**Network management and SNMP**

**Network Management (NM)**

* Introduzione al network management: motivazione e componenti principali
* Soluzioni SNMP
* Evoluzione delle soluzioni di network managements:
  + NETCONF/RESTCONF
  + YANG
  + YAML/JSON/XML

**Cosa è il network management?**

Il NM è la gestione di un particolare tipo di sistema autonomo (chiamato rete) composto da migliaia di componenti hardware/software che interagiscono tra loro.

Data la cardinalità degli elementi che interagiscono e l’eterogeneità di queste interazioni viene considerato un sistema complesso e ha quindi bisogno di opportune pratiche per essere monitorato e controllato.

Esempi di sistemi che necessitano monitoraggio e controllo: aereo, impianto nucleare, rete ferroviaria, …

Definizione (non va imparata a memoria): “Il network management include la distribuzione, l’integrazione e il coordinamento di elementi hardware, software e umani per monitorare, testare, interrogare, configurare, analizzare, valutare e controllare la rete e le risorse elementari per soddisfare i requisiti di real-time, prestazioni operative e qualità di servizio ad un COSTO RAGIONEVOLE.”

Esempi di compiti NM da gestire:

* Rilevamento del guasto di una scheda di rete su un host, un router o uno switch
* Monitoraggio degli host e/o dei dispositivi di rete
* Monitoraggio del traffico per una corretta distribuzione delle risorse (un corretto dimensionamento del sistema)
* Rilevamento di cambiamenti rapidi nelle tabelle di routing
* Monitoraggio dei Service Level Agreements (SLAs), cioè i contratti che un operatore stabilisce con i propri clienti
* Rilevamento delle intrusioni

Più formalmente, l’ISO ha definito il modello del network management utile per collocare scenari aneddotici di cui sopra in un modo più strutturato.

L’ISO (l’organizzazione internazionale che definisce gli standard) ha raggruppato tutti questi compiti in una sigla che prende il nome di FCAPS (Fault management, Configuration management, Accounting management, Performance management, Security management).

* **Performance management**: l’obiettivo è quantificare, misurare, segnalare, analizzare e controllare le prestazioni dei diversi componenti di rete. Questi componenti includono singoli dispositivi e astrazioni end-to-end, come un percorso attraverso la rete.
* **Fault management**: l’obiettivo è registrare, rilevare e rispondere alle condizioni di guasto nella rete. Il confine tra fault management e performance management non è ben delineato. Si può pensare al fault management come alla gestione immediata di guasti temporanei della rete, mentre la performance management ha una visione a lungo termine per fornire livelli accettabili di prestazioni.
* **Configuration management**: consente di tenere traccia di quali dispositivi si trovano sulla rete gestita e delle configurazioni hardware e software di questi dispositivi.
* **Accounting management**: consente di specificare, registrare e controllare l’utente e il dispositivo di accesso alle risorse di rete. Gestisce quindi i permessi degli utenti (chi può accedere dove e a che cosa, a quale macchina e a quali applicativi o a quale macchina e a quale interfaccia) e assegna i privilegi di accesso alle risorse.
* **Security management**: l’obiettivo è controllare l’accesso alle risorse di rete secondo una politica ben definita. La distribuzione delle chiavi o l’utilizzo dei firewalls per monitorare e controllare i punti di accesso esterni alla propria rete sono due componenti fondamentali.

Queste 5 aree sono le aree principali di intervento della gestione della rete.

In realtà tutte queste aree non rappresentano dei compartimenti stagni, essendo fortemente interlacciate tra loro. Ad esempio, un problema di prestazioni potrebbe essere anche essere risolto come un problema di configurazione.

Immagine che contiene testo, schermata, linea, diagramma

Descrizione generata automaticamenteLe azioni del network management sono azioni a ciclo chiuso. La principale azione con cui inizia tutto è l’osservazione del sistema (observe). Una volta che abbiamo i dati che ci permettono di capire come funziona il sistema, possiamo analizzarli per capire se effettivamente il sistema si sta comportando come vogliamo oppure no. Se la risposta è positiva lasceremo il sistema nello stato corrente e continuiamo a monitorarlo: da analyze chiudiamo il cerchio passando su observe passiamo su observe. Se invece l’elaborazione dell’analisi ci da risultati non in linea con le nostre aspettative (un guasto, un problema di accounting, ecc) allora è necessario mettere in piedi un certo numero di azioni che dovrebbero risolvere questo problema. Poi chiaramente il ciclo si chiude perché dobbiamo continuare ad osservare il sistema per capire se le azioni messe in atto effettivamente hanno risolto il problema, lo hanno mitigato o lo hanno addirittura peggiorato.

Negli ultimi anni non si parla tanto di control management quanto di AIOps (Artificial Intelligence Operations), quindi operatività del sistema guidata dall’intelligenza artificiale.

È un ciclo chiuso quello dell’intelligenza artificiale? No, dipende da caso a caso. In alcuni casi l'intelligenza artificiale può essere utilizzata solo per analizzare il sistema, in altri casi, anche per decidere quali sono le azioni da mettere in campo. Dipende dai singoli casi d’uso, dalla tecnologia che si ha a disposizione e anche dai dati che si hanno a disposizione.

Di solito l’intelligenza artificiale viene utilizzata nella fase di osservazione (vengono utilizzati gli schemi avanzati ad esempio per campionare lo stato del sistema invece di fare un monitoraggio continuo.

Iniziamo la parte del corso che si fonderà sull’osservazione del sistema, cioè come facciamo ad osservare una rete.

Un sistema è OSSERVABILE (dalla teoria dei controlli) se riesco a stimare in maniera completa lo stato del sistema a partire dall’osservazione delle sue uscite. Sulla base delle uscite posso cambiare gli ingressi per pilotare il sistema e ottenere le uscite desiderate. Un sistema osservabile è anche controllabile, c’è la dualità.

Queste sono le proprietà dei sistemi lineari e invarianti, cioè sistemi LTI.

Una rete non è né lineare né invariante -> quindi non possiamo osservare solo le uscite del sistema. Dobbiamo capire cosa è necessario osservare caso per caso, e utilizzare tecnologie e tecniche diverse per effettuare l’osservazione, anche se esistono dei principi guida generali.

Per osservare un sistema ho bisogno di piazzare delle sonde, dei dispositivi hardware, software o entrambi, che permettono di estrarre dei dati dal sistema. Una rete è un sistema molto complesso, ottenuto dall’interconnessione di migliaia di dispositivi.

Le sonde andranno posizionate in base al problema e in base a cosa voglio osservare. (Ad esempio, se sono un fornitore di servizi cloud dovrò andare a monitorare lo stato di salute di quello che metto a disposizione degli utenti; se invece ospito servizi web dovrò monitorare che i server che offro nel servizio siano funzionanti, ecc).

Le modalità di osservazione sono di vari tipi e dipendono dallo strato protocollare che andiamo ad osservare.

Immagine che contiene testo, schermata, diagramma, Carattere

Descrizione generata automaticamente**Infrastruttura per la gestione della rete**

La struttura classica per il network management si basa sul concetto di gestore o entità di gestione e dispositivo gestito.

Il dispositivo può essere sia un componente hardware che software.

Nel protocollo SNMP, che è l’infrastruttura di base per l’osservazione del sistema le informazioni sono organizzate in oggetti.

Un dispositivo gestito contiene un certo numero di oggetti e questi oggetti contengono i propri dati, che vengono prodotti, aggiornati e mantenuti all’interno dell’oggetto stesso in un database locale che prende il nome di MIB (Management Information Base).

Per osservare il sistema avrò bisogno di un’entità di gestione che va a recuperare gli oggetti e li mantiene anch’essa in un MIB locale che però rappresenta il più possibile lo stato della rete.

Affinché io possa recuperare i dati relativi agli oggetti ho bisogno che sui dispositivi gestiti vengano eseguiti dei componenti software specifici che prendono il nome di agenti.

Gli agenti vengono interrogati da un protocollo di gestione per recuperare i dati e, attraverso questo protocollo, il risultato viene raccolto e gestito in un’entità di gestione.

* Ho quindi un insieme distribuito di informazioni che sono gestite in modo diverso dai vari agenti, che vengono interrogati per recuperare i dati. Gli agenti verranno interrogati in maniera diversa in base al protocollo utilizzato e in base alla struttura della MIB.

Se tutti i dispositivi sono compatibili con SNMP, utilizzerò questo protocollo per recuperare i dati.

Se invece alcuni dispositivi utilizzano SNMP e altri non lo utilizzano, oppure parte dei dati non sono tra quelli gestiti da SNMP (caso più comune) e quindi ho bisogno di meccanismi aggiuntivi per catturare e gestire i dati, allora avrò bisogno di un’infrastruttura eterogenea che sia in grado di recuperare dati potenzialmente eterogenei, aggregarli, normalizzarli, memorizzarli, analizzarli e così via.

**Standards del network management**

Il primo tentativo di mettere in piedi un protocollo per la gestione dei dispositivi internet prende il nome di CMIP (Common Management Information Protocol):

* Progettato negli anni 80: rete che unifica gli standard del management
* Standardizzato troppo lentamente
* Non ha avuto successo perché SNMP è più semplice e si è quindi imposto

SNMP Simple Network Management Protocol:

* Roots di internet (SGMP)
  + Protocollo di monitoraggio con un semplice gateway
* Iniziato in modo semplice
* Utilizzato non solo per il network management ma anche per la gestione di dispositivi in rete (esempio: si usa SNMP per interrogare lo stato delle stampanti di rete)
* Schierato e adottato rapidamente, nel tempo è diventato più grande e complesso
* Attualmente la versione corrente è SNMP v3
  + Di fatto è lo standard del network management di base, anche perché sono state aggiunte delle estensioni di sicurezza (SNMP all’inizio era in chiaro)

**Panoramica SNMP: 4 parti fondamentali**

S perché è semplice; è infatti un protocollo senza connessione trasportato dal protocollo di trasporto UDP (questo è un esempio di protocollo importante che viaggia con UDP, l’altro è il DNS, su cui si basa tutto il funzionamento di internet).

SNMP è solo un protocollo che serve a trasportare i dati dalle unità di gestione agli agenti che girano sui dispositivi.

Quando parliamo di SNMP parliamo non solo del protocollo ma anche del MIB.

**Management Information Base (MIB)**: database di informazioni distribuite sul network management (contatori, informazioni sullo stato, versione del software, ecc). Informazioni eterogenee: stringhe (per identificare informazioni di stato, es: acceso o spento) o interi.

**Structure of Management Information (SMI)**: linguaggio di definizione per la struttura degli oggetti del MIB. Non è un componente di SNMP.

**Protocollo SNMP**: trasporta le informazioni dalle entità gestite al gestore e trasporta i comandi dal gestore alle entità gestite.

**Estensioni di sicurezza e amministrazione**: importante aggiunta in SNMP v3

**SMI: data definition language**

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamenteScopo: definire la sintassi e la semantica dei dati in maniera non ambigua, univoca. Quello che otteniamo è una qualcosa di semplice e un po’ noioso.

