PARTE 11

MONITORAGGIO DELLE RETI

Modulo 1:
Protocollo SNMP

SNMP

- **SNMP**
  - Simple
  - Network
  - Management
  - Protocol
- **Necessario introdurre alcuni concetti sulla gestione delle reti**
- **Focus su reti complesse e di grandi dimensioni**

Cosa si intende con network management

- **Gestione delle configurazione**
  - Tenere traccia dei dispositivi in una rete e delle loro funzioni
- **Gestione dei problemi**
  - Identificare problemi
  - Predisporre soluzioni temporanee o definitive
- **Gestione delle prestazioni**
  - Identificazione dei colli di bottiglia
  - Capacity planning

Requisiti del network management

- **Necessaria interfaccia di gestione**
  - Standardizzata
  - Estendibile
  - Portabile
- **Necessari meccanismi di gestione**
  - Poco costosi
  - Che non richiedono hardware specifico

Aree funzionali del network mgm

- **Configuration management**
  - Inventario, configurazione
- **Gestione problemi (Fault)**
  - Reattivo e proattivo
- **Gestione prestazioni**
  - Definizione KPI
  - Es: perdita pacchetti, timeout, RTT, collisioni, ...
- **Account management**
  - Gestione costi

Aree funzionali del network mgm

- **Asset management**
  - Statistiche su uso dispositivi, personale, spazi
- **Planning management**
  - Analisi trend
  - Scenari what if
  - Legato a gestione prestazioni

SNMP & Network Management History

- 1983 - TCP/IP replaces ARPANET at U.S. Dept. of Defense, effective birth of Internet
- First model for net management - HEMS - High-Level Entity Management System (RFCs 1021,1022,1024,1076)
- 1987 - ISO OSI proposes CMIP - Common Management Information Protocol, and CMOT (CMIP over TCP) for the actual network management protocol for use on the internet
- Nov. 1987 - SGMP - Simple Gateway Monitoring protocol (RFC 1028)
- 1989 - Marshall T. Rose heads up SNMP working group to create a common network management framework to be used by both SGMP and CMOT to allow for transition to CMOT

SNMP & Network Management History

- Aug. 1989 - “Internet-standard Network Management Framework” defined (RFCs 1065, 1066, 1067)
- Apr. 1989 - SNMP promoted to recommended status as the de facto TCP/IP network management framework (RFC 1098)
- June 1989 - IAB committee decides to let SNMP and CMOT develop separately
- May 1990 - IAB promotes SNMP to a standard protocol with a recommended status (RFC 1157)
- Mar. 1991 - format of MIBs and traps defined (RFCs 1212, 1215)
- TCP/IP MIB definition revised to create SNMPv1 (RFC 1213)

Versioni

- **Tre versioni principali**
  - SNMPv1, SNMPv2, SNMPv3
- **SNMPv1 standard**
- **SNMPv2 diviso in:**
  - SNMPv2u - SNMPv2 con user-based security
  - SNMPv2* - SNMPv2 con user-based security e feature aggiuntive
  - SNMPv2c - SNMPv2 senza sicurezza
- **SNMPv3**
  - Maggiore attenzione alla sicurezza

Cosa è SNMP

- **Protcollo per la gestione di elementi di rete**
  - Gestione locale e remota
  - Supporto per molteplici dispositivi
- **Due ruoli**
  - Agent (server) che gira sui dispositivi da gestire
  - Manager (client) che interpella gli agenti per raccogliere informazioni e dare comandi

Vantaggi di SNMP

- **Standardizzazione**
- **Supporto universale**
- **Estensibile**
- **Portabile**
- **Consente accesso per il management distribuito**
- **Protocollo leggero**

Approccio SNMP

- **Client pull**
  - Il client (manager) interroga i server (agent) per la raccolta di informazioni
- **Server push**
  - I server (agent) possono segnalare anomalie contattando i manager (client) mediante messaggi di tipo trap

Porte e protocollo di trasporto

Uso di UDP per il trasporto

Uso di due well-known port

- **UDP Port 161** - SNMP Messages
- **UDP Port 162** - SNMP Trap Messages

Parti di SNMP

- **Protocollo SNMP**
  - Definizione dei messaggi scambiati tra agent e manager
  - Specifica operazioni:
    Get, GetNext, Set, Trap
- **Structure of Managment Information (SMI)**
  - Regole per la specifica del formato usato per definire gli oggetti cui accede SNMP
- **Management Information Base (MIB)**
  - Mappa gerarchica degli oggetti gestiti e a cui è possibile accedere

Tipi di nodi

- **In una rete SNMP ci sono vari tipi di elementi chiamati nodi**
- **I nodi sono di diversi tipi**
- **Nodi gestiti**
  - Tipicamente ospitano un agent SNMP
- **Nodi gestori**
  - Tipicamente ospitano un manager SNMP
- **Nodi non gestiti**
  - Nodi che non supportano SNMP, ma possono essere gestiti mediante un proxy che gira su un altro nodo
- **I nodi possono essere di più di un tipo contemporaneamente**

Community names

- **Community**
- **Definite per indicare i destinatari di un messaggio SNMP**
- **Simili a domini Windows**
- **Gli agenti appartengono a una o più comunità**
- **I manager ricevono trap da una o più comunità**

Tipi di agenti SNMP

- **Due tipi di agenti**
- **Estensibili**
  - Aperti, modulari
  - Consentono di definire nuovi tipi di dati e di operazioni
- **Monolitici**
  - Non estensibili
  - Ottimizzati per una specifica piattaforma HW/SW

Agenti proxy e gateway

- **Consentono di estendere le operazioni di management verso**
  - Gestione di dispositivi che non supportano nativamente SNMP
  - Integrazione di agenti non-SNMP
  - Integrazione di manager non-SNMP
  - Filtraggio (firewalling) SNMP
  - Traduzione SNMPv1/SNMPv2/SNMPv2
  - Aggregazione di più agenti/monitor

Operazioni SNMP

- **Get**
  - Recupera un valore da una MIB in un agente remoto
- **GetNext**
  - Recupera valore successivo navigando dentro a una MIB
- **GetBulk**
  - Get di elementi multipli da una MIB
- **Set**
  - Imposta un valore in una variabile di una MIB
- **Trap**
  - Notifica asincrona da un agent a un manager

Trap SNMP

- **Esmpi di eventi che generano trap**
  - Power failure, Rottura disco, software failure, problema di rete, ...
- **Si può limitare il massimo rate di trap generate da un agente (throttling)**
- **Ogni trap ha una priorità associata**
  - Critica, Major, Minor, Warning, ...

