


Communication protocols and network safety

Network control and management

Network management

- What is network management??
Why is it needed?



Boiler Operator Jeff Craig sits in the Boiler Room and monitors flow, temperature, and pressure of the boilers and feed-water system. Photo by Ryan Solomon

Mani Subramanian, *Network Management: An introduction to principles and practice*, Addison Wesley Longman, 2000

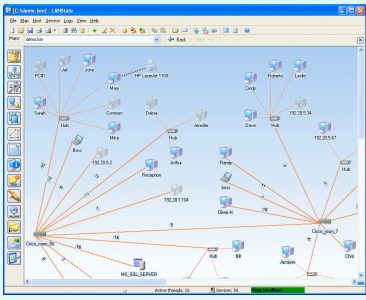
Network management

- Growth of internet and local networks caused small networks to connect into one **LARGE** infrastructure. With it increased the need for **SYSTEMATIC** management of hardware and software components of this system. Frequent questions:
 - Which resources are available in the network?
 - How much traffic is traveling through a certain network equipment?
 - Who uses network connections that cause their director to receive his email too slowly?
 - Why cant I send data to a certain computer?
- Definition: Managing a network involves **deployment, integration and coordination** of hardware, software and human resources for the purpose of **observation, testing, configuration, analysis and control** of network resources, for which we want to provide **operation** in real-time (or operation with appropriate quality - QoS) at an affordable price.

Examples of management activities

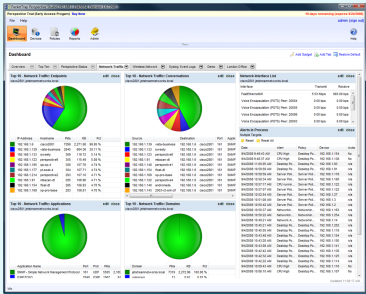
1. **detection of errors on the computer or router interface:** administrator can be notified by the software that the interface has a problem (even before it fails!)
2. **controlling computer operation and network analysis**
3. **controlling network traffic:** administrator can observe frequent communications and direction finding bottlenecks,
4. **detection of rapid changes in routing tables:** this phenomenon may indicate problems with routing or error in the router,
5. **controlling levels of service provision:** network service providers are able to guarantee availability, latency and certain service throughput; administrator can measure and verify,
6. **intrusion detection:** administrator can be notified if certain traffic arrives from suspicious sources; he can also detect a particular type of traffic (eg, a set of SYN packets intended for one single interface)

Examples of activities



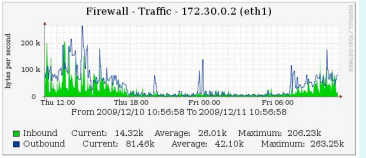
controlling computer operation and network analysis (detection of network topology)

Examples of activities



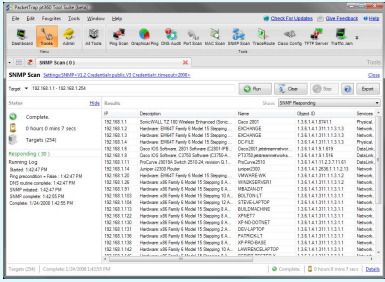
controlling network traffic (profiling)

Examples of activities



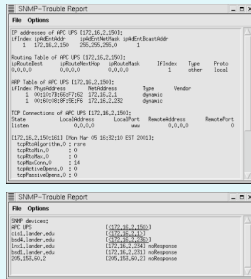
controlling the level of service provision (data flow)

Examples of activities



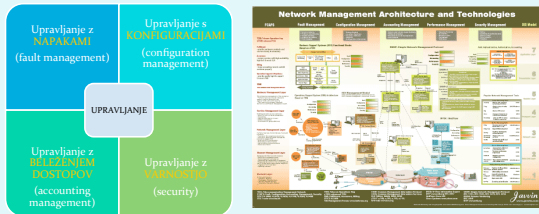
controlling computer operation and network analysis (list of IP addresses)