Per ogni oggetto avremo: tipo di dato, stato, semantica (il significato) dell’oggetto stesso.

Concetto di modulo: oggetti simili vengono raggruppati all’interno di un unico modulo; all’interno di un dispositivo ci possono essere più moduli.

Nel tipo di dato opaco ci possono scrivere qualsiasi cosa e il significato di questo dato non deve essere noto a priori al SMI.

Immagine che contiene testo, schermata, Carattere, diagramma

Descrizione generata automaticamente**SNMP MIB**

All’interno del MIB avrò uno o più moduli; ogni modulo avrà il suo identificativo (quindi c’è l’identificativo dell’oggetto e anche l’identificativo del modulo); all’interno di ogni modulo ci sono uno o più oggetti simili per semantica (es: un oggetto rappresenterà il protocollo UDP e un altro oggetto rappresenterà il protocollo TCP).

**SMI: esempio di oggetto e modulo**

Esempio del protocollo IP:

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

::= { ip 9 } è un nome compatto di tipo gerarchico e significa che questo oggetto ipInDelivers fa parte del modulo ipMIB e nella gerarchia del database è il 9° oggetto

::= {mib-2 48} è un nome compatto di tipo gerarchico e indica che è il 48° modulo del mib-2

**Esempio MIB: modulo UDP**

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

Per ciascuno di questi oggetti che fanno tutti parte del modulo UDP abbiamo una descrizione, un tipo, un nome e un identificativo numerico. L’identificativo numerico è standardizzato dall’ISO.

**SNMP naming**

I numeri separati dal punto rappresentano la navigazione all’interno di un albero (ISO Object Identifier tree): in nome ha quindi una struttura gerarchica.

La prima parte per noi è sempre fissa: 1.3.6.1

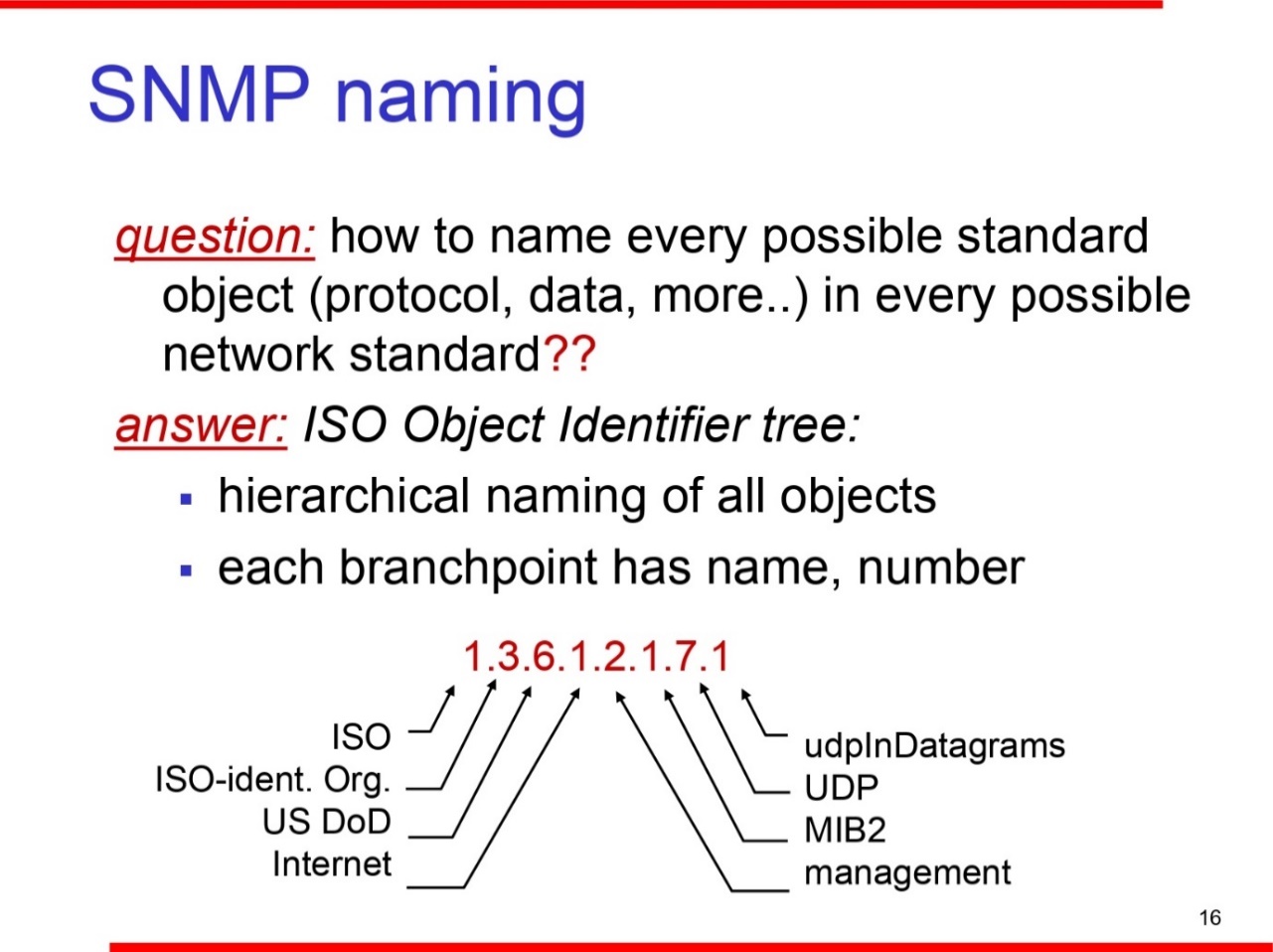


Immagine che contiene testo, diagramma, Carattere, linea

Descrizione generata automaticamente

All’inizio (prima riga) troviamo l’ente di standardizzazione e alla fine (ultima riga) troviamo tutta una serie di moduli.

Sotto ciascun modulo ci sarà poi un certo numero di oggetti, ognuno con il suo tipo di dati e la sua descrizione.

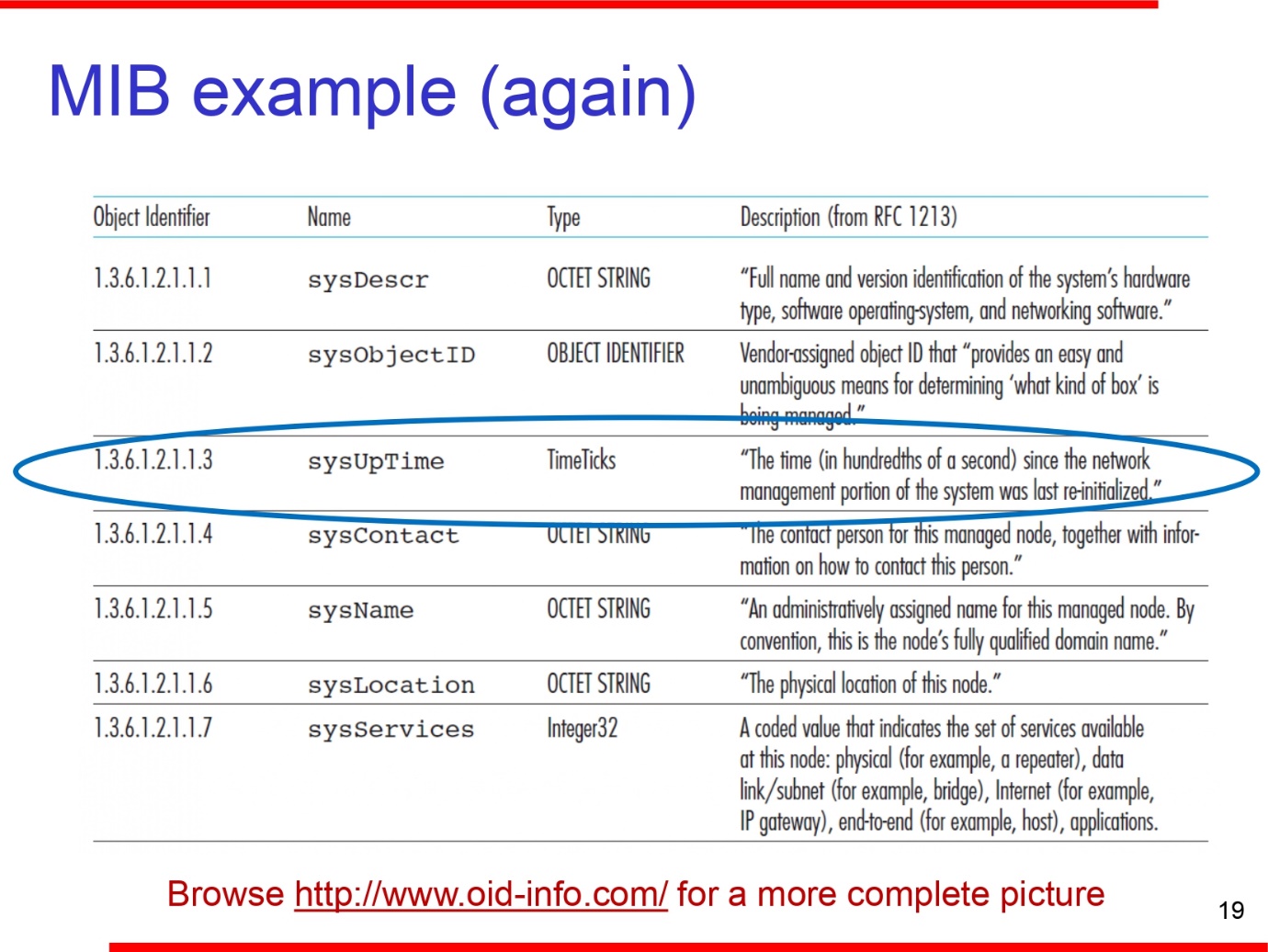
**Altro esempio di MIB**

Esempio relativo al modulo sistema.

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

Informazione compatta ::= { system 3 } indica che all’interno del modulo di sistema questo è il 3° oggetto.



**Protocollo SNMP**

Due modalità per interagire tra gestore e agenti:

* **Modalità classica**: richiesta/risposta

L’unità di gestione chiede informazione e l’agente recupera l’informazione e la invia all’interno di una risposta.

Ogni volta che il gestore vuole un’informazione la deve richiedere -> potrebbero venire inviati troppi messaggi

* **Modalità trap**: posso installare uno stato all’interno dell’agente con una condizione: quando la condizione è verificata la trap deve inviare un messaggio. La trap è una notifica inviata nel caso in cui si verifichi l’evento associato alla trap. I messaggi di trap non hanno riscontri perché trasportati con UDP.

Immagine che contiene testo, schermata, cartone animato, diagramma

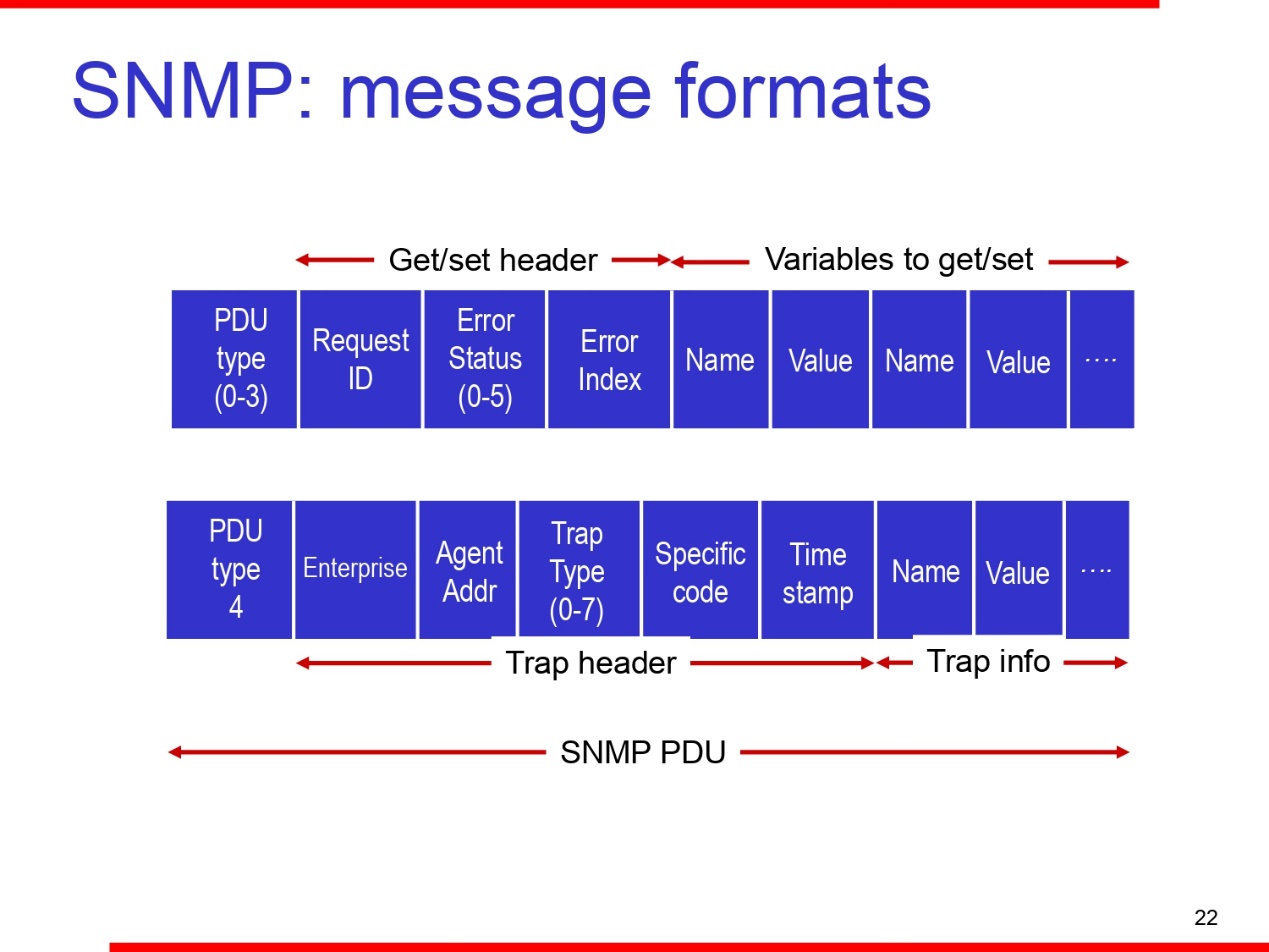
Descrizione generata automaticamente

**SNMP: tipi di messaggi**

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamenteMgr: manager (gestore)

**SNMP: formato dei messaggi**

****

**SNMP sicurezza e amministrazione**

SNMP, essendo un protocollo trasportato da UDP, è un protocollo applicativo (applicazione non di utente ma di network management).

Poiché non state definite delle stensioni di sicurezza per UDP, se voglio aggiungere sicurezza al protocollo devo farlo all’interno del protocollo applicativo stesso.

Vediamo le estensioni di sicurezza aggiunte a SNMP:

* **Cifratura**: messaggio SNMP con cifratura DES (sistema a chiave comune: mittente e destinatario condividono la stessa chiave)
* **Autenticazione**: per la protezione dell’integrità dei messaggi viene utilizzato un MIC (hash crittografico). Al pacchetto m viene agganciato un hash crittografico che dipende dal contenuto del pacchetto e dalla chiave segreta relativa all’hash.

Il pacchetto, di dimensione variabile, se viene fatto passare attraverso una funzione di hashing, si ottiene una stringa di dimensione fissa, indipendentemente dalla dimensione iniziale del pacchetto. Viene poi cifrata la stringa.

In trasmissione viene calcolato e inviato il MIC(m,k):

calcolare l’hash (MIC) sul messaggio (m) utilizzando chiave condivisa segreta (k).

In ricezione, assumendo che il ricevitore abbia la chiave, il messaggio viene concatenato con la chiave; viene applicata sopra questo la funzione di hashing nota a priori e vede se ottiene lo stesso hash. Se è lo stesso ok, se non è lo stesso il pacchetto non è integro e dovrà essere scartato.

* **Protezione contro gli attacchi di playback**: gli attacchi di playback sono quelli con cui uno stesso pacchetto viene inviato più volte per forzare il sistema ad avere lo stesso comportamento. Sono attacchi che possono potenzialmente dare origine a situazioni di DoS.

Per proteggersi contro gli attacchi di playback devo rendere univoca ogni richiesta. Viene utilizzato quindi un contatore, chiamato nonce, basato sull’ultima volta che il software del nm del ricevitore è stato riavviato.

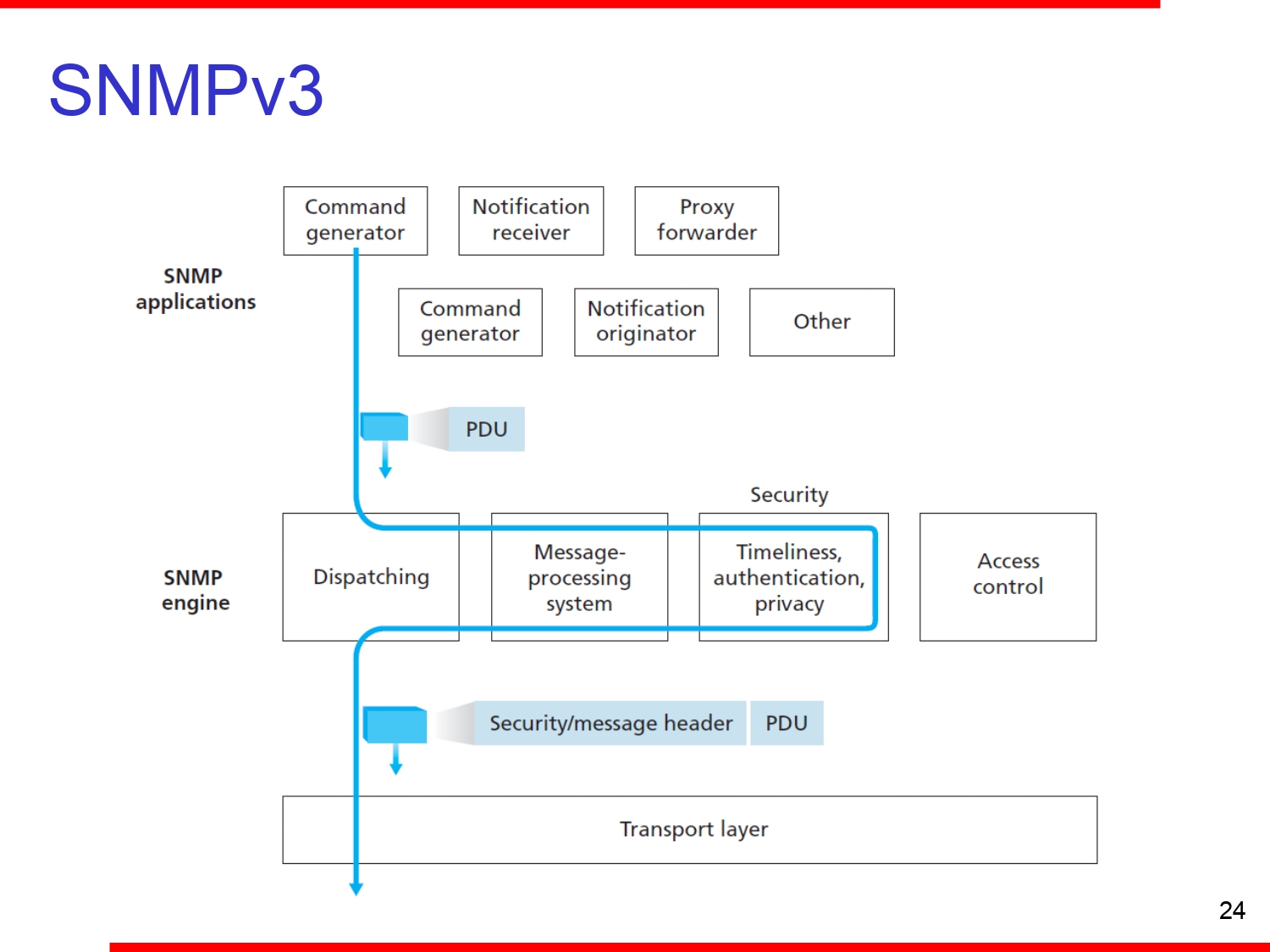
Il mittente include un valore in ciascun messaggio basato su un contatore nel ricevitore, che riflette l’ora dall’ultimo riavvio del software del nm del ricevitore.

* **Controllo degli accessi basato sulla vista**:
  + L’entità SNMP mantiene il database dei diritti di accesso per i vari utenti
  + Il database stesso è accessibile come un oggetto gestito!

Tutte queste misure sono comunque abbastanza deboli: la cifratura viene fatta con il DES; l’autenticazione non prevede la cifratura dell’hash; la protezione contro l’attacco di playback utilizza un’informazione che può essere dedotta; il controllo degli accessi viene basato sull’entità, senza particolari misure di sicurezza.