Tipiche reazioni a un trap

- **Interrogazione per avere informazioni maggiori**
- **Invio di mail di notifica a gli amministratori**
- **Logging**

Linguaggi definiti in SNMP

- **Structure of Management Information (SMI)**
  - Definisce la struttura dei dati acceduti via SNMP
- **Abstract Syntax Notation 1 (ASN.1)**
  - Definisce i messaggi e le MIB in formato privo di ambiguità
- **Basic Encoding Rules (BER)**
  - Codifica i messaggi SNMP in formato che network-friendly

SMIv1

- **Riferimento:**
  - RFC 1155, 1212, 1215
- **Gli RFC descrivono:**
  - Come si definiscono le MIB
  - Il sottoinsieme di ASN.1 che si usa per le MIB
  - Un nuovo tipo di dato legato alle applicazioni
  - La parte standard dell'albero delle MIB
  - Definizione e descrizione di oggetti gestiti mediante SNMP

SMIv2

- **Riferimento:**
  - RFC 1442, 1443, 1444
- **Gli RFC descrivono**
  - Compatibilità con SMIv1
  - Definizione di un nuovo tipo (Counter64)

ASN.1

- **Abstract Syntax Notation One**
- **Sintassi simiare a quella del C**
- **Esempio di sintassi**

```
-- two dashes is a comment
-- The C equivalent is written in the comment
MostSevereAlarm ::= INTEGER           -- typedef MostSevereAlarm int;
circuitAlarms MostSevereAlarm ::= 3    -- MostSevereAlarm circuitAlarms = 3;
MostSevereAlarm ::= INTEGER (1..5)  -- specify a valid range
ErrorCounts ::= SEQUENCE {
    circuitID       OCTET STRING,
    erroredSeconds        INTEGER,
    unavailableSeconds  INTEGER
}    -- data structures are defined using the SEQUENCE keyword
```

BER

- **BER: Basic Encoding Rules**
- **Rapporto tra ASN.1 e BER**
  - ASN.1 → codice sorgente
  - BER → eseguibile
- **I messaggi SNMP sono convertiti in formato BER**
- **Messaggio BER:**
  - Label
  - Lunghezza
  - Valore

Label BER

- **1 byte composto da:**
  - Classe (2 bit)
    - 00 → universale
    - 01 → applicazione
    - 10 → dipendente dal contesto
    - 11 → privata
  - Formato (1 bit)
    - 0 → tipo di dati semplice
    - 1 → tipo di dato strutturato
  - Numero (5 bit): tipo di dato

    Es. classe universale, tipo semplice, numero 2 → intero

Lunghezza BER

- **Se bit più significativo = 0**
  - I rimanenti bit contengono la lunghezza dei dati
- **Se bit più significativo = 1**
  - I rimanenti bit indicano la lunghezza del campo lunghezza
  - La lunghezza vera viene memorizzata nei byte successibi al primo

Tipi di dato SNMP

INTEGER -- signed 32-bit integer

OCTET STRING

OBJECT IDENTIFIER (OID)

NULL -- not actually data type, but data value

IpAddress -- OCTET STRING of size 4, in network byte order (B.E.)

Counter -- unsigned 32-bit integer (rolls over)

Gauge -- unsigned 32-bit integer (will top out and stay there)