Examples of activities



controlling computer operation and network analysis (diagnostics and fault detection)

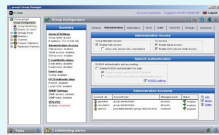
Areas of management



Management software

- CLI (Command Line Interface):
 - ✓ precise control,
 - ✓ possibility of using command lines (batch),
 - problem of syntax knowledge, storage configurations difficulty, less general – specific to a particular network equipment
- GUI (Graphical User Interface) applications:
 - ✓ visually beautiful, provides an overview of the whole system/network, uses its own (concise) protocol to communicate with a device – speed,
 - we lose the ability of readable configuration storage (binary), it can mask all configuration options

```
login net admin
admin@192.168.2.151's password:
CLI version 1.0
Available commands:
autoshare - Test autoshare config
passwd    - Change my administrator password
radius   - Radius Service
reset    - Reset device to defaults
shell    - Start system shell
show     - Show device configuration
status   - Show device status
quit     - Exit CLI
cli>
```



Management infrastructure

Management system components:

- operator = entity (application + human), BOSS,
- controlled device (contains NMA agent and controlled OBJECTS containing controlled PARAMETERS),
- management protocol (eg, SNMP).

The diagram illustrates a management system where an operator (represented by a person icon) is connected to three controlled devices (represented by server icons). Each controlled device contains an agent (represented by a small server icon). The operator and agents are connected via a management protocol, indicated by arrows labeled 'management protocol'. The agents are also connected to their respective controlled devices.

History: management protocols

OSI CMIP

- Common Management Information Protocol,
- ITU-T X.700 standard created in 1980: first management standard,
- standardized too slow, never implemented in practice

SNMP

- Simple Network Management Protocol,
- IETF standard
- very simple first version,
- rapid deployment and expansion in practice
- currently: SNMP V3 (added safety!),
- de facto standard for network management.

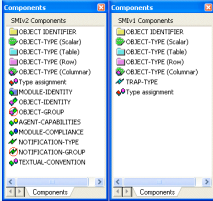
Management data

- For each type of controlled device we have our own MIB (Management Information Base) where information regarding managed OBJECTS and their PARAMETERS is stored.
- The operator has his own MDB (Management Database), where he stores concrete values for MIB objects/parameters for each managed device.
- A language that defines how OBJECTS and PARAMETERS are written is needed: SMI (Structure of Management Information)

Management Information Base (MIB)			
Object #1 Name	Syntax	Access /	Max-Access
Status	Definition /	Optional	Characteristics
Object #2 Name	Syntax	Access /	Max-Access
Status	Definition /	Optional	Characteristics
Object #3 Name	Syntax	Access /	Max-Access
Status	Definition /	Optional	Characteristics
Object #4 Name	Syntax	Access /	Max-Access
Status	Definition /	Optional	Characteristics

SMI: language for defining objects in MIB

- basic data types: INTEGER, Integer32, Unsigned32, OCTET STRING, OBJECT IDENTIFIED, IPAddress, Counter32, Counter64, Gauge32, Time Ticks, Opaque
- structured data types:
 - OBJECT-TYPE
 - MODULE-TYPE



SMI: object definition

- object definition: it contains data type, status, and meaning description

```

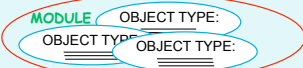
ipSystemStatsInDelivers OBJECT TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The total number of input datagrams successfully
    delivered to IP user-protocols (including ICMP)"
 ::= { ip 9}
    
```

SMI: grouping objects into modules

- MODULE: content-related group of objects

```

ipMIB MODULE-IDENTITY
LAST-UPDATED "941101000Z"
ORGANIZATION "IETF SNMPv2 Working Group"
CONTACT-INFO " Keith McCloghrie ....."
DESCRIPTION
    "The MIB module for managing IP and ICMP implementations,
    but excluding their management of IP routes."
REVISION "019331000Z"
 ::= { mib-2 48}
    
```