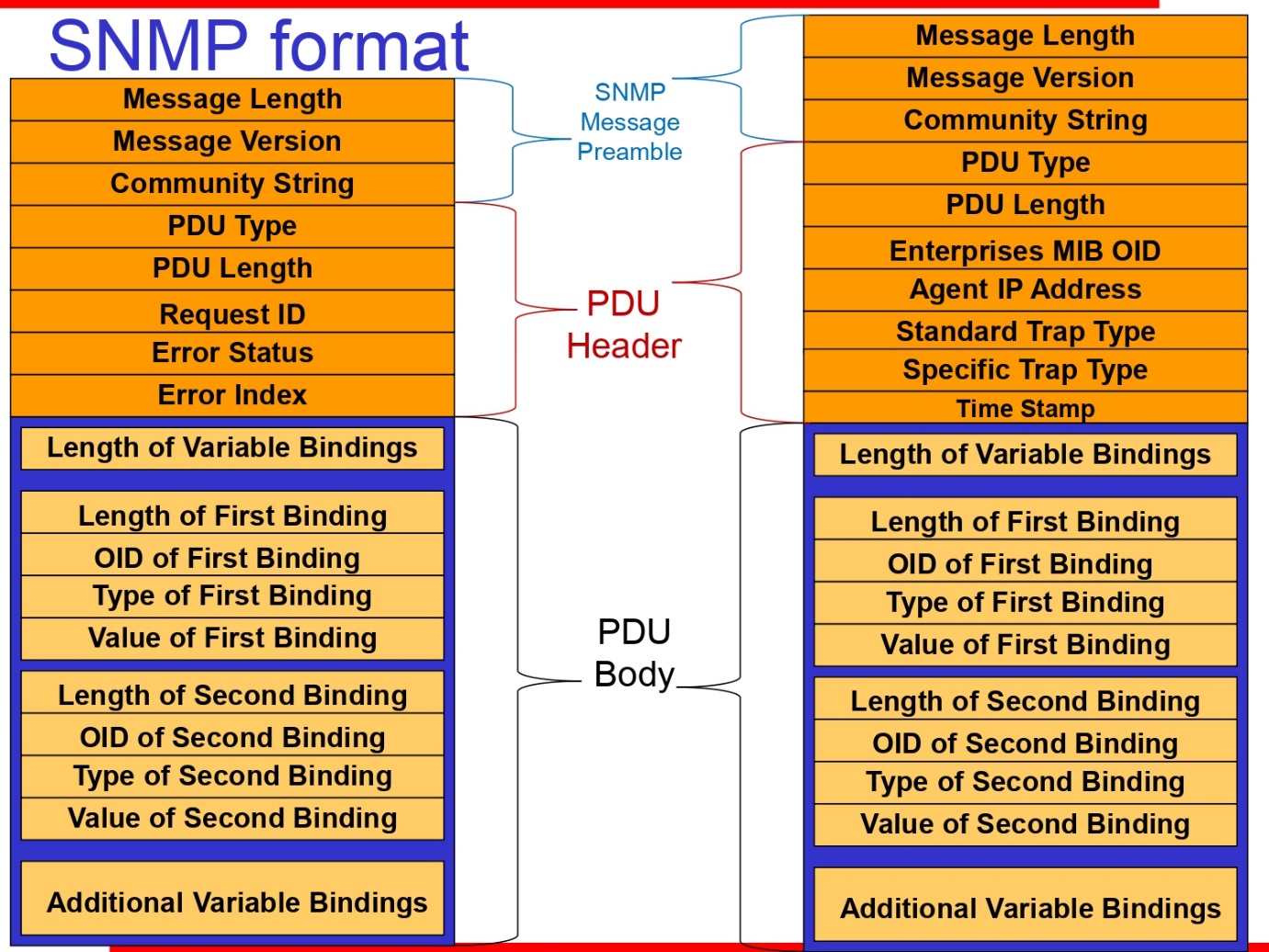
**SNMPv3**

Il meccanismo di base di come funziona l’estensione di sicurezza per l’SNMP è il seguente:



L’applicazione SNMP invia un comando e quindi genera una PDU di tipo SNMP. La PDU entra all’interno del protocollo SNMP che aggiungerà le caratteristiche di timeliness (utilizzo del contatore per prevenire l’attacco di replay), autenticazione (aggiunta hash crittografico) e privacy (cifratura). Viene aggiunta quindi alla PDU l’intestazione crittografica e viene poi inviata al protocollo di trasporto, che è l’UDP e non aggiunge niente di particolare.

Come sono fatti i pacchetti SNMP:



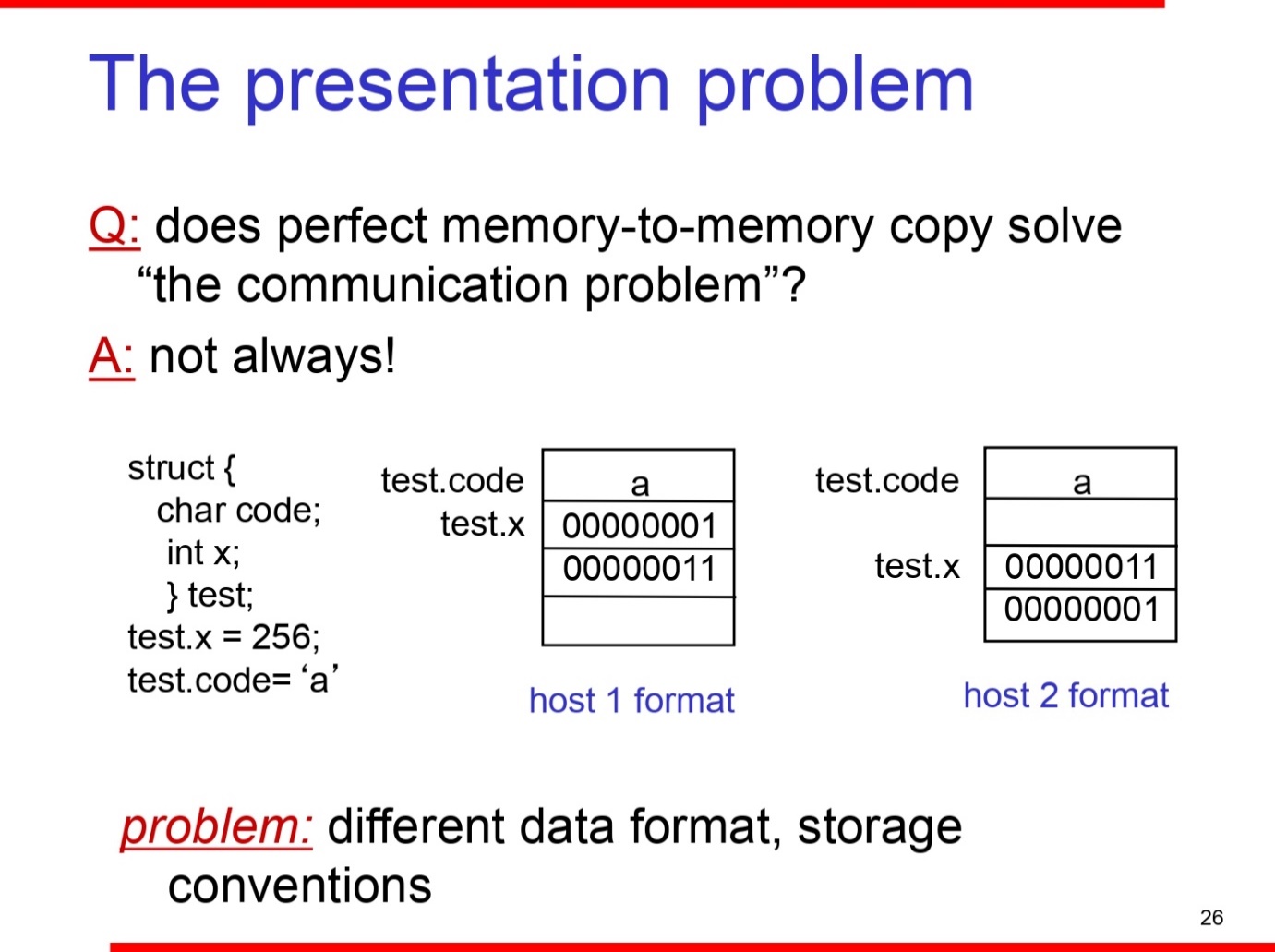
A sinistra: messaggio di richiesta/risposta

A destra: messaggio di trap

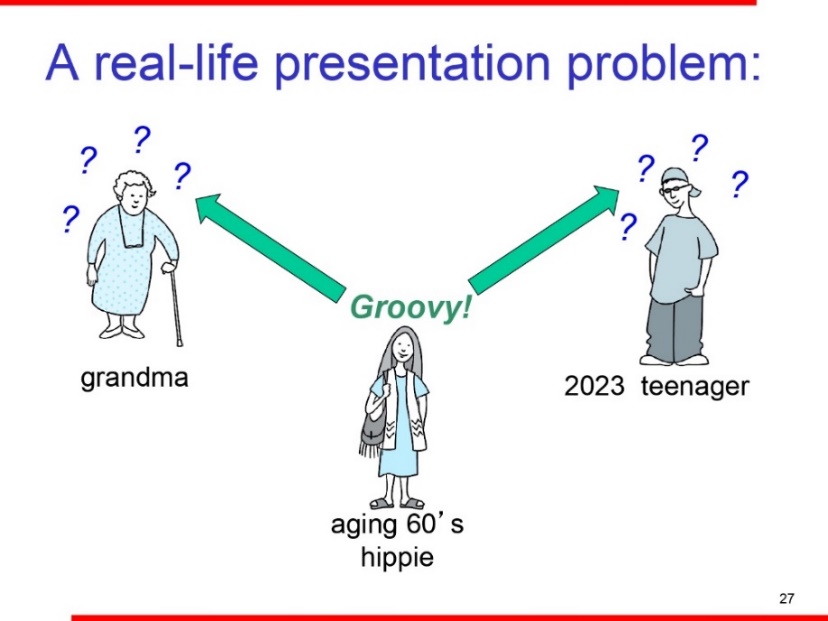
Nel PDU body ho all’inizio un’informazione che ci dice quanti blocchi ho e poi ciascun blocco rappresenta una diversa informazione. Riesco a capire quando un blocco inizia e finisce grazie alla codifica TLV.

**Il problema di presentazione**

Problema: diversi formati dei dati, dobbiamo trovare delle convenzioni di storage



**Un problema di presentazione nella vita reale:**



**Problema di presentazione: potenziali soluzioni**

1. Il mittente conosce il formato del destinatario.

Prepara prima la propria richiesta, la traduce nel formato del destinatario e poi la invia.

* Analogia con il mondo reale: fare una domanda a una persona di lingua inglese (pensi la domanda in italiano, la traduci in inglese e poi gliela fai)

1. Il mittente invia la richiesta. Il destinatario conosce il formato del mittente, quindi traduce la richiesta nel formato locale del destinatario stesso.

* Analogia con il mondo reale: la persona di lingua inglese risponde, io conosco l’inglese; quindi, traduco in italiano nella testa e acquisisco l’informazione.