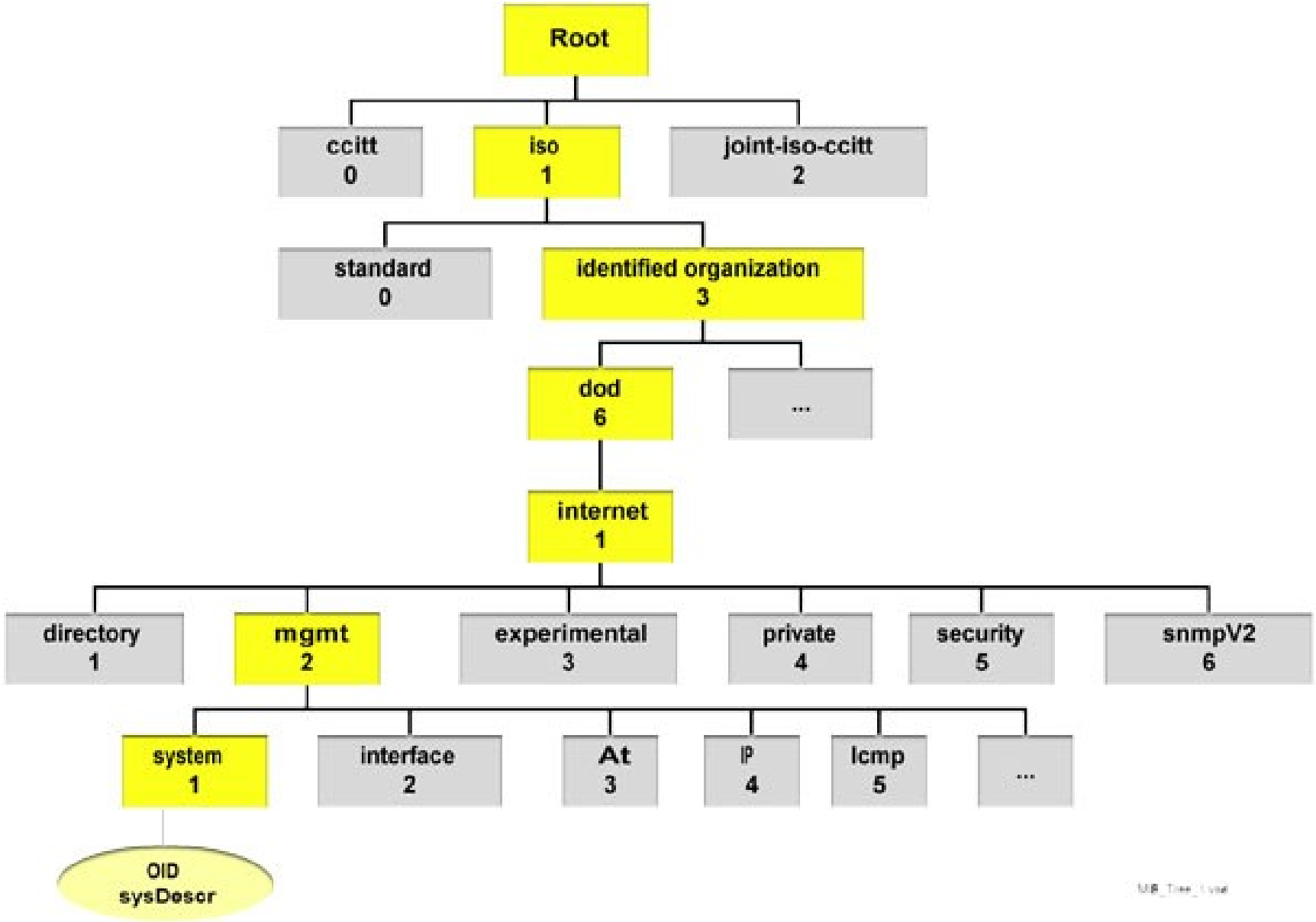
Tipi di dato SNMP

TimeTicks -- unsigned 32-bit integer (rolls over after 497 days)

Opaque -- used to create new data types not in SNMPv1

DateAndTime, DisplayString, MacAddress, PhysAddress, TimeInterval, TimeStamp, TruthValue, VariablePointer -- textual conventions used as types

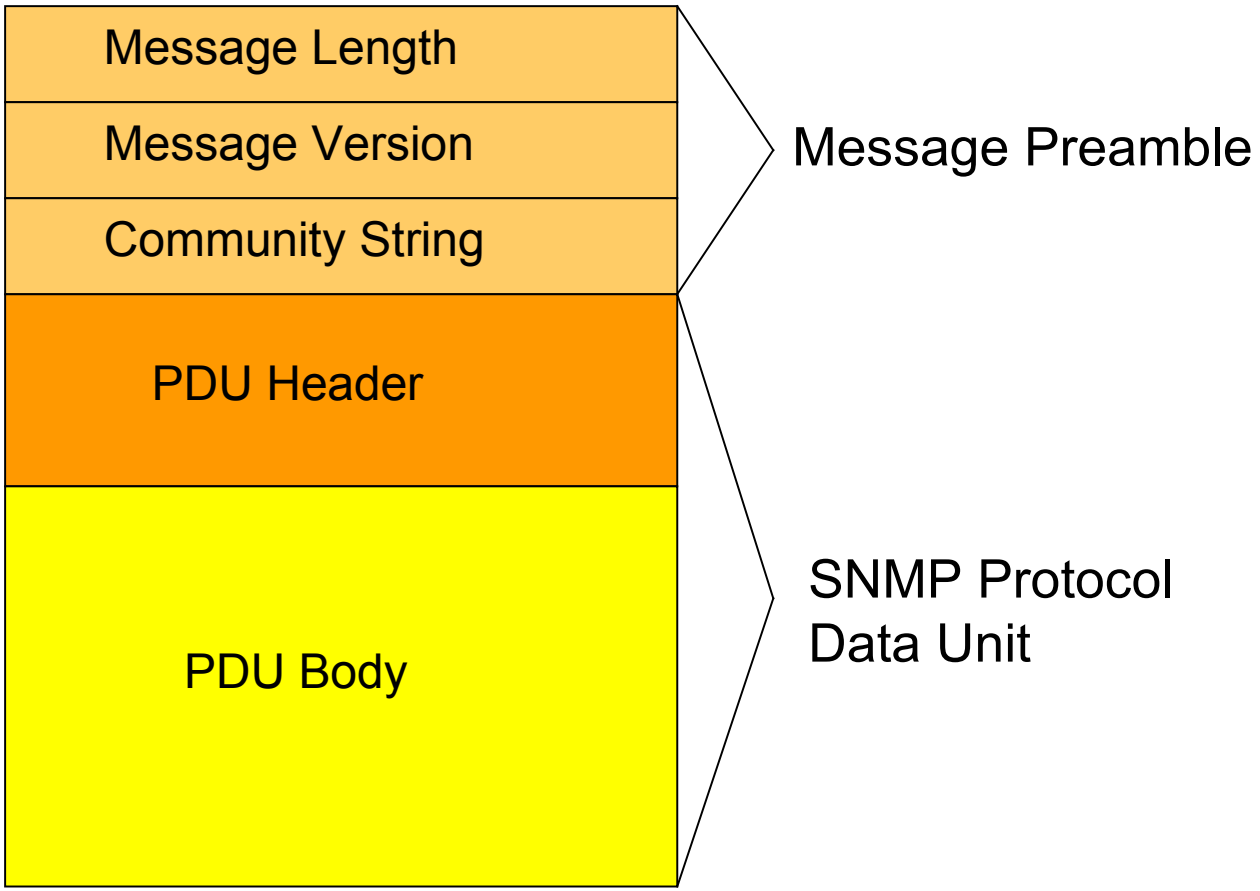
Albero MIB


Root
ccitt
0
iso
1
joint-iso-ccitt
2
standard
0
identified organization
3
dod
6
...
internet
1
directory
1
mgmt
2
experimental
3
private
4
security
5
snmpV2
6
system
1
interface
2
At
3
IP
4
Icmp
5
...
OID
sysDescr

