



HÁLÓZATI RENDSZEREK
ÉS SZOLGÁLTATÁSOK
TANSZÉK

BMEVIHