di management

Operazioni: consentono l'accesso all'MO. Operazioni tipiche sono get, set, create e delete, ma il numero e tipo delle operazioni influenzano perfomance e complessità dell'oggetto

Comportamento: determina la semantica e l'interazione con la risorsa reale. Normalmente definito in linguaggio naturale

Notifiche: quantità e tipologia dei messaggi, che possono essere generati da situazioni pre-definite da un MO quando avviene una specifica situazione

Management Information Base L'unione di tutti i MO contenuti in un sistema forma la MIB del sistema. La **Management Information Base** è la collezione di tutti i management object all'interno del sistema, con i loro attributi.
Una MIB deve essere conosciuta sia da chi la implementa che da chi la gestisce.

Modularità Gli MO di un sistema sono solitamente definiti in più MIB. Nelle MIB sono introdotti i moduli per consentire un design modulare: moduli diversi possono essere definiti da team diversi, le funzionalità di gestione possono essere estese e modificate...

Paradigma Gestore/Agente Agent

Implementa i MIB delle MO accedendo alle risorse reali

Riceve le richieste da un gestore, le processa e trasmette le risposte appropriate

Smista le notifica riguardanti cambiamenti di stato importanti nel MIB

Protegge gli MO da accessi non autorizzati usando regole di controllo degli accessi e autenticazione della comunicazione

Manager

Esercita il controllo delle funzioni

Avvia operazioni di gestione tramite opportune operazioni del protocollo per la manipolazione degli MO

Riceve messaggi dagli agenti e li inoltra alle applicazioni interessate per la gestione

Management Protocol Un protocollo di gestione implementa l'accesso a MO distanti attraverso la codifica di dati di gestione (management data)

0.1 lezione 2

Livello 2 consente di identificare un device sulla rete. In tutte le reti c'è la necessità di identificare la porta di rete. Ogni dispositivo ha almeno un'interfaccia di rete: loopback, che consente di far comunicare processi di rete sulla stessa macchina. `127.0.0.1` consente di parlare su stessa macchina senza trasmettere sul filo, fondamentalmente un cortocircuito.
`ifconfig` consente di vedere le interfacce di rete disponibili su unix.
Se si vuole gestire una rete è fondamentale la standardizzazione.

Output `ifconfig`. Parte degli indirizzi, no indirizzo hw su loopback perché il traffico non esce mai (loopback sulla pila OSI è nel livello 3 Network, il MAC address è sul livello 2 Data Link, che non viene toccato da loopback). Indirizzo MAC 6 byte divisi in blocchi dai due punti. I primi 3 identificano il costruttore della scheda di rete. I successivi tre identificano la scheda di rete per il costruttore, che lo setta univocamente. Ciò garantisce univocità. Per primo blocco di tre ho 16M di dispositivi possibili. I MAC address quindi **non sono univoci**, lo sono *probabilmente*. L'univocità è fondamentale sulla stessa rete. Quindi indirizzo hw identifica univocamente device sulla rete locale. divisi in due blocchi, il primo identifica costruttore della scheda di rete.

Qualsiasi dispositivo ha indirizzo hw diverso per ciascuna porta.

0.2 Ethernet

Ethernet è un cavo seriale, trasmissione e ricezione. Mezzo seriale. Un filo.
Quando si mandano dati non posso tutti insieme ma man mano. Non c'è collisione perché ricezione e trasmissione sono su due fili separati.
Pacchetti inviati nel tempo sul filo. Vengono distinti tra loro dal `preamble`. Pacchetti inviati in una direzione: preambolo, destinazione, sorgente, tipo dei dati, dati effettivi, padding (per rendere pacchetto di 64 se pacchetto è troppo corto), CRC.
Quindi per spedire pacchetto necessito di indirizzi (chi voglio e chi sono) e cosa mandare. chi sono lo so, è scritto nella scheda. Voglio conoscere indirizzo di chi voglio.
Alla connessione del cavo, se DHCP manda fuori pacchetto per richiesta quindi switch lo impara, se IP statico manda pacchetto ARP quindi switch lo impara.
MAC address randomizzato per privacy, spesso e volentieri sui dispositivi mobili.
Possibile più di un utente sulla stessa rete con soliti indirizzi, apparati avanzati se ne accorgono.

0.3 lezione 3

un pacchetto è interamente creato dal computer, quindi "non ci si può fidare"
Bisogna andare a livello fisico e autenticare, un po' come chiedere la carta d'identità. Metter in atto meccanismi che impediscano di inibire riconoscimento della sorgente.
802.1x permette di entrare in rete. Se configurato, il device prima di entrare in rete espone delle credenziali (utente, password, protocollo autenticazione...)
Da quel momento in poi **allegato** al pacchetto c'è il mio nome, ma le informazioni di autenticazione non fanno parte del pacchetto: pacchetti creati quando non c'era preoccupazione e interesse in fattori di sicurezza delle trasmissioni.
L'informazione non è parte del pacchetto ma lo riconosce in qualche altro modo il device e **rimane nel device** (Access Point). Ciò non serve per autenticazione fisica sul cavo: so che sei tu su questo cavo. Ma è necessaria per autenticazione su mezzi condivisi (wifi).
Su router MAC cambia ad ogni hop (ethernet comunicazione punto-punto), IP cambia solo se c'è NAT. Le parti da lv 3 in su non cambiano (a meno di frammentazioni...)
Robustezza delle reti si fa tramite la ridondanza. Tipico mettere più strade per spedire il traffico: load balancing.
Vale sia per corrente elettrica che per traffico di rete.

0.3.1 Aree funzionali

FCAPS per gestire **qualsiasi sistema**, da giochi a sistemi di rete. Non sono mutualmente indipendenti.

Fault Management: error detection, isolation and repair
Se qualcuno rileva malfunzionamento (riempito disco, ram, sovraccarico CPU...) lo deve notificare

Configuration Management: devo sapere com'è configurato il sistema. Leggere la configurazione è importante, così che le app si possano basare sulle API comuni e funzionare correttamente. Fondamentale capire la configurazione perché permette di definire l'amministrazione, servizi..., possibile riconoscere anche le adiacenze e "questo filo qui va su questa porta qua". che impatto ho se stacco questo cavo, o si rovina? Informazioni sufficenti per amministrare la rete

Account Management: rilevare il consumo di risorse

Performance Management: efficenze e statistiche, performance di sistema sia lato utente sia lato fornitore. Per l'utente è riuscire ad usare la rete, per l'operatore è il giusto compromesso tra investimento sul mezzo e contentezza utente.

Security Management: assicurarsi che ciò che uno fa è effettivamente possibile farlo, autoproteggendosi perché con le reti odierne posso intasare rete (volente o no) e quindi intasare internet, provocando danni

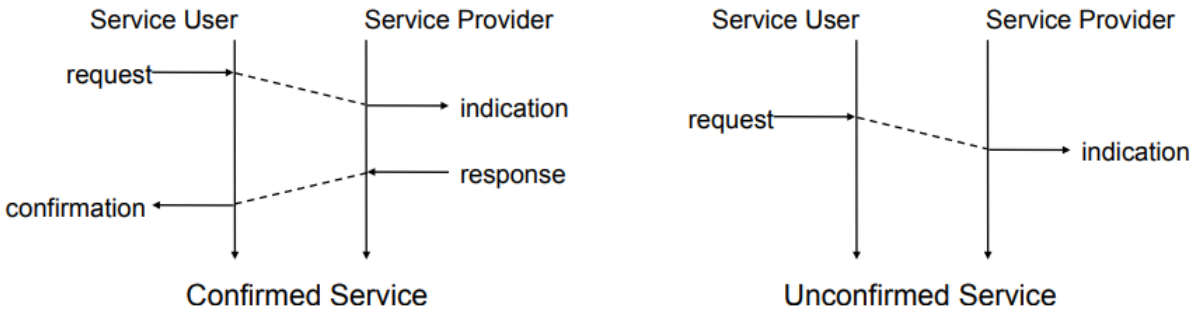
0.3.2 Interagire con management object

Primitive: get, set, create, delete
Quando faccio richiesta ad un protocollo mi aspetto una risposta: richiesta – risposta
Contenuto richieste varia durante il transito aggiungendo determinate informazioni. Es: SMS durante il transito aggiunge numero mittente per poter comunicare a destinatario chi inviava.

0.3.3 Servizi

Confermati Faccio richiesta → mi aspetto risposta.
Es: Telegram/Whatsapp

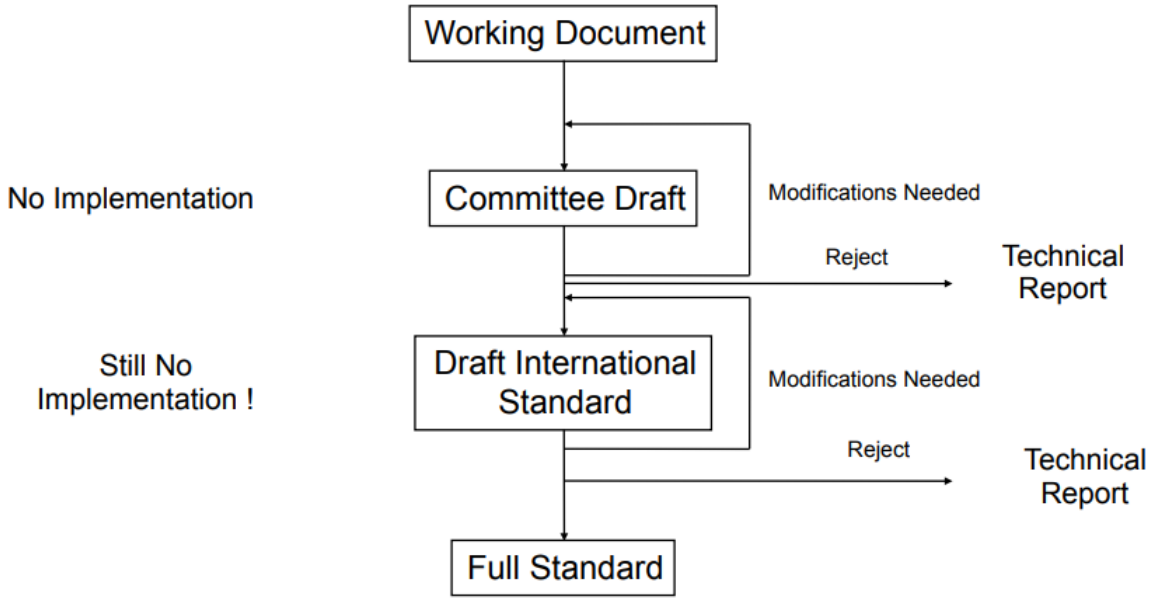
Non confermati Faccio richiesta e fine.
Es: SMS



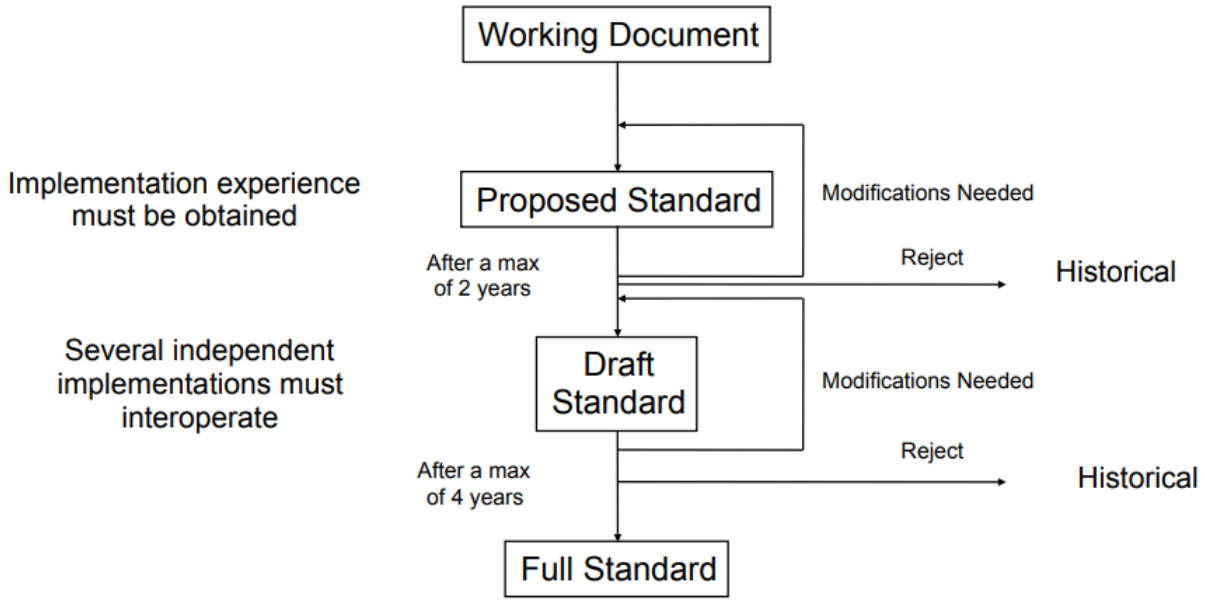
0.3.4 Standardizzazione

La grande differenza tra ISO/OSI e Internet è il processo di standardizzazione.

ISO Nella standardizzazione ISO tutte le varie aziende si accordano su come fare la rete: si creano gruppi di lavoro che si riuniscono (es. ICANN) e producono un documento di lavoro, poi vari comitati lo discutono (fino a qui **ad alto livello**).
Se si mettono d'accordo, pubblicano un **Draft International Standard** (**senza implementazione**). Dopodiché, se non viene accettato, creano technical report oppure un **full standard**.
La parte importante è l'assenza di implementazione fino ad avere la creazione dello standard.



Internet Nella standardizzazione internet è tutto seguito dal comitato IETF, che pubblica aree d'interesse in cui ritiene ci sia possibilità di sviluppo. La sottomissione di un'idea ad un'area d'interesse è libera, oppure si può mandare una mail per idee completamente nuove. Dal working document a draft passa poco e i draft dopo pochi mesi scadono. Dopo massimo due anni o si rifiuta o si fa il draft standard (draft RFC) che o lo si rifiuta o diventa standard in max 4 anni. Necessita di più implementazioni interoperabili.



0.4 Abstract syntax notation one

ASN1 Sintassi per la definizione di strutture dati e formato di messaggi. Ha l'obiettivo di consentire a macchine dalle differenti architetture hardware di scambiare dati, essere language neutral e consentire la negoziazione della codifica di trasmissione.
Come spostare le informazioni? Vari costruttori all'inizio lo facevano "in casa" senza interoperabilità. Col tempo si è reso necessario costruire qualcosa per scambiare le informazioni in maniera interoperabile.

Endian Come si ordinano i dati in spedizione, per sapere qual è il più significativo.
Bit più significativo a sx è big-endian, ormai poco usato. Altrimenti è little-endian.

0.5 lezione 4

Nel monitorare il traffico di rete c'è il problema di come riceverlo. Non sempre siamo nel posto giusto. Se voglio vedere cosa fa altro dispositivo/sottorete a livello di traffico, come faccio? Opzioni: o possiamo mettere la mano sul pc (wireshark) o posso fare finta di essere il pc (chiedendo allo switch, non intrusivamente, di mandare il traffico verso pc pure a me) Prima di iniziare a guardare il traffico, il traffico va visto.

0.5.1 Common problems with packet capture

...
Perché `root`? Perché scavalco ciò che fa un'applicazione, perché vedo tutto il traffico indipendentemente dall'applicazione. Per questo devo essere root.
Container condivide kernel con host, macchina virtuale emula il kernel.
Necessità vedere traffico. Non vorrei mettere mano sulla macchina, perché devo avere so che permette, utente, installare software...
Quindi lo faccio da fuori, prelevandolo dallo switch.
Metodi software: ho switch, che ha delle porte: port mirror: tutto traffico diretto verso tale macchina oltre a mandarglielo lo mandi anche a me su questa porta. 1:1 una porta verso una porta, 1:N tot porte switch le mandi qua. VLAN mirror: simile al port mirror: dammi tutto traffico tale VLAN e mandalo qua traffic filter/mirroring: dammi solamente traffico di tale porta tale ip...

Singolo cavo ha 2gbps (1gps in upload e 1gbps in dwnld), quindi con port mirror, che posso scaricare al massimo a 1gbps, ho efficacia se traffico sta solo a 1gbps. Dovrei avere scheda di rete da 10gbps per reggere comodamente traffico e non perderlo. Scheda direte più veloce della somma delle due direzioni.
Hardware: Network Tap prende le singole direzioni del traffico e ne fa una copia. PC monitor con due schede di rete perché tap divide il filo le direzioni: una prende le direzioni in entrata e una prende la direzione in uscita. Così scopro anche chi invia cosa. Nel port mirror non sono preservate le direzioni.

0.6 Lezione

Software di switch e infrastruttura di rete devono essere duraturi, perché infrastruttura di rete si cambia quando c'è veramente necessità. Altrimenti infrastruttura rimane lì.

Problema Gestire le cose nel tempo, che rimangano interoperabili negli anni. Siccome informatica va avanti per mode, si sono posti come problema (fatta negli anni '80) dover gestire qualche sistema non attraverso la url come si fa ora, perché è un modo di fare molto volatile che cambia spesso. Allora hanno fatto standard con negoziazione alto livello: oggetto ha attributi ad alto livello, funzionali al suo funzionamento (macchinettà caffè: c'è acqua, quanti caffè fatti, se ha bicchierini...).
Attributi, cosa ti mostro io che sia rilevante per te (non quante viti ha, ma lo stato), operazioni che si possono fare su quegli attributi (accendi, spegni...), comportamento. Software si occupa dell'accensione, non so com'è fatto internamente, io mi limito a chiamare l'operazione. Standardizzazione di funzionamento.

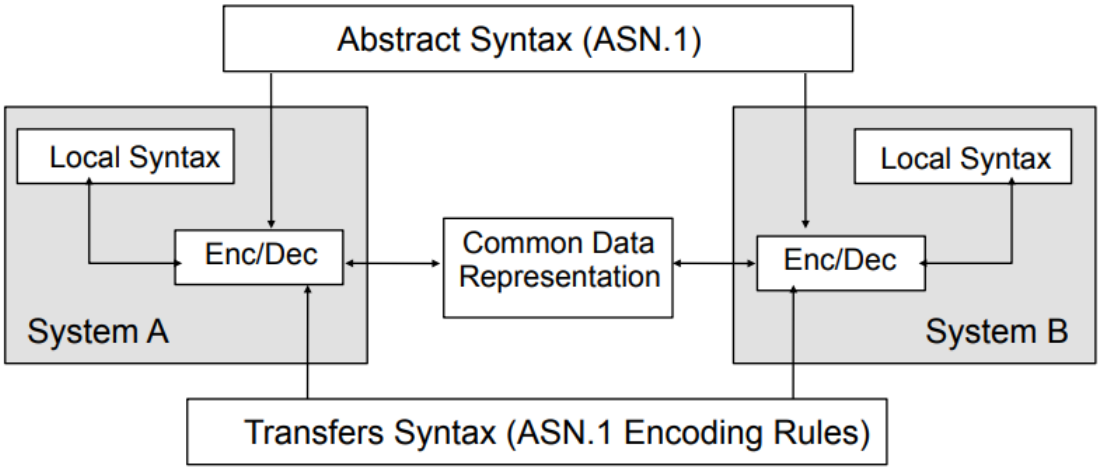
Manager Comanda l'operazione, impone politica di gestione.

Agent Gira dentro la macchina gestita e fa cosa chiede manager e solo quello.

Paradigma Agent gira nella macchina monitorata, e un solo manager raccoglie dati e visualizza.

Come realizzarlo Bisogna negoziare rappresentazione dati. Per lavorare col web hanno risolto problema scambio dati convertendo tutto a stringhe. Questo modo di fare non ha grandi problemi, ma è inefficiente. Grande quantità di dati per poche informazioni (`true` scritto invece di un bit). Poco efficiente per tanti dati, chiaro ma non compatto. La URL è breve. Tra due macchine posso scambiarmi dati in maniera binaria, ma bisogna mettersi d'accordo. Se io a 16 bit parlo con una a 64 bit non ci capiamo, quindi bisogna accordarci (Little Endian e Big Endian).

ASN1 Sintassi astratta, implementata dai linguaggi. Quando iniziano a parlare due macchine negoziano rappresentazione e codifica.



La sintassi locale (**local syntax**) è diversa e tipicamente dipendente dal linguaggio utilizzato: ad esempio una in GO l'altra in C, ma anche per sistema A Arduino Nano e sistema B workstation Windows.
L'ASN1 quindi definisce una sintassi astratta standardizzata. Permette diverse regole di codifica che trasformano la sintassi astratta in un flusso di byte adatto al trasferimento: **BER** (Basic Encoding Rules) definisce il mapping tra sintassi astratta e sintassi di trasferimento.
ASN1 rimane architettura, idea. La sintassi di trasferimento può essere JSON, GO o qualsiasi altra cosa: l'**importante è che le due applicazioni si capiscano**. I **tipi di dato** (datatypes) primitivi dell'ASN1 sono:

`BOOLEAN`

`INTEGER`

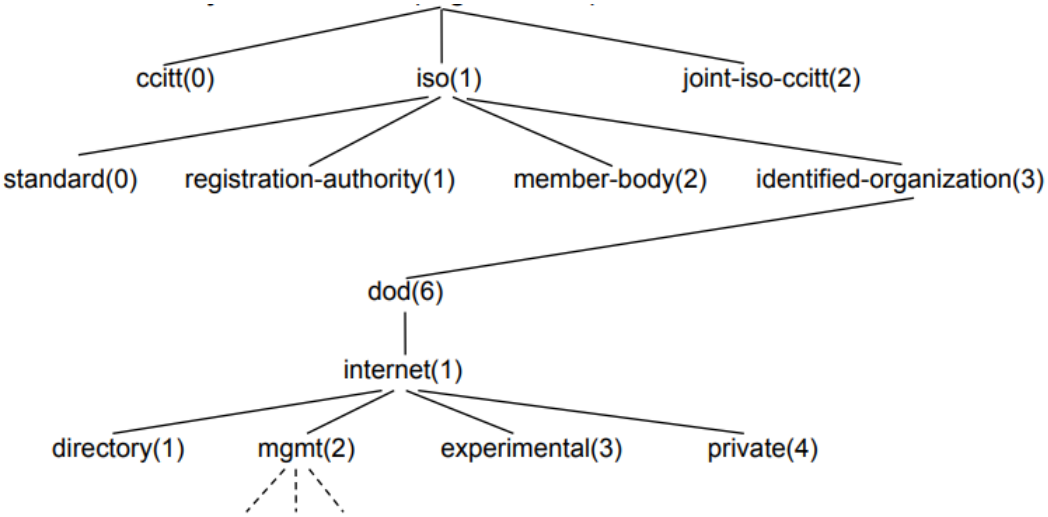
`BIT STRING`

`OCTET STRING`

`OBJECT IDENTIFIER`
Quando trasferisco dati da applicazione ad applicazione, devo poter indicare un campo chiave. Questo tipo di dato identifica univocamente l'oggetto che sto trasferendo all'interno dell'albero ISO

...

ISO Registration Tree Usato per identificare univocamente definizioni, documenti, oggetti... Ha una struttura gerarchica, simile ai file system gerarchici. Tutti i nodi di un livello sono univocamente identificati da un numero. Il **percorso dalla radice al nodo** fornisce una **sequenza numerica** chiamata **Object Identifier**.
Per esempio, Internet è 1.3.6.1



Internet si trova sotto il **Dipartimento della Difesa**, che è una **Organizzazione Identificata** facente parte dell'**ISO**. Noi parleremo della **Gestione** (**m**ana**g**e**m**en**t**) di Internet.
L'**object identifier** quindi, ovvero la sequenza di numeri, risolve il problema dell'identificazione univoca della tipologia dell'oggetto trasmesso sulla rete.

Tipi Complessi ASN1 ha anche tipi complessi, come `SEQUENCE OF` che specifica una lista di dati omogenei, o `REAL` che specifica i numeri reali con mantissa ed esponente dagli `INTEGER`.

Basic Encoding Rules Regole di codifica, compattano i dati in una stringa di byte da spedire sul filo. Basato su un algoritmo tag/length/value (TLV), dove ogni variabile è identificata da un tag, la lunghezza del valore in byte e il valore di quei byte. Questo permette al ricevente di ricostruire il tipo del messaggio a partire dal flusso di byte ricevuto.

Capitolo 1

Gestione di Rete

1.1 Nascita

Gestione di rete nacque, storicamente, nel mondo della telefonia. Necessità di codificare numero, connettersi al centralino, riconoscere il numero.... Standardizzazione necessaria per poter far telefonare a distanze elevate, attraverso le nazioni.

Dati Poi arrivò internet. Prima bisognava instradare la voce, ora vanno istradati i dati (anche voce, `VoIP`, ma pur sempre dati). Eredità di tutte le tecnologie e teorie dei tempi della telefonia (es. 5G) ma anche innovazione (es. Browser Web) ma mantenendo il paradigma che era tutto sommato efficiente.

1.2 Gestione di Rete Internet

Cos'è Sistema di protocolli e tecnologie che permettono di mettere in funzione e controllare un'infrastruttura di rete, per far sì che sia efficiente e che faccia ciò che voglio e che segnali eventuali problemi e comportamenti non previsti.

Reti Geografiche Questo discorso si applica a reti geografiche, ampie e complesse, che interconnettono un elevatissimo numero di device. Serve anche per far sì che la connessione/disconnessione di dispositivi non crei problemi, e che un utente della rete non possa creare disservizi e potenzialmente tirarla giù.

Anni '90 Il problema principale era mantenere bassi i costi, perché doveva essere pervasivo e poter mettere router in ogni casa. Centraline telefoniche, al contrario, non devono stare in ogni casa.
Gli apparati di rete quindi devono costare poco, perciò la gestione di rete non deve costare tanto (nella telefonia costa tanto ed è complicata, quindi "*non facciamo lo stesso errore*": se è semplice anche il costo computazionale è basso, quindi il dispositivo è più economico). Il protocollo, quindi doveva essere **semplice** e **efficiente**.
Altra cosa importante era **l'ubiquità** del protocollo: doveva essere disponibile su tutti i dispositivi, così da poterli gestire tutti.
Inoltre il protocollo doveva essere **estensibile**. Almeno **retrocompatibile**.

1.3 SNMP

SNMP Protocollo raggiunto "piano piano" con il diffondersi di internet, prima a livello universitario e poi industriale.
Non ci siamo arrivati tardi, perché è importante che l'infrastruttura stia in piedi. SNMP monitora lo stato della rete. Bisogna far si che la rete risponda alle esigenze.
Ci sono più standard, con primi standard nel 1990. Doveva essere **semplice**, poiché i sistemi erano semplici, non potenti, a volte anche non multitasking. SNMP non poteva girare in hardware, ma necessita di un computer perché necessita di elaborazione dati. Necessita di stack IP per comunicare. Negli anni '90 lo stack IP non era necessariamente presente sui computer. Stack IP basato sul'UDP.
SNMP è separata dallo switching, non fa parte della parte di comunicazione anche se aiuta all'instradamento. Non interferisce, come il cruscotto della macchina (SNMP) col motore.
Trasparente, fuori dalla comunicazione, ma deve poterla controllare e monitorare senza interferire. Idea è che se SNMP viene compromesso lo switch continua a funzionare.

Evoluzione 1990 versione molto semplice (SNMPv1), 1991 Management Information Base (oggetti manipolati tramite gli SNMP).
Anni successivi evoluzioni protocollo, perché fu fatto "di fretta" per avere qualcosa di pronto per il mercato che stava esplodendo. Furono privilegiate funzionalità base rispetto ad altre funzionalità.
SNMPv1 supportata da tutti, v2 aggiunge poco in più e usata soprattutto perché in v1 contatori nelle interfacce erano a 32 bit quindi sono stati aggiornati.
SNMPv3 ha perso il "Simple", quindi non è stata presa molto in considerazione dall'utenza.

Semplice Parte importante è che sia semplice e funzionare sotto UDP.

Utile Utile per monitoring centralizzato su reti importanti. è importante che questi protocolli siano in funzione in ogni momento, per riconoscere problemi in anticipo e avere uno storico della rete. Anche solo contare il traffico in byte è un'informazione importante.

Agent Apparato di rete, ad es. nella rete del Fibonacci ci sono diversi agent: access point, computer, stampanti...
Oggetti da gestire e possono cambiare nel tempo (aggiungo/tolgo, ma anche tipologia)

Trap Directed Polling Gli agent rispondon ad un manager: n agent per un manager. Tipicamente di manager ce ne sono due per ridondanza.
Con lo stesso protocollo e stessa funzionalità devo poter controllare computer, stampanti ma anche i badge, schermi... SNMP non distingue dispositivi, ma prende info dal MIB dell'agent.
La differenza tra S.O. sta nel MIB.

Polling Controllo. Manager non può continuamente comunicare con gli agent, ma ogni tanto va a contattare l'agent. Periodicamente.
Ogni agent ha proprio IP. Ma la prima versione non pensava alla sicurezza.

Traps Informazione. Una volta configurata la periferica avverte il manager se qualcosa non sta andando. Non informa il manager ogni volta che succede qualcosa (stampo un foglio, prendo un caffè, mi connetto all'Access Point) perché il manager verrebbe inondato di informazioni. Ma l'apparato informa il manager se ci sono cose che non funzionano. Se non mandi niente va tutto bene.
Può non essere sufficiente, in caso di problema di rete ad esempio. Quindi polling può risolvere questa cosa, anche se non tempestivamente.

SMI Si basa su un sottoinsieme dei datatype ASN.1 più altri sottotipi: `Integer32` interi con il segno, `Unsigned32`...`Gauge32` per misure tra min e max, `Counter32` e `Counter64` indica che il numero letto non è bello e pronto ma per sapere il valore devo fare una differenza tra due valori letti in istanti diversi.
`IpAddress` per IPv4. `TimeTicks`. `Opaque` che è come il `void`.

Scalari esistono una sola volta per agent. Es: nome di un host.
Concettuali esistono in una tabella concettuale, con valori che cambiano nel tempo.
Read e **write**: lette o scritte. Non incrementi, o resettare a valore iniziale.

SMIv2 Definita tramite macro di ASN.1

Use Case ...

Instance Identifier Concatenazione Object Identifier con identificativo dell'istanza. Prendo object identifier**.0** per identificare scalare. il .0 identifica che quell'object identifier c'è una volta sola in tutto l'agent.
Come si crea l'instance identifier? Object identifier + . + valore colonna indice.
Es. 1.3.1.2.9 $\rightarrow$ 5