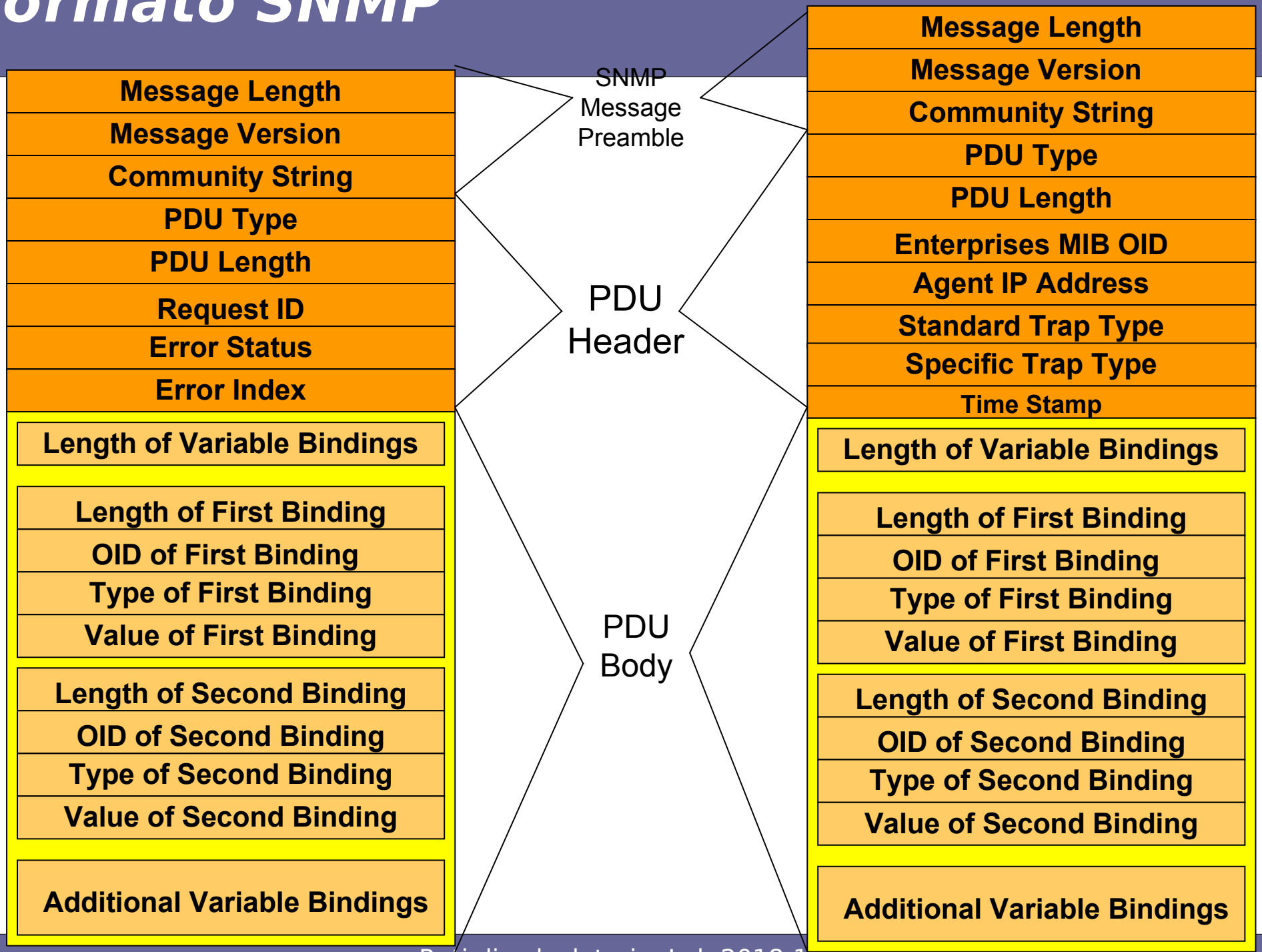
Esempio di MIB

```
sysContact OBJECT-TYPE
-- OBJECT-TYPE is a macro
      SYNTAX  DisplayString (SIZE (0..255))
      ACCESS  read-write
-- or read-write, write-only, not-accessible
      STATUS  mandatory
-- or optional, deprecated, obsolete
      DESCRIPTION
"Chris Francois
  cfrancois@acm.org
  (360)650-0000"
      ::= { system 4 }
```

Formato base del messaggio SNMP

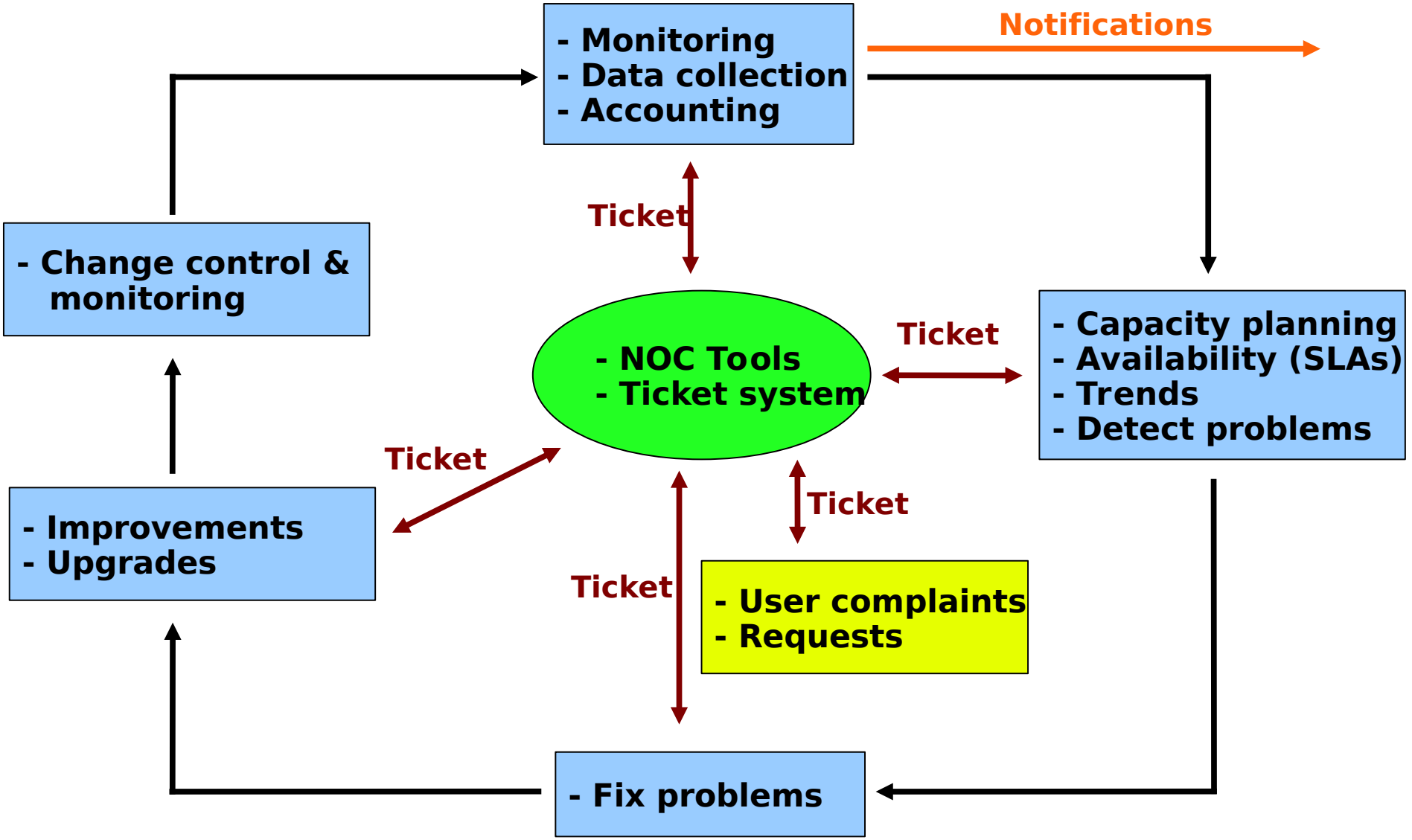


Formato SNMP



Modulo 2: Alcuni strumenti per il network management

Visione d'insieme



Strumenti di amministrazione

Availability

- Nagios — Services, servers/routers/switches. Alerting, availability reports

Utilisation

- Cacti — Total traffic, port usage, CPU, RAM, Disk, processes

Funzioni dei tool

- **Data collection**
- **Data storage**
- **Data visualisation**
- **Configuration**
  - e.g. what to monitor and how
- **Additional functionality**
  - e.g. alerting via E-mail

Data collection: SNMP

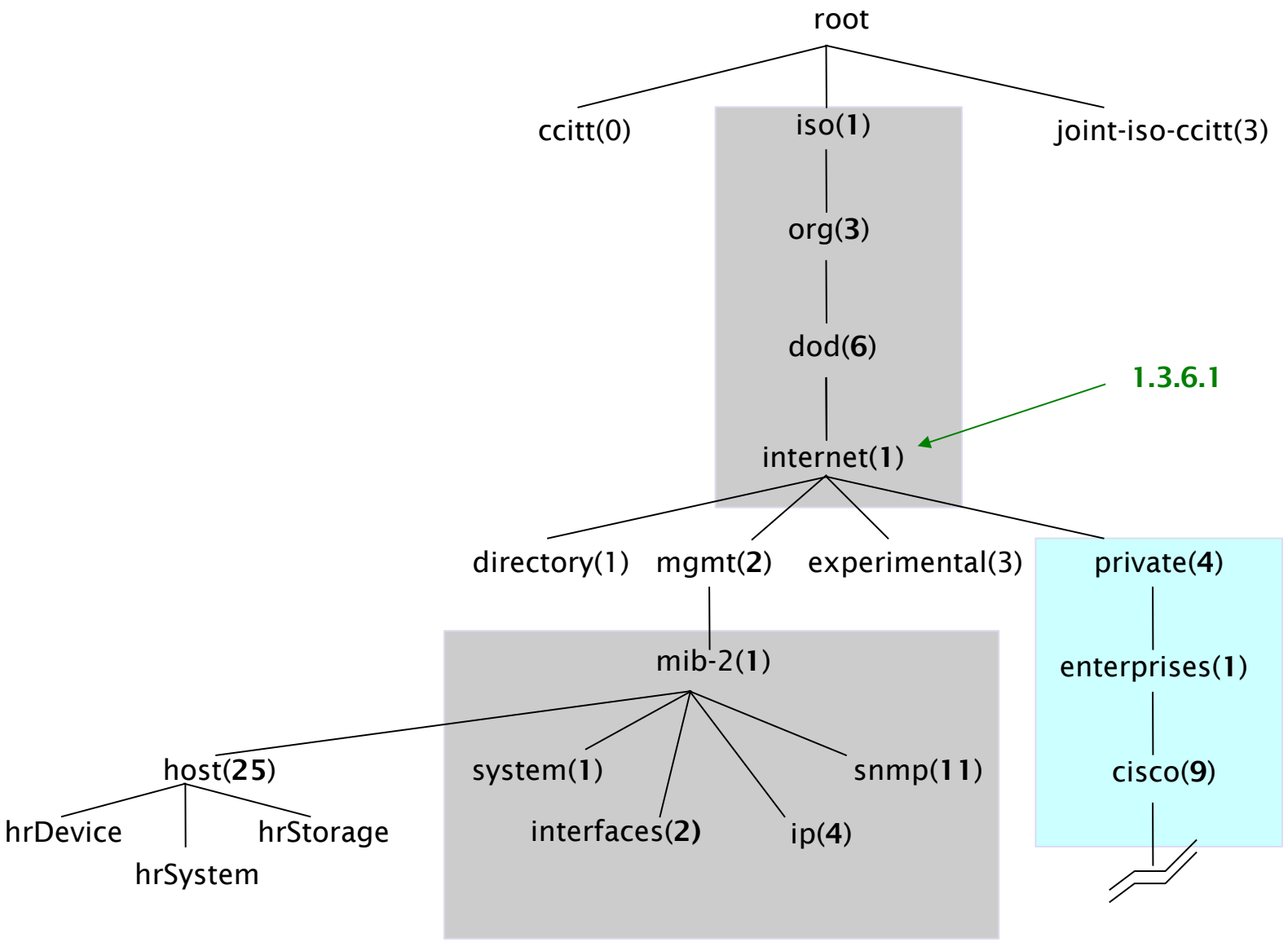
SNMP – Simple Network Management Protocol
- Industry standard, hundreds of tools exist to exploit it
- Present on any decent network equipment
- There are SNMP agents for Unix and Windows servers

Query – response based: **GET / SET**
- GET is mostly used for monitoring

Each piece of information is identified by a numeric Object Identifier or ”OID”
- Forms a unique key within any particular device

A collection of related OIDs is called a MIB (Management Information Base)

The MIB Tree


root
ccitt(0)
iso(1)
joint-iso-ccitt(3)
org(3)
dod(6)
1.3.6.1
internet(1)
directory(1)
mgmt(2)
experimental(3)
private(4)
mib-2(1)
enterprises(1)
host(25)
system(1)
snmp(11)
cisco(9)
hrDevice
hrStorage
interfaces(2)
ip(4)
hrSystem

OIDs and MIBs

- Navigate tree downwards
- OIDs separated by '.'
  - `1.3.6.1.2.1.1.5`
- Text labels
  - `.iso.org.dod.internet.mgmt.mib-2.system.sysName`
- Usually the end label is unique by itself
  - `.1.3.6.1.2.1.1.5 => sysName`
- MIB files resolve labels to OIDs and vice versa
  - only OIDs are actually sent over the wire

Testing SNMP by hand

Useful for debugging

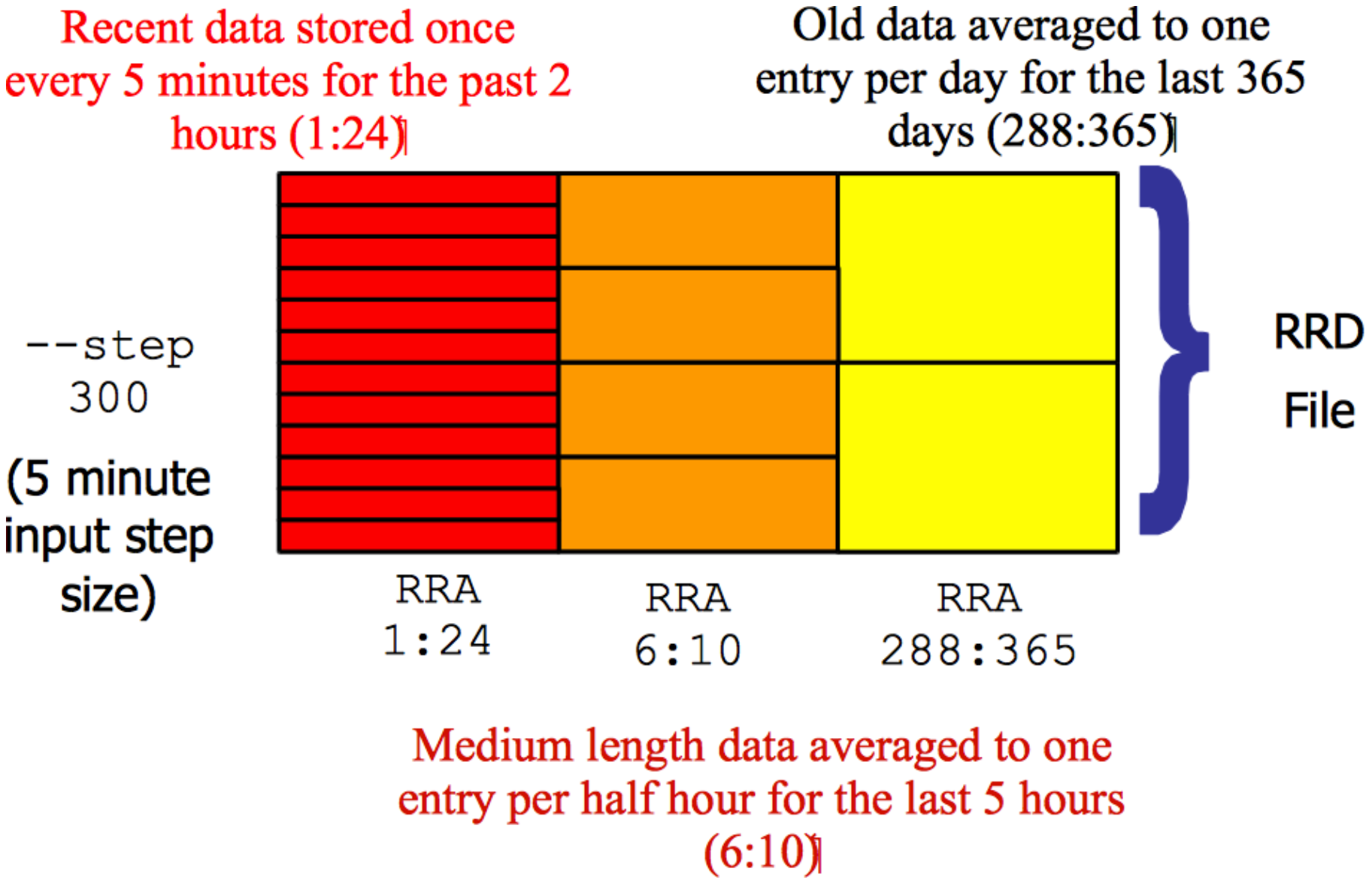
- `snmpstatus -c public -v2c 10.10.0.254`
- `snmpget -c public -v2c 10.10.0.254 sysUptime.0`
- `snmpwalk -c public -v2c 10.10.0.254 ifDescr`

(Note: SNMP won't respond if community string is wrong)

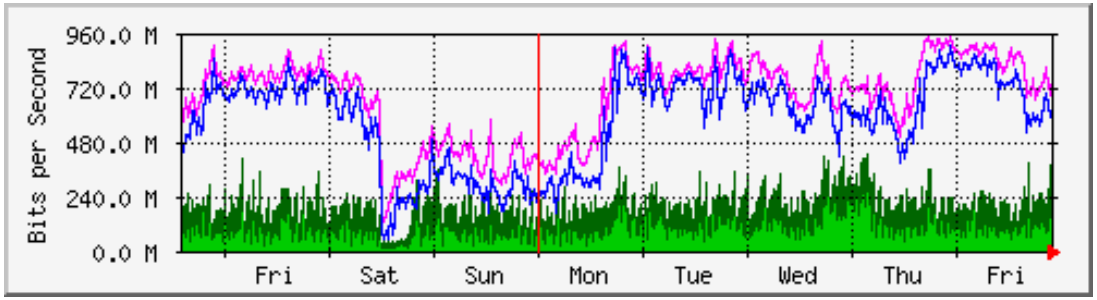
Storage and visualisation: rrdtool

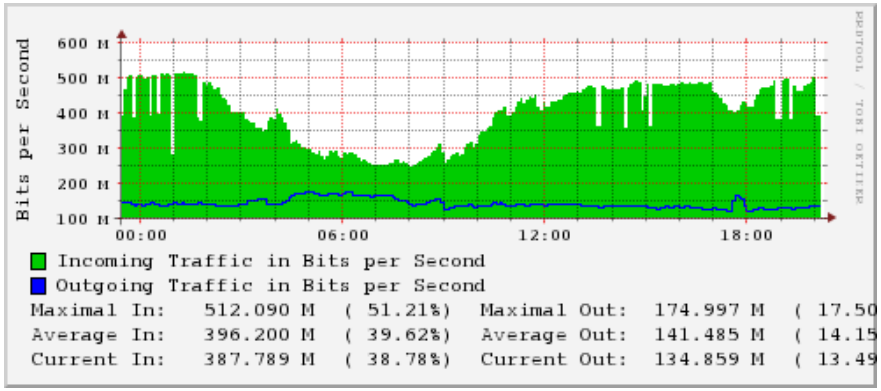
- **Has become the de-facto method of storing time-sequence data**
- **Data written in a “round-robin” file**
- **rrd files are of *fixed size***
- **As newer data is entered, older data is *consolidated* to make space**
  - so older data has lower resolution
- **Hugely flexible API for generating graphs**