MIB modules: standardization

- **MODULES:**
 - "standardized",
 - vendor-specific
- IETF (Internet Engineering Task Force) responsible for standardization of MIB modules for routers, interfaces and other network equipment
 - -> naming (labeling) of standard components is required!
 - ISO ASN.1 (Abstract Syntax Notation 1) designation is used

MIB modules: standardization

- hierarchical arrangement of objects with tree identifiers
- each object has a name consisting of a sequence of number identifiers from the tree root to a leaf
 - example: 1.3.6.1.2.1.7 means UDP protocol

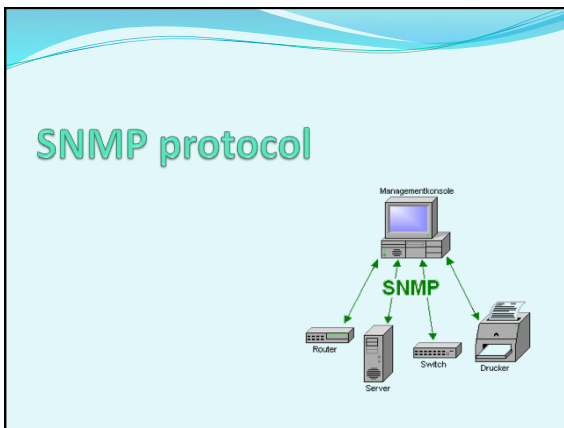
> challenge: what is on the second and third level of the tree identifiers?

MIB: naming, example

- Example:
 - 1.3.6.1.2.1.7 provides protocol UDP
 - 1.3.6.1.2.1.7.* provides the observed parameters of the UDP protocol

MIB: naming, example

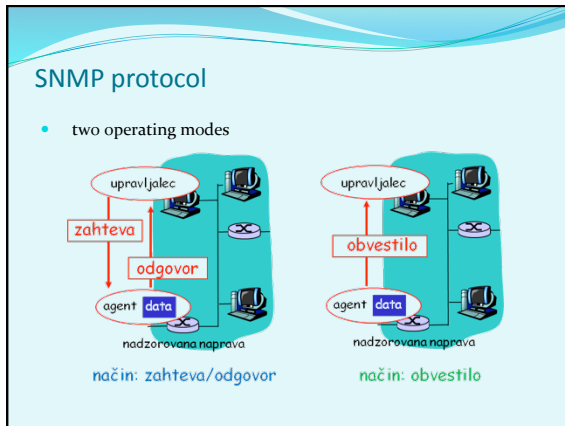
Object ID	Name	Type	Comments
1.3.6.1.2.1.7.1	UDPInDatagrams	Counter32	total # datagrams delivered at this node
1.3.6.1.2.1.7.2	UDPNoPorts	Counter32	# undeliverable datagrams no app at port1
1.3.6.1.2.1.7.3	UDInErrors	Counter32	# undeliverable datagrams all other reasons
1.3.6.1.2.1.7.4	UDPOutDatagrams	Counter32	# datagrams sent
1.3.6.1.2.1.7.5	udpTable	SEQUENCE	one entry for each port in use by app, gives port # and IP address



SNMP protocol

- *Simple Network Management Protokol*
- protocol for exchanging control information between the operator and monitored objects.
- information of controlled objects is being transferred between controlled equipment and the operator with accordance to the MIB definition.
- Two operating modes:
 - *request-response*: reading and setting values
 - *trap message*: the device informs the operator about the event

The diagram shows the interaction between a 'Linux Console' (operator) and a 'Linux Host' (monitored object) through a 'Manager Work' station. A dashed arrow labeled 'SNMP GET & SET' points from the Manager Work to the Linux Host. Another dashed arrow labeled 'SNMP TRAP' points from the Linux Host back to the Manager Work.

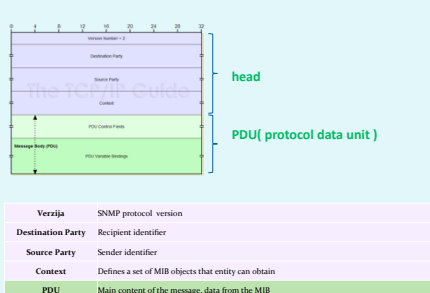


SNMP: message types

Message	Direction	Meaning
GetRequest GetNextRequest GetBulkRequest	operator -> agent	"give me information" (value, next in list, data block-table)
InformRequest	operator -> operator	mutual transmission of values from MIB
SetRequest	operator -> agent	set the value in MIB
Response	agent -> operator	"here is the value", response to Request
Trap	agent -> operator	notification to operator about the incident

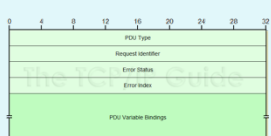
- ### SNMP protocol
- > challenge: find RFC documents about SNMP and find differences between them
 - SNMP uses UDP transport protocol
 - port 161: "general" SNMP port, where devices listen for SNMP requests
 - port 162: notifications port (traps), usually where systems listen for control and management of a network
 - SNMP implementation must address the following problems:
 - package size:** SNMP packets can contain extensive information about objects in MIB, UDP on the other hand has an upper limit for the size of the segment (TCP doesn't),
 - resending:** since UDP is used, delivery and confirmation is not guaranteed. Delivery control should therefore be addressed at a higher OSI level.
 - problem with lost notifications:** if a notification is lost during transfer, the sender doesn't know anything about it; the recipient also doesn't receive it
 - > challenge: how does SNMPv3 address these problems?

SNMP: message form



Verzija	SNMP protocol version
Destination Party	Recipient identifier
Source Party	Sender identifier
Context	Defines a set of MIB objects that entity can obtain
PDU	Main content of the message, data from the MIB

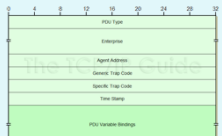
SNMP: request-response message type



PDU Type Value	PDU Type
0	GetRequest-PDU
1	GetNextRequest-PDU
2	Response-PDU
3	SetRequest-PDU
4	Trap (obsolete, not used; should use the old Trap-PDU or SNMPv1 Trap-Request-PDU)
5	Trap (obsolete, not used; should use the old Trap-PDU or SNMPv1 Trap-Request-PDU)
6	InformRequest-PDU
7	Trapv2-PDU
8	Request-PDU

Request ID	Integer	Number that relates a request with response. A device that answers, when it stores into a package of Response type. It is also used for artificial control of received packets (SNMP uses UDP transport protocol which doesn't provide this!)
Error Status	Integer	Error code which agent forwards with a Response type package. Value 0 means that there was no error and any other value defines a specific error. -> challenge: look at different types of errors
Error Index	Integer	If there was an error, this value is the index of an object that caused the error.
Variable Bindings	Variable	Name-value pairs, that define objects and their values.

SNMP: notification type message



PDU Type	Integer	Value that defines the type of message. Value 4/7 means notification (trap message).
Enterprise	Sequence of Integer	Group identifier.
Agent Address	Network Address	IP address of the agent that generated a notification.
Generic Trap Code	Integer	General error code - from predefined coding.
Specific Trap Code	Integer	Specific error code (depends on the manufacturer equipment)
Time Stamp	Time/Ticks	Time since the last time the device initialized. Used for recording.
Variable Bindings	Variable	Name-value pairs that define objects and their values.

Verzije SNMP

- **SNMPv1**
 - defined in the late 80s
 - turned out to be too weak to implement all the necessary requirements (limited in composition of PDU)
- **SNMPv2**
 - improved SNMPv1 in speed (added GetBulkRequest), safety (but too complex implementation), communication between operators,
 - RFC 1901, RFC 2578
 - uses SMlv2 (improved standard for structuring information)
- **SNMPv3**
 - improved SNMPv2 – added safety mechanisms,
 - enables cryptography, assures safety, integrity, authentication
 - also uses SMlv2

Safety

- **Why is it important?**
 - SetRequest adjusts controlled devices. Request can be sent at any time?
 - challenge: find 3 more examples of other possible SNMP abuses.
 - Safety elements are only introduced in SNMPv3, previous version did not have it. SNMPv3 has built-in security based on user names
 - challenge: read RFC 3414 and find information about which kind of intrusions does SNMPv3 enable protection against. How about Denial of Service attacks and eavesdropping on traffic?

SNMP. Safety mechanisms

1. **packets content encryption (PDU):** DES is used (exchange of keys is required prior to use)
2. **integrity:** used for message densification with a key which is known to both sender and recipient. With examination of sent densified value we have control over active message counterfeiting

The red fox jumps over the blue dog

→

cryptographic hash function

→

FCD3 7FD8 5AF2 C5FF 915F
D401 C0A9 7D5A 46AF F845

The red fox jumps over the blue dog

→

cryptographic hash function

→

8ACA D682 D588 4C75 43F4
1799 7D88 9CF8 9289 66AC

SNMP: Safety mechanisms

3. **protection against repetition of already completed communication (replay attack):** use of one-time chips (*nonce*, *žeton*): the sender must encode the message according to the nonce which is defined by the receiver (this is usually the number of system start-ups and the time passed since the last start-up)

MAC = f(sporočilo, koda, žeton)

SNMP: Safety mechanisms

4. **access control:** access control based on user names. The user rights specify which users can read/change which information. User data is stored in *Local Configuration DataStore* database which also contains controlled objects s SNMP!

> challenge: examine RFC 3445. What is a View-based Access Control Model Configuration MIB?

(C) SNMP Research International, Inc.

Encoding PDU content

• How to encode packet content so that it is understood on all platforms (different data types are of different lengths, thick/thin end)?

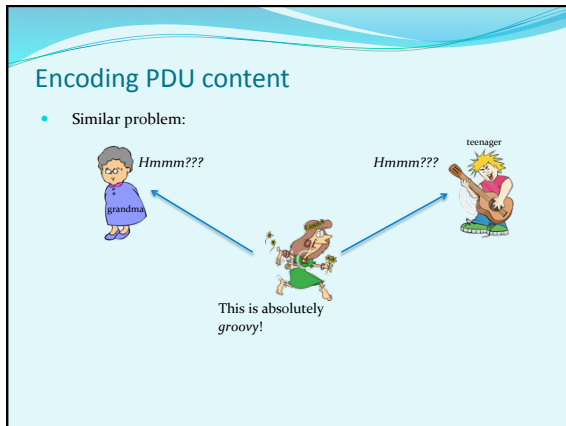
test.code	a	test.x	256	test.code	a
test.x	00000001	test.x	a	test.x	00000011
	00000011				00000001

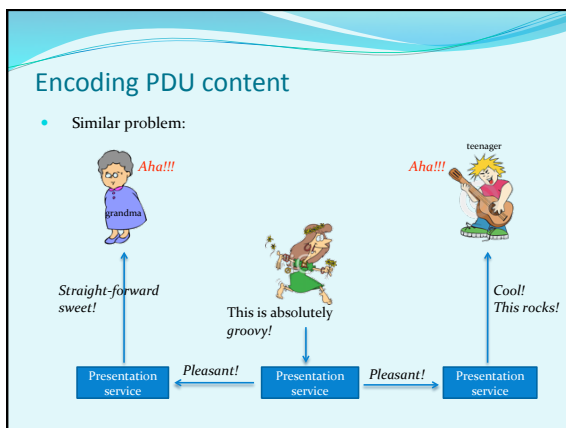
host 1 format host 2 format

How to make this transfer?

• we need a uniform coding or some **demonstration level of this data**

- ASN.1 standard in addition to data types also defines encoding standards.
- we will see that TLV notation is used for presentation of these operators.