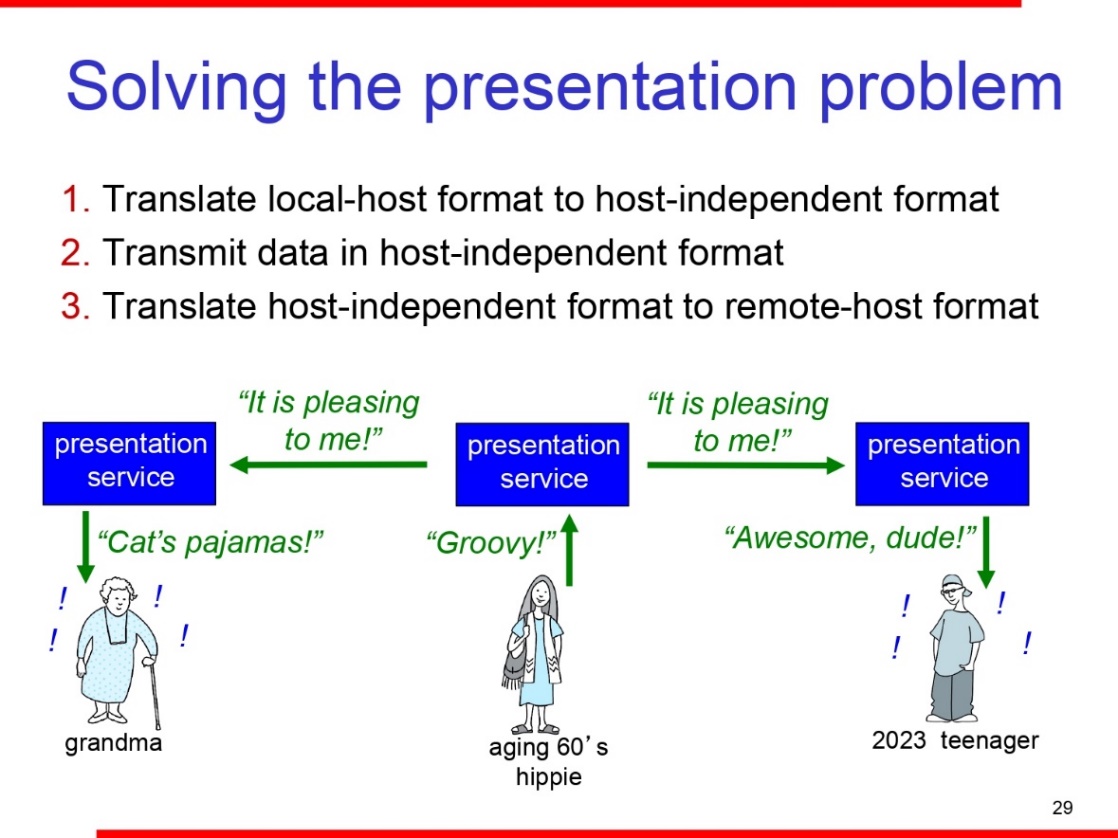
1. Utilizzare un sistema terzo noto a tutti in cui convertire le informazioni e che permetterà anche al destinatario di interpretare le informazioni in, maniera corretta, attraverso la conversione dal formato comune al formato proprio. Questo formato prende il nome di ASN.

* Analogia con il mondo reale: devo parlare con uno spagnolo e utilizzo l’inglese.

**Soluzione del problema di presentazione**

A livello di applicazione abbiamo la soluzione al problema di presentazione

1. Traduzione del formato dell'host locale in un formato indipendente dall'host
2. Trasmettere dati in formato indipendente dall'host
3. Traduzione del formato indipendente dall'host nel formato dell’host remoto



**ASN.1: Abstract Syntax Notation 1**

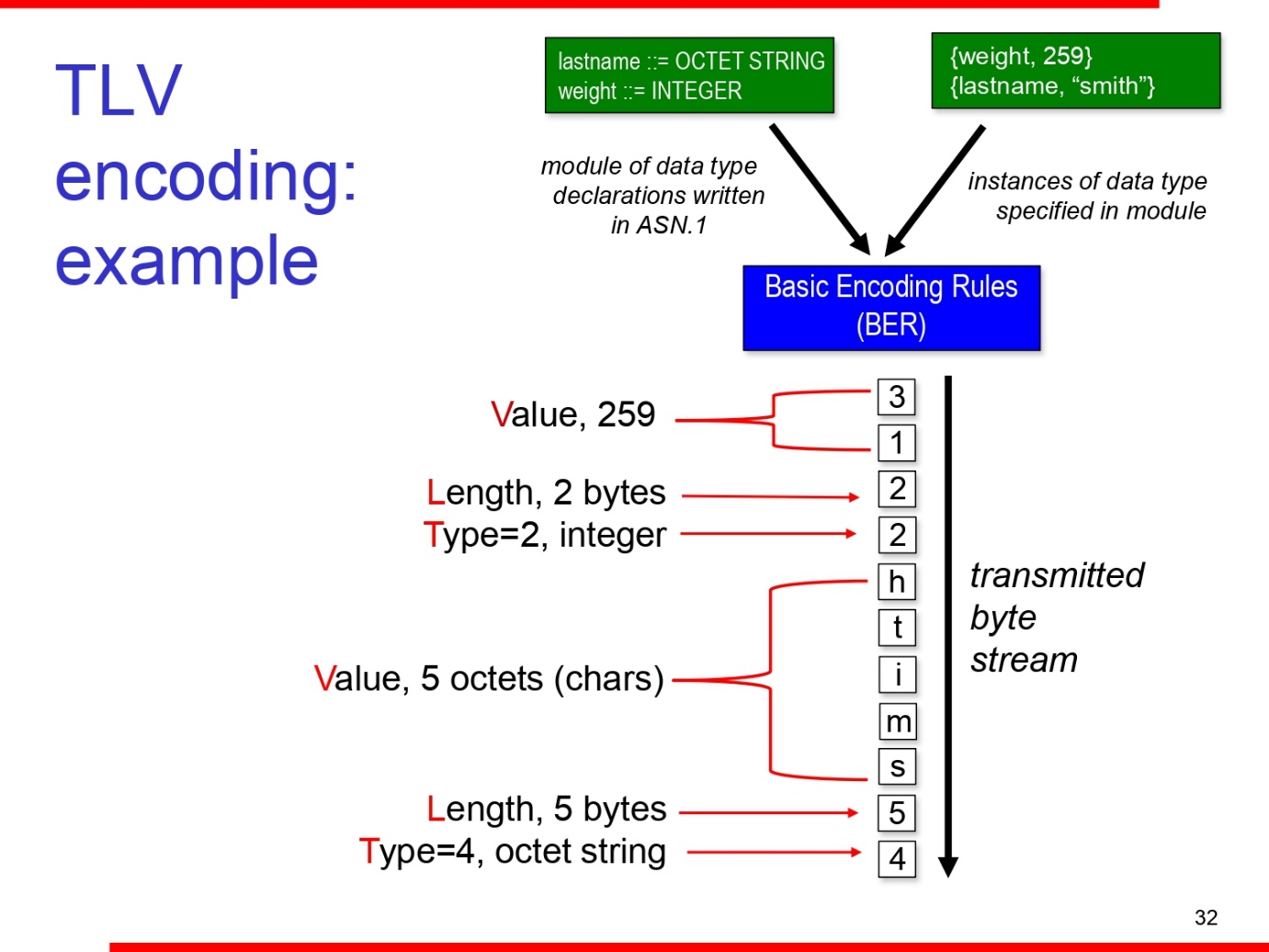
* Standard ISO X.680
  + Usato estensivamente in Internet
* Definisce i tipo di dati e i costruttori degli oggetti
  + Come SMI (SMI molto verboso e utilizzato all’interno del MIB; ASN.1 viene invece utilizzato all’interno dei messaggi che viaggiano in rete)
* Definisce il BER: Basic Encoding Rules
  + specifica come devono essere formattate le informazioni che viaggiano in rete
  + ogni oggetto trasmesso ha la codifica TLV (Tipo, Lunghezza, Valore)

**Codifica TLV**

Idea: i dati trasmessi sono autoidentificativi

* Immagine che contiene testo, schermata, Carattere, numero

  Descrizione generata automaticamenteT: tipo di dato, uno dei tipi definiti da ASN.1; l’informazione sul tipo è normalmente di lunghezza fissa (i tipi sono 9: bastano due bit per rappresentarli)
* L: lunghezza dei dati in byte (ci dice quanto è lungo il campo valore); anche questo è di solito un campo di dimensione fissa
* V: valore dei dati, codificato secondo lo standard ASN.1; campo di lunghezza variabile



**NETCONF panoramica**

SNMP offre solo la possibilità di raccogliere informazioni aggregate (e non puntuali!). Le capacità di SNMP nel tempo sono state quindi complementate da altri protocolli (li vedremo in seguito).

Vediamo ora alcuni protocolli che si sono affiancati ad SNMP perché sono più moderni ma hanno le stesse caratteristiche e anche gli stessi limiti.

Il primo protocollo è NETCONF (Network Configuration).

Obiettivo: gestire/configurare attivamente i dispositivi a livello di rete. Protocollo alternativo alla configurazione dei dispositivi via SSH o via interfaccia utente.

NETCONF pusha un file di configurazione direttamente nel dispositivo e lo carica. Permette quindi di cambiare runtime la configurazione di un dispositivo tramite dei commit atomici (uno isolato dall’altro).

Funziona tra il server di gestione e i dispositivi di rete gestiti:

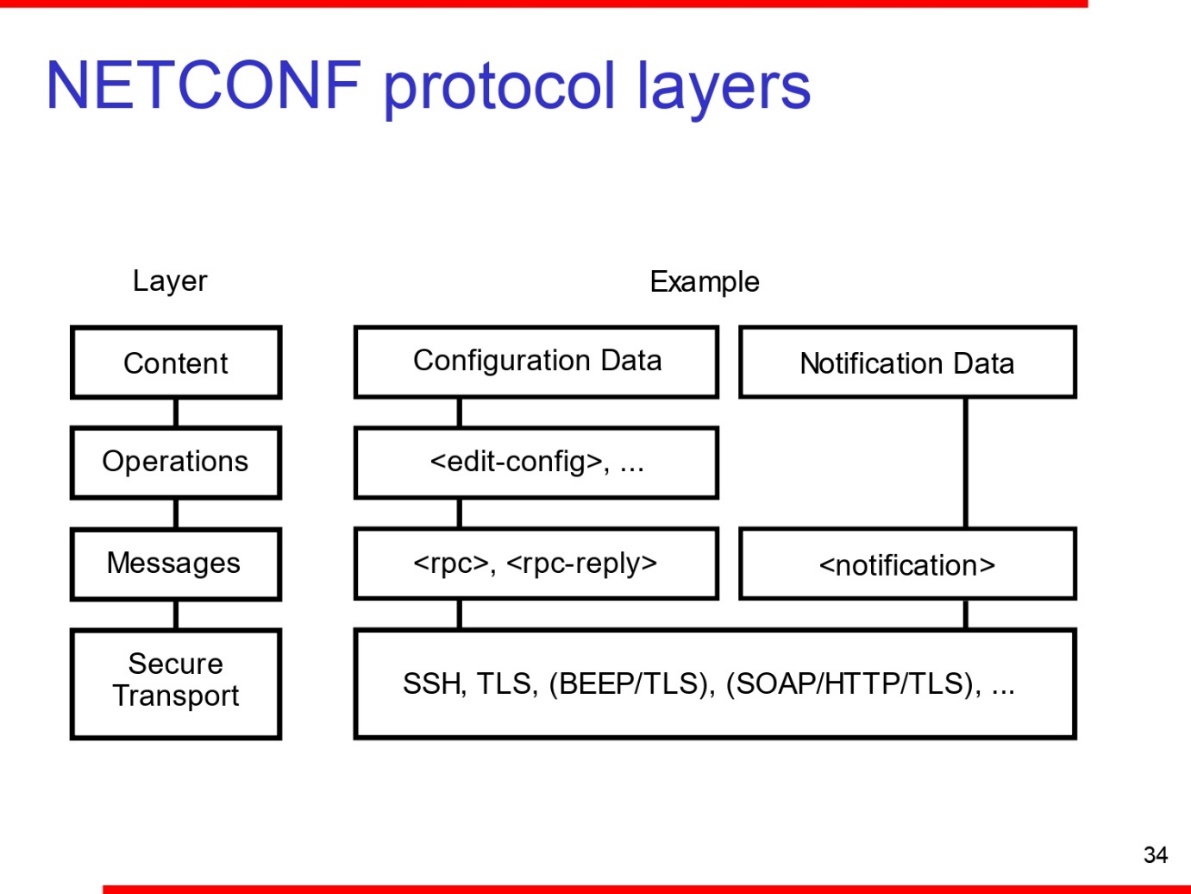
* azioni: recuperare, impostare, modificare, attivare le varie configurazioni
* azioni di commit atomico su più dispositivi
* interrogare dati operativi e statistiche
* sottoscrive una notifica e poi quando si verifica la condizione associata a quella verifica parte un messaggio di notifica verso chi ha impostato la notifica

Utilizza il protocollo RPC (Remote Procedure Call) in cui un client remoto, invece di invocare una funzione che gira su se stesso, invoca una funziona che gira su un server remoto.

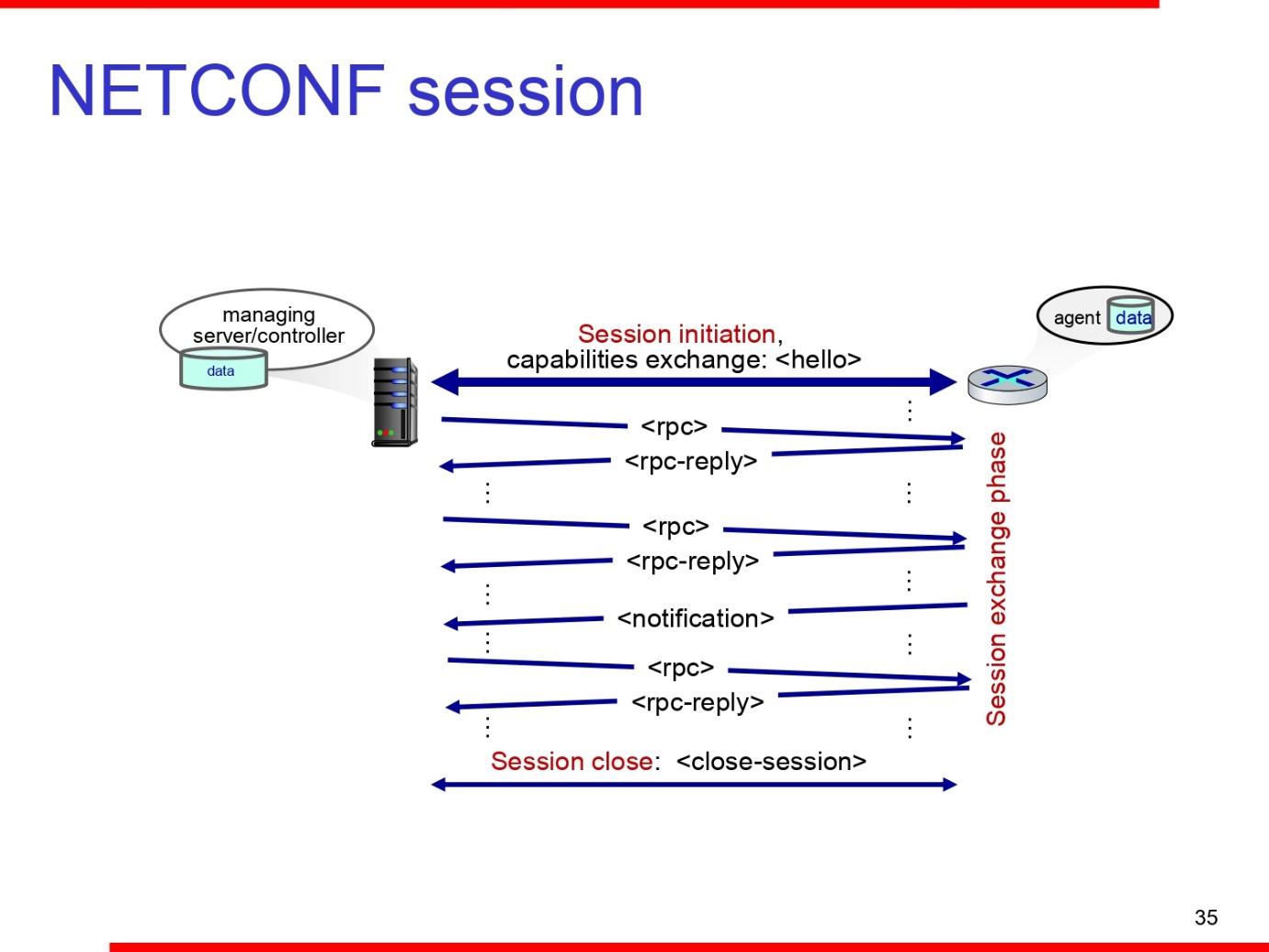
* NETCONF codifica i messaggi in XML
* tipicamente viene utilizzato il TLS per inviare messaggi di NETCONF (trasporto sicuro e affidabile)

Differenza con SNMP: SNMP è più orientato al monitoraggio, NETCONF è più orientato alla configurazione attiva.

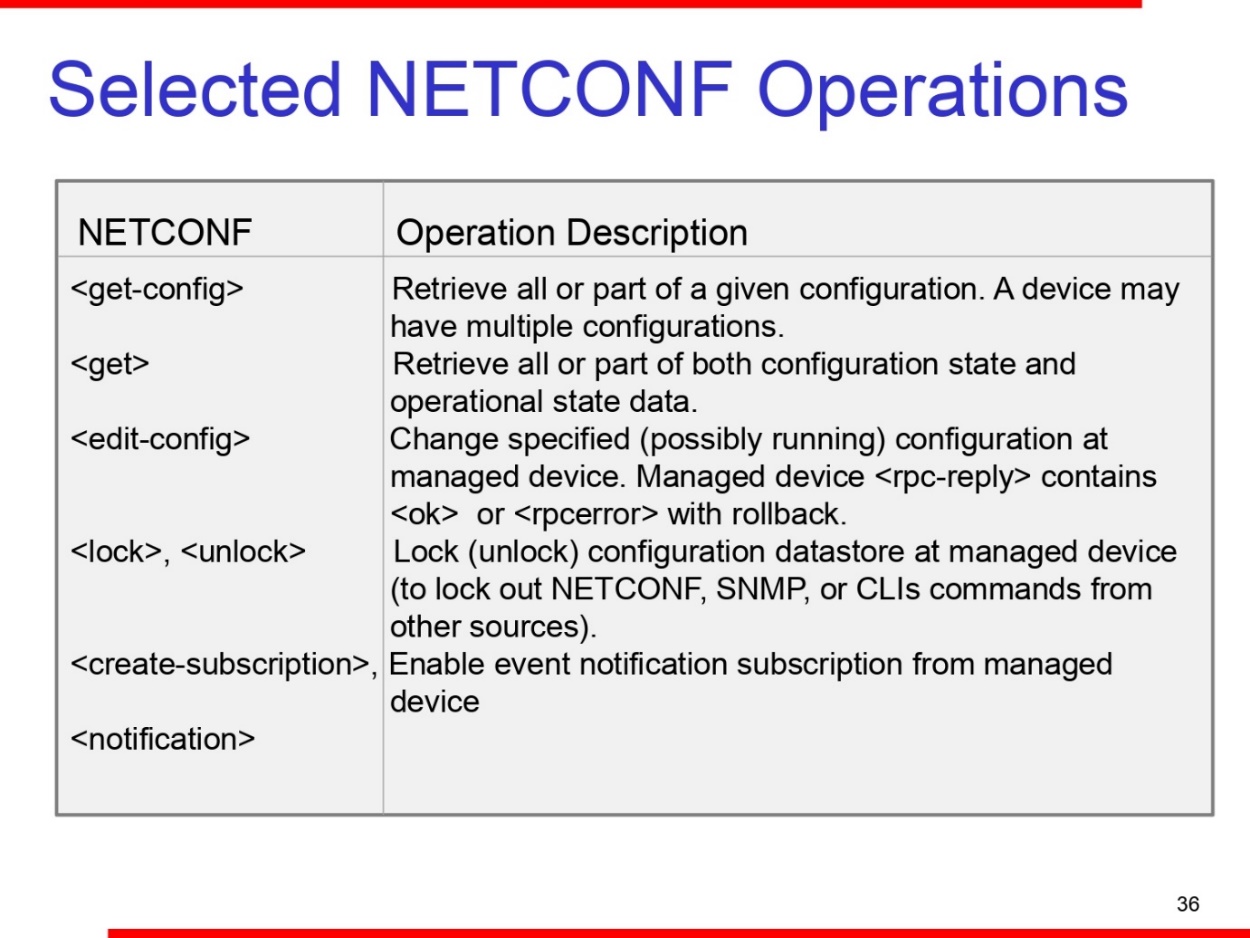
**NETCONF livelli di protocollo**



**NETCONF sessione**



**NETCONF operazioni permesse**



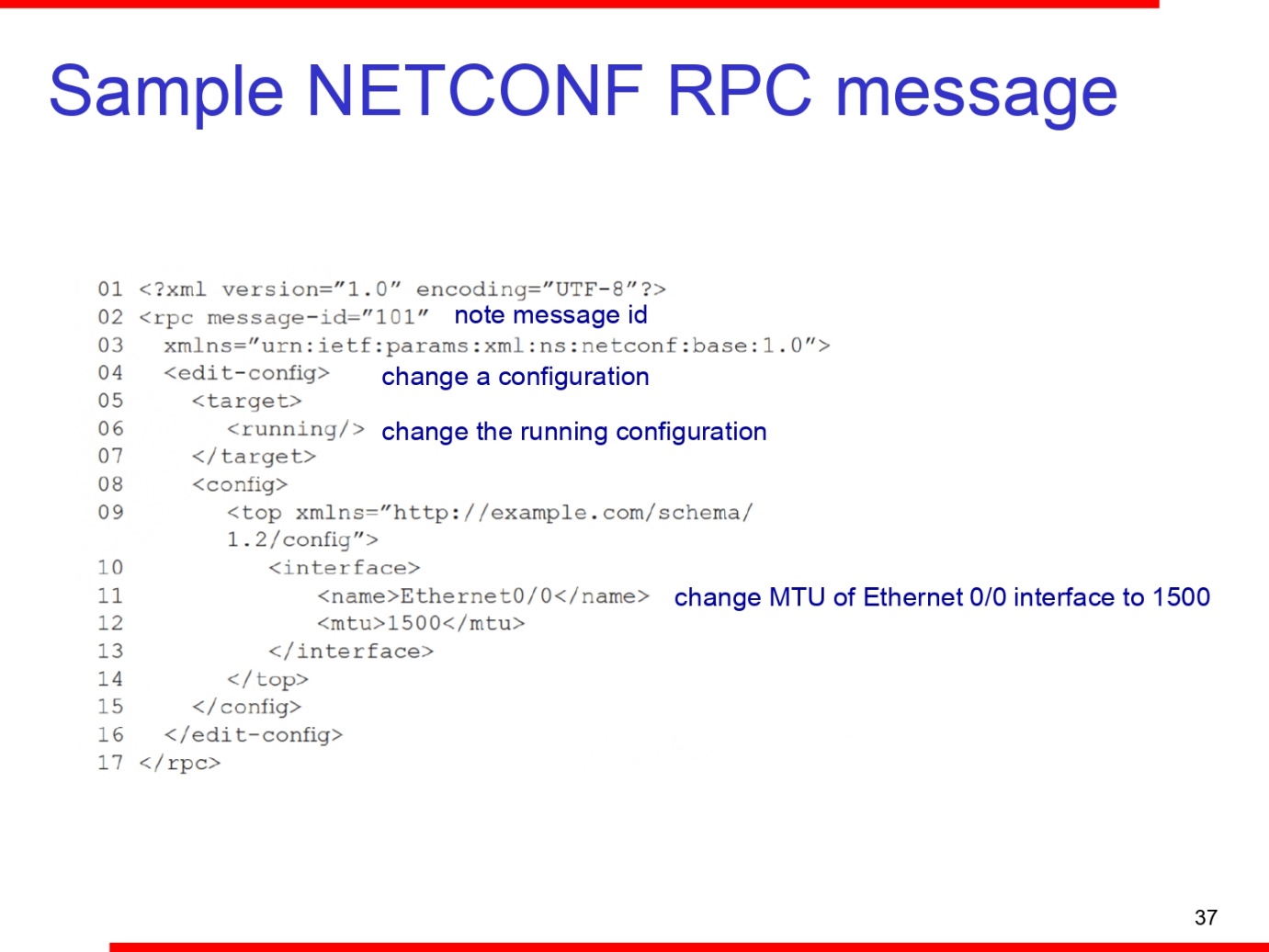
**Esempio di messaggio RPC NETCONF**

Immagine che contiene testo, schermata, design

Descrizione generata automaticamente**YANG**

I dati di XML vengono generati a partire da un linguaggio di definizione delle informazioni che prende il nome di YANG.

* Linguaggio di modellazione dei dati utilizzato per specificare struttura, sintassi, semantica dei dati gestionali della rete NETCONF
  + tipi di dati incorporati, come SMI
* Documento XML che descrive il dispositivo, le funzionalità possono essere generate dalla descrizione YANG
* Può esprimere vincoli tra i dati che devono essere soddisfatti da una configurazione NETCONF valida
  + garantire che le configurazioni NETCONF soddisfino i vincoli di correttezza e coerenza

27.09

**NETCONF vs RESTCONF**

RESTCONF è abbastanza simile a NETCONF, è una sua evoluzione.

NETCONF e RESTCONF sono due paradigmi nati per la configurazione della rete ma che possono essere utilizzati anche per recuperare informazioni sullo stato della rete.

La principale differenza tra i due paradigmi è il modo in cui interagiscono con l’agente remoto, mentre dal punto di vista dell’architettura di sistema abbiamo sempre un meccanismo di richiesta e risposta e un meccanismo di notifica.

In SNMP la notifica è chiamata **trap**, mentre in altri protocolli prende tipicamente il nome di **notifica** o **sottoscrizione**. Avremo quindi un’entità centralizzata che raccoglie i dati e per quanto riguarda NETCONF/RESTCONF si occupa anche di pushare le configurazioni.

Per quanto riguarda i protocolli utilizzati dai due paradigmi cambia solo il protocollo utilizzato per interfacciarsi con l’AGENT (protocollo a strato applicativo): NETCONF utilizza un protocollo RPC, quindi la ME (Managing Entity) è il client che chiede di eseguire la funzione su un server remoto, cioè l’AGENT che gira sul dispositivo (SNMP lavora allo stesso modo). Per quanto riguarda l’interfaccia/API orientata al paradigma REST, l’accento non è posto sull’esecuzione di una funzione remota, ma piuttosto sul modificare una risorsa remota tramite operazioni di tipo CRUD.

RESTCONF, essendo un protocollo di tipo REST, applicherà le 4 operazioni tipiche del REST, ovvero le operazioni CRUD (Create Read Update Delete), che vengono mappate uno a uno ai diversi metodi dell’HTTP.

**Operazioni CRUD**

C -> Create Metodo POST

R -> Read Metodo GET

U -> Update Metodo PUT

D -> Delete Metodo DELETE

Quindi un API realizzata seguendo i dettami del REST le operazioni fondamentali sono le 4 appena viste e vengono eseguite su risorse remote. L’API REST si appoggia su protocollo HTTP.

Anche l’RPC viene trasportato tipicamente sopra HTTP, solo che in quel caso HTTP è un contenitore del messaggio RPC, che viene formattato in XML, mentre nel caso del RESTCONF i metodi dell’HTTP vengono utilizzati per implementare le operazioni tipiche della manipolazione degli oggetti.

In RESTCONF, così come in NETCONF, posso cambiare la configurazione o monitorare i dispositivi.

RESTCONF fornisce ancora le funzionalità di astrazione del dispositivo di NETCONF.

Quindi la differenza principale tra NETCONF/RESTCONF e SNMP non è lo schema di base, in quanto entrambi dispongono dei metodi **Request/Response** e **Notify**, ma piuttosto il protocollo utilizzato: SNMP usa SNMP mentre sia RPC (NETCONF) sia REST (RESTCONF) utilizzano HTTP/HTTPS.

Un ulteriore differenza riguarda la memorizzazione dei dati:

* SNMP: i dati vengono memorizzati internamente in base al formato SMI che definisce la struttura dei dati in moduli composti da oggetti. Un modulo tipicamente è un’astrazione di livello superiore: può essere un’interfaccia, un protocollo o un pezzo di un nostro componente.
* NETCONF/RESTCONF: utilizzano il protocollo YANG come formalismo per la memorizzazione dei dati.

Per quanto riguarda il trasferimento invece:

* SNMP: utilizza il protocollo SNMP e i dati sono formattati secondo il formato ASN.1 o Abstract Syntax Notation 1;
* NETCONF/RESTCONF: utilizzano tipicamente il protocollo HTTP/HTTPS e i dati sono formattati attraverso XML oppure JSON.

Viste le considerazioni precedenti è evidente che NETCONF/RESTCONF utilizzano protocolli più maneggevoli.

In SNMP ho sempre a che fare con gli OID, che sono standardizzati ISO ma rimangono comunque una tecnologia proprietaria. In altre parole, mentre ci sono delle parti dell’albero che sono comuni a tutti, quindi per esempio tutti i dispositivi che utilizzano il protocollo UDP avranno gli stessi oggetti, ovvero un modulo comune UDP e gli oggetti che rappresentano i parametri UDP di SNMP. Che cosa succede se invece vado considerare un router della Cisco di uno specifico modello? I costruttori sono autorizzati ad aggiungere parti dell’albero delle gerarchie, ci saranno quindi delle parti “private” che dipendono dal singolo costruttore. Quindi ogni azienda manifatturiera è autorizzata, non solo ad utilizzare gli OID già definiti, ma ne può aggiungere di nuove; quindi, se per esempio Cisco vuole aggiungere un nuovo campo per il monitoraggio della rete è libra di farlo.

Quindi come fa un amministratore di rete se vuole visualizzare questo nuovo campo?

Deve disporre di una libreria aggiornata contenente gli OID di quel dispositivo di quel costruttore, in quanto il costruttore può decidere di aggiungere degli OID solo per alcuni modelli di dispositivo.

Di conseguenza per interpretare i codici presenti nei messaggi SNMP abbiamo necessariamente bisogno di avere le librerie degli OID aggiornate.

Il corpo dei messaggi SNMP è composto da una struttura che si ripete in cui sono presenti i campi TLV e OID.

Volendo astrarre a più alto livello: OID ci fornisce la semantica mentre il TLV ci dà la sintassi.

SNMP sfrutta un approccio non nuovissimo a differenza di NETCONF/RESTCONF che si appoggiano tipicamente al protocollo HTTP, risultando più maneggevoli e potenti. Nonostante esista quindi più software per questi ultimi, non esistono grandi basi di dati come per SNMP.

La cosa interessante del RESTCONF è la flessibilità nella formattazione dei dati, che non necessariamente devono essere memorizzati in formato XML, il quale risulta essere piuttosto verboso anche se facilmente parsabili dalla macchina e leggibile dall’uomo. È più semplice invece utilizzare il formato JSON quando si utilizza l’approccio RESTCONF: questo tipo di formato ci permette di modificare oggetti remoti tramite le 4 operazioni fondamentali CRUD del REST. Inoltre, il formato JSON, oltre ad essere più leggero risulta anche molto più leggibile in quanto non ha il meccanismo verboso dei tag di apertura e chiusura caratteristico di XML.

**RESTful paradigm**

Representational State Transfer (REST): insieme di principi architettonici per i sistemi distribuiti

* **Client-Server**: interazioni basate su un modello di comunicazione di richiesta-risposta
* **Interfacce uniformi**: interfaccia standard univoca e non ambigua per l'accesso alle risorse (ad esempio l'URI per l’interfaccia REST: in particolare per il caso HTTP viene utilizzato un particolare URI che è URL e ci dà la posizione all’interno del dispositivo di dove si trova la risorsa)
* **Transazioni Stateless**: ogni transazione è atomica, ovvero inizia e finisce con la transazione stessa. Il contesto e lo stato del client non vengono archiviati sul server
* **Cacheable**: i dati ricevuti possono essere memorizzati dai client per evitare di richiederle in continuazione
* **Sistema a livelli**: i componenti intermedi possono nascondere ciò che c'è dietro di loro. Un sistema può essere diviso in livelli, il livello frontale nasconde la complessità di tutti gli altri livelli (principio di base su cui si basano i sistemi web)

**Linguaggio JSON**

Un sistema REST può utilizzare i dati in formato JSON e XML. Il JSON è basato su una struttura chiamata Documento (che inizia e finisce con le parentesi graffe), al cui interno abbiamo una struttura chiave valore in cui al valore può essere associato un numero, una stringa, un vettore o altri json, con la possibilità di avere una struttura annidata e facilmente leggibile.

JavaScript Object Notation: formato di descrizione dei dati

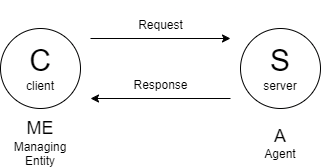
* Un file JSON è anche chiamato “documento”
* Formato leggero per lo scambio dei dati
  + facile da leggere e scrivere per gli esseri umani
  + facile da analizzare e generare per le macchine
  + il codice per l'analisi e la generazione di dati JSON è prontamente disponibile nella maggior parte dei linguaggi di programmazione => favorisce il sistema di integrazione e interoperabilità tra componenti software di terze parti
* Modello dati singolo (senza schema), molti casi d’uso
  + Un documento JSON è racchiuso tra parentesi {}
  + Ogni voce di dati è una coppia <chiave,valore>

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

Cosa accomuna questi due tipi di approccio (SNMP vs RESTCONF/NETCONF)?

Questi approcci sono non soltanto passivi, ma è anche possibile interrogare o farsi notificare dal sistema, statistiche di tipo aggregato, tramite uno schema Client-Server invertito.



Nel caso di Request-Response la ME (Managing Entity) è il client e A (Agent) è il server.

Immagine che contiene testo, schermata, cerchio, diagramma

Descrizione generata automaticamenteInvece nel caso in cui si utilizza il meccanismo di trap notify si invertono i ruoli, la ME assume prima il ruolo di Client e A di server quando si sottoscrive (subscribe) una notifica o si installa una trap su A. Da questo momento in poi si invertono i ruoli: ME diventa server e A diventa Client. Così avremo le Trap/Notifiche lanciate dall’agent verso la ME.

Sia SNMP che i paradigmi NETCONF/RESTCONF supportano entrambi le tipologie di comunicazione; quindi, di fatto i componenti software per il Network Management devono implementare sia la componente Client che la componente Server.

Quindi meccanismi simili, implicano limitazioni simili. Posso quindi ricevere informazioni di stato (principalmente binarie) oppure posso avere dei contatori aggregati. Entrambi questi approcci sono basati su una struttura che lega gli oggetti che devo modificare/monitorare. Quindi ho sempre un MIB (una base di dati) con una struttura dati che è nota a priori ed è l’unica utilizzabile, che posso andare a modificare in scrittura oppure ad interrogare in lettura. Come già detto, dal punto di vista numerico posso avere solo contatori aggregati o informazioni di stato semplici.

* Si usano ancora sia SNMP sia NETCONF sia RESTCONF, oltre ad altri modi per gestire i dispositivi. Come, ad esempio, l’accesso SSH è considerato parte del mondo NETCONF, poiché quando si entra con SSH su un dispositivo, si apre la configurazione, si cambia e poi si salva la modifica, che è equivalente a caricare un file in cui è indicata la modifica alla configurazione.

Vediamo un esempio pratico, andando a vedere come i contatori aggregati, quelli che del numero di byte per pacchetto, sono utilizzati attualmente dal GARR. Il GARR è la rete che unisce tutte le università italiane e gli enti di ricerca. È una rete particolarmente moderna che ha un equivalente in tutte le nazioni europee, e tutte le reti sono interconnesse tra loro attraverso una macrorete europea chiamata GÉANT. L’infrastruttura portante di questa rete è dell’odine del terabit al secondo (1 terabit = 1000 gigabit).

<https://gins.garr.it/info/xservice.php?id=267>

Immagine che contiene testo, Diagramma, linea, schermata

Descrizione generata automaticamente

SNMP ci fa vedere il profilo del grafico e tutta una serie di informazioni sulla salute dei dispositivi.

Immagine che contiene schermata, linea, diagramma, testo

Descrizione generata automaticamente

Questi collegamenti sono doppi perché rappresentano il nodo principale e il nodo di backup (l’università per stranieri e il CNR non hanno il nodo di backup).

Immagine che contiene testo, schermata, Policromia, Carattere

Descrizione generata automaticamente

**Riepilogo**

Network management:

* estremamente importante: 80% del “costo” della rete
* ASN.1 è la codifica per formattare i dati
* Protocollo SNMP come strumento per trasmettere informazioni

Network management: più arte che scienza (non esiste una regola universale per configurare un sistema di gestione)

* cosa misurare/monitorare?
* come rispondere ai problemi?
* come faccio a codificare il problema? come filtro gli allarmi?

**Questioni aperte**

* SNMP e protocolli simili possono collezionare solo statistiche aggregate basate su contatori (informazioni binarie e contatori)
* SNMP non è adatto per il monitoraggio in tempo reale
  + scale temporali grandi, ordine di secondi (a volte anche minuti)
* SNMP è di tipo passivo, non prevede monitoraggio attivo della rete

Esistono invece degli altri tool che permettono un monitoraggio attivo della rete:

* Ping, traceroute, nmap
* Misura/informazioni limitate

**Problemi aperti**

Con queste soluzioni sono possibili solo misurazioni passive

* SNMP e protocolli simili possono raccogliere solo dati aggregati statistiche basate su contatori
* Non è adatto per un vero monitoraggio in tempo reale (scale temporali grandi, ordine di secondi)

Utilizza strumenti di rete noti per fare misurazioni attive:

* Ping, traceroute (per capire la strada che fa), nmap (per capire le porte aperte sul sistema)
* Svantaggio: Misura/informazioni limitate

Una tecnica è attiva quando inietta del traffico in rete. Le tecniche attive servono per acquisire informazioni che con le tecniche passive non riesco a raggiungere.

Dispositivi più moderni permettono di creare programmi in C

Permettono di effettuare sia operazioni di osservazione sia di applicare delle azioni. Queste soluzioni di programmabilità del data playing?

I dispositivi che supportano questo schema costano 95.000 hanno senso solo in contesti enterprice. Esistono però soluzioni open source che permettono di effettuare operazioni molto simili e più o meno con le stesse prestazioni.

**Direzioni future**

Tutte le misure viste fino ad ora sono di tipo passivo, le informazioni catturate sono principalmente di tipo aggregate, e questa limitazione è dovuta alle basi di dati a cui si appoggia questo schema. Per superare questa limitazione devo utilizzare degli schemi di cattura che vadano oltre la cattura aggregata, lo step successivo al contare i pacchetti che passano su un’interfaccia, è il monitoraggio i flussi che passano sull’interfaccia.

Monitorare i flussi significa creare un record per tutte le connessioni TCP e tutti i flussi UDP, è chiaro che non esistono dei veri e propri flussi UDP però posso mappare una richiesta con una risposta, quindi posso creare una specie di meta-flusso.

ESEMPIO: Teams usa sempre le stesse porte mittente e destinazione per l’audio e il video anche se utilizza UDP, e quindi creo un flusso attraverso queste due porte.

Quindi se si vuole monitorare chi sta accedendo e non soltanto quanto traffico passa, devo trovare una soluzione che faccia un passo ulteriore, ovvero per che permetta di monitorare tutti i flussi che passano attraverso una interfaccia.

Questo chiaramente è interessante per i dispositivi di rete, come switch e router, poiché sui sistemi terminali non ci sono i flussi che attraversano l’interfaccia, ma solo flussi in entrata e in uscita, che tipicamente sono mappati con i servizi che girano sulla macchina, quindi nel caso di sistemi terminali, dobbiamo andare ad analizzare non tanto i flussi, quanto i log dei servizi che stanno girando sulla macchina.

Quindi ciò che è necessario è complementare i dati di SNMP con altri dati eterogenei e possibilmente con un maggiore contenuto informativo e una maggiore complessità.

Se voglio fare un ulteriore passo in avanti?

La cattura di un flusso infatti è un record che ci dice che il flusso è iniziato ad una determinata ora, l’indirizzo mittente e quello destinatario, i numeri di porte e numero di pacchetti e bytes. È quindi ancora un’informazione aggregata anche se più di dettaglio rispetto al numero di bytes che attraversano un’interfaccia. La si considera aggregata perché non ci dà una visione temporale di quello che sta succedendo.

Per avere dei dati completi anche da un punto di vista temporale devo fare una cattura di tipo TCPDUMP o WIRESHARK che mi permette di effettuare una cattura in tempo reale.

Come in SNMP, anche in questo caso i dati andranno memorizzati e trasferiti, in quanto siamo sempre in una architettura Client-Server in cui una parte produce dati e una li consuma. C’è quindi la necessità di fare particolare attenzione a quanti dati vado a generare per evitare che la rete venga sommersa dal traffico di gestione della rete stessa.

Tutte le tecniche visto fino ad ora: SNMP, statistiche sui flussi o cattura dei singoli pacchetti sono tutte tecniche di tipo passivo. Esistono però anche tecniche classificate come attive, nello specifico si parla di tecnica attiva quando questa inietta del traffico in rete, riuscendo ad acquisire informazioni che le tecniche passive non riescono ad acquisire.

Se per esempio sono interessato a capire se un sistema remoto, non sotto il mio controllo è attivo, posso eseguire un Ping. Se invece è sotto il mio controllo posso vedere tramite NETCONF e RESTCONF. Se voglio capire che strada fa una richiesta tra mittente e destinatario posso usare traceroute, se voglio conoscere le porte di un determinato sistema, posso utilizzare NMAP.

Tuttavia, questi tool attivi, tipicamente hanno lo svantaggio che le informazioni fornite sono comunque limitate, esistono quindi degli altri tool ancora più sofisticati che riguardano la programmazione attiva della rete. Nello specifico questi dispositivi più moderni permettono creare dei programmi in C ed iniettarli nella macchina, facendoli eseguire in una sandbox del kernel. Questi programmi permettono di effettuare sia operazioni di osservazione, ovvero monitorare statistiche del traffico particolari, sia di applicare delle azioni.

Riprendendo lo scopo del Network Management: osservare, analizzare e agire; queste soluzioni di programmabilità ci permettono di automatizzare alcuni processi di osservabilità e la reazione ad eventuali situazioni di problemi o malfunzionamenti.

Per quanto riguarda l’analisi invece, non possiamo pensare di svolgere un’analisi completa utilizzando un programmino in C. L’analisi va comunque fatta online o offline, dall’umano o automatizzata. Posso quindi osservare le statistiche e se trovo un problema che sono in grado di risolvere tramite configurazione, utilizzerò i metodi standard, se invece de fare un’operazione più sofisticata posso pensare di pushare nel dispositivo un ulteriore software; è ovvio che un software di data programmability può aiutare nel caso di load balancing, a dirottare il traffico, a reagire ad un attacco hacker ma se una ventola del router si è bloccata, non ci si può fare niente, quindi non è una soluzione per tutti i problemi.

Telemetria di rete è solo passiva, quindi utile solo per l’osservabilità, anche se esistono degli schemi che permettono di renderla attiva. Si parla di tecnica attiva perché per effettuare questa osservazione in modo più dettagliato si attaccano delle etichette sui pacchetti e facendo ciò si va modificare il traffico, si può quindi parlare di schemi attivi. D’altro canto, questa tecnica ha il problema che tende a creare volumi di traffico estremamente rilevante poiché va a creare traffico per ogni dispositivo che il pacchetto attraversa. Vengono quindi utilizzati solo quando estremamente necessario, per evitare di intasare la rete con il traffico di management, perché permettono delle operazioni che altrimenti non sarebbero possibili.

Misure passive:

* Basate sull'acquisizione e sull'analisi del traffico
* Monitoraggio basato sul flusso: NetFlow
* Campionamento basato sul flusso: sFlow

Misurazione attiva:

* Telemetria di rete

Tutti richiedono un ampio supporto da parte dei dispositivi