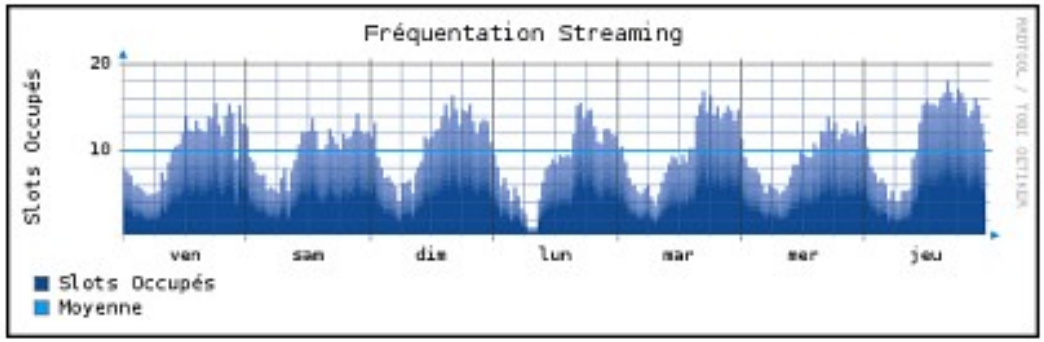
Example RRD file

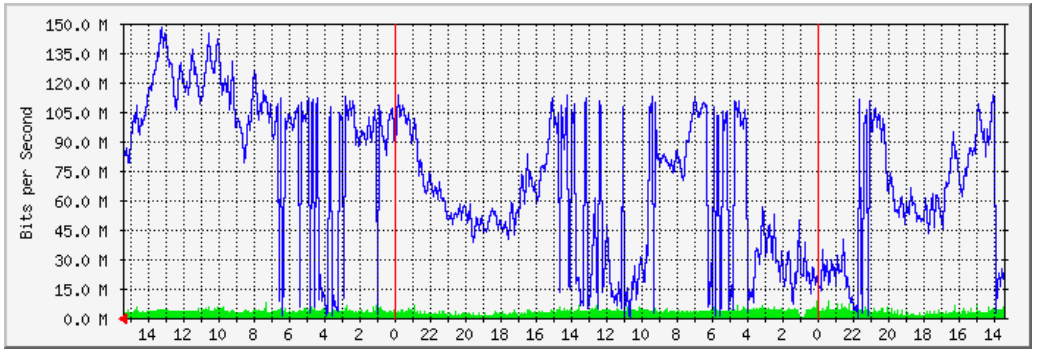


What it looks like…









Nagios

- Periodically tests hosts and services for availability
- Sends alerts and/or triggers event handlers
- Logs history, generates SLA reports

Nagios architecture

- Data collection: “Nagios Plugins”
  - Small, self-contained applications which make a single connection to test a service then quit
  - Return OK, Warning, Critical or Unknown
  - Many plugins supplied, even more available
  - Easy to write your own
- Data storage: plain text files
- Data visualisation: CGI web interface
- Configuration: plain text files

Pre-installed plugins in Ubuntu

/usr/lib/nagios/plugins

check_apt	check_file_age	check_jabber	check_nntp	check_procs	check_swap
check_bgpstate	check_flexlm	check_ldap	check_nntps	check_radius	check_tcp
check_breeze	check_ftp	check_ldaps	check_nt	check_real	check_time
check_by_ssh	check_host	check_linux_raid	check_ntp	check_rpc	check_udp
check_clamd	check_hpjd	check_load	check_ntp_peer	check_rta_multi	check_ups
check_cluster	check_http	check_log	check_ntp_time	check_sensors	check_users
check_dhcp	check_icmp	check_mailq	check_nwstat	check_simap	check_wave
check_dig	check_ide_smart	check_mrtg	check_oracle	check_smtp	negate
check_disk	check_ifoperstatus	check_mrtgtraf	check_overcr	check_snmp	urlize
check_disk_smb	check_ifstatus	check_mysql	check_pgsql	check_spop	utils.pm
check_dns	check_imap	check_mysql_query	check_ping	check_ssh	utils.sh
check_dummy	check_ircd	check_nagios	check_pop	check_ssmtp	

/etc/nagios-plugins/config

apt.cfg	disk-smb.cfg	ftp.cfg	ldap.cfg	mysql.cfg	ntp.cfg	radius.cfg	ssh.cfg
breeze.cfg	dns.cfg	hppjd.cfg	load.cfg	netware.cfg	pgsql.cfg	real.cfg	tcp_udp.cfg
dhcp.cfg	dummy.cfg	http.cfg	mail.cfg	news.cfg	ping.cfg	rpc-nfs.cfg	telnet.cfg
disk.cfg	flexlm.cfg	ifstatus.cfg	mrtg.cfg	nt.cfg	procs.cfg	snmp.cfg	users.cfg

Nagios core state machine

- Nagios schedules the checks to run evenly over the monitoring interval (e.g. 5 minutes)
- When a plugin returns Warning or Critical, Nagios enters a “soft” error state
- After a certain number of re-checks, enters a “hard” error state. At this point an alert is sent
- Repeated state changes enter “flapping” state which suppresses further notifications
- Designed to limit the day-to-day noise

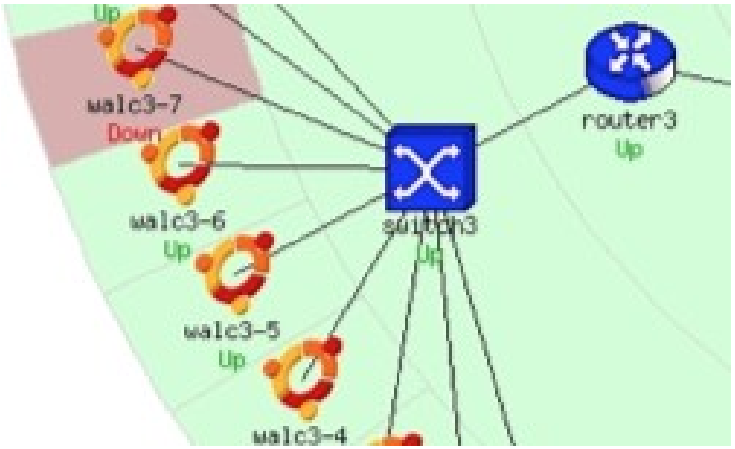
Hosts and services

- Generally we are interested in checking *services*
- Services run on *hosts*
- Nagios checks both hosts and services
- If a host fails, it’s smart enough to send you one notification for the host, rather than separate notifications for each service on that host

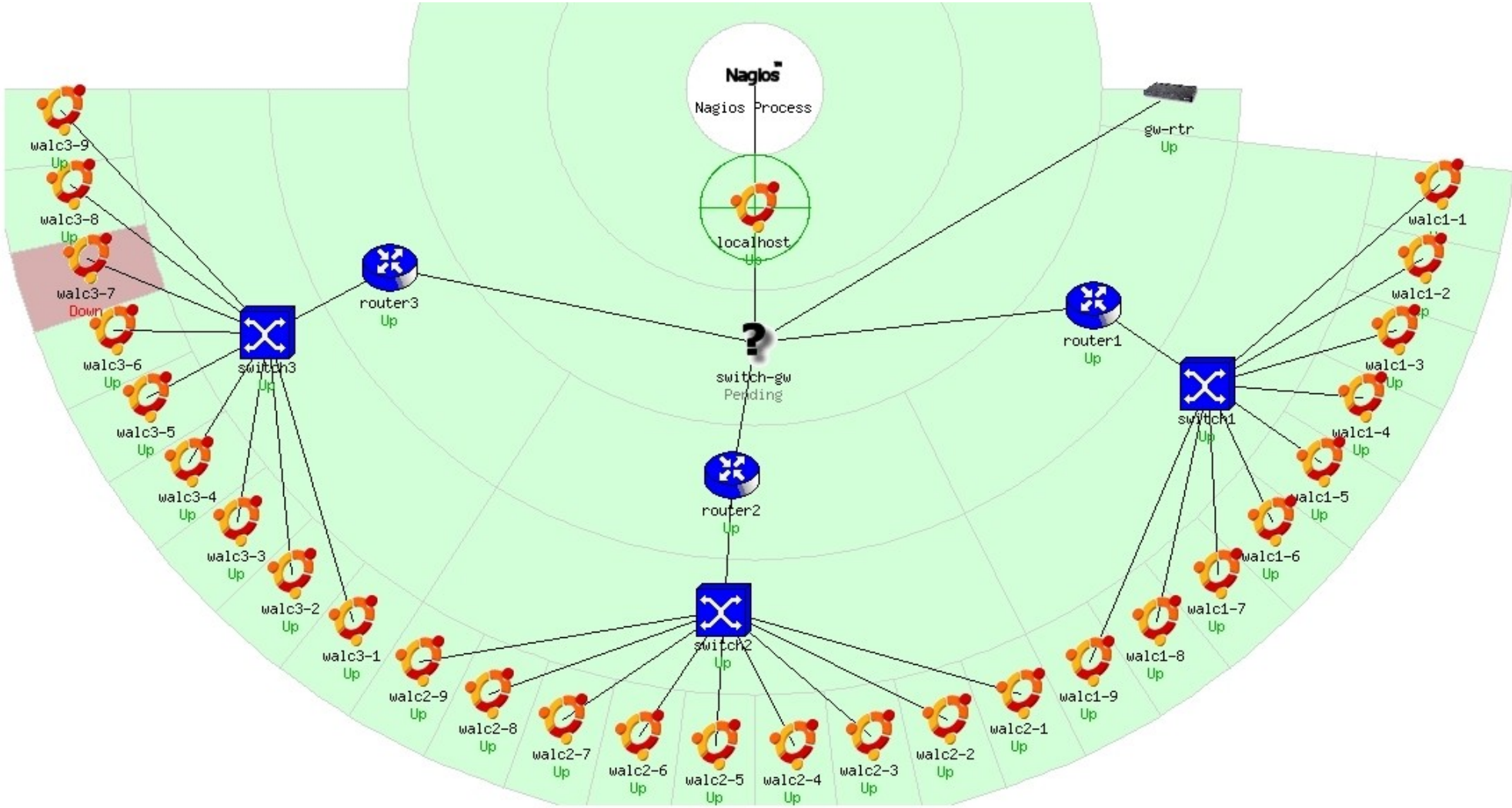
The concept of “parents”

Hosts can have parents:

- The parent of a PC connected to a switch would be the switch.
- Allows us to specify the dependencies between devices.
- Avoids sending alarms when parent does not respond.
- A node can have multiple parents (dual homed).



Network viewpoint



Cacti

- A tool to monitor, store and present network and system/server statistics
- Designed around RRDTool with a special emphasis on the graphical interface
- Almost all of Cacti's functionality can be configured via the Web.
- Acts as portal: let customers see their own graphs

Cacti architecture

1. Cacti is written as a group of PHP scripts.
2. The key script is “poller.php”, which runs every 5 minutes (by default). Data can be collected using SNMP or via PHP scripts.
3. Cacti uses RRDtool to store data on disk and create graphs for each device. You can configure them from within the Cacti web interface.
4. User configuration data is stored in a MySQL database
5. Configuration of what types of data to collect and how to graph them is stored in template files (XML)
6. Cacti Plugin Architecture allows Cacti functionality to be extended

Adding graphs to Cacti

1. Add a **Device** with the right **Data sources**
2. Create **Graphs** for that device
3. Add the graphs to **Graph Trees**

Simple when you've done it a few times

Tedious if you have lots of devices to add