Presentation service: possible solutions

1. **Sender accounts** the data form used by the recipient: he converts data into the correct form for recipient and only then sends it.
2. sender sends data in his own form, **recipient converts** into his own form
3. Sender converts into **independent form** and then sends. Recipient transforms independent form into his own.
 - challenge: what are advantages and disadvantages of these three approaches?

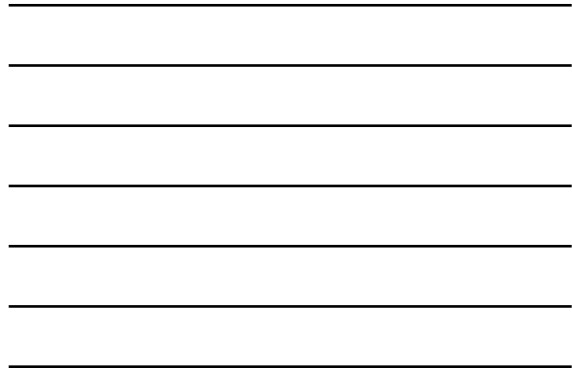
- ASN.1 uses the (3). third solution(**independent form**).
- **BER rules** are used when writing types (Binary Encoding Rules). They define the recording of **data according to TLV principle** (Type, Length, Value).

Example of BER encoding according to TLV principle

Basic ASN.1 data type	Type No.	Use
BOOLEAN	1	Model logical, two-state variable values
INTEGER	2	Model integer variable values
BIT STRING	3	Model binary data of arbitrary length
OCTET STRING	4	Model binary data whose length is a multiple of eight
NULL	5	Indicate effective absence of a sequence element
OBJECT IDENTIFIER	6	Name information objects
REAL	9	Model real variable values
ENUMERATED	10	Model values of variables with at least three states
CHARACTER STRING	*	Model values that are strings of characters from a specified character set

Value, 259
Length, 2 bytes
Type, 2, integer

Value, 5 octets (chars)
Length, 5 bytes
Type, 4, octet string



SNMP package capture




SNMP program structure



Other monitoring approaches

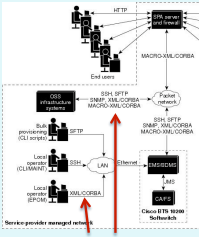
MAIL-ORDER ALTERNATIVE MEDICINE



Skip the herbs...
skip the needles...
simply write us a
check and pretend
it worked!

Alternative boutique solutions

1. XML & SOAP (application level): XML enables graphic and hierarchical way of encoding data which represent elements and content of controlled objects in the network. SOAP is a simple protocol that enables exchange of XML documents in the network.
 - ✓ easy reading and understanding of content on the receiver side.
 - large overhead compared to binary data encoding
2. CORBA (Common Object Request Broker Architecture) (application level): architecture that defines inter-utility of objects of different programming languages and on different architectures.

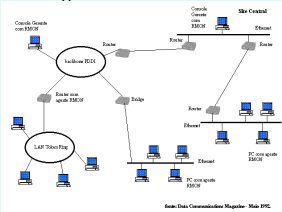


protocol combination!

Event-driven monitoring

RMON (Remote Monitoring) (additional mechanism): Classical SNMP can control the network from a control station. RMON collects and analyses measures locally and sends the results to a remote control station. It has its own MIB with extensions for different media types.

- ✓ every RMON agent is responsible for local control,
- ✓ sending already completed analysis reduces SNMP traffic between sub-networks
- ✓ It isn't necessary that agents are always visible from the central control system side.
- longer establishment and installation time of system is required.



Homework

Assignment for additional points with homework's:

Read RFC 789 which describes a known ARPAnet network failure which happened in 1980.

How could the network failure be avoided or it's recovery time improved if the network administrators would have today's tool for network management and control at their disposal?

Next time we are moving on!

- traffic for applications in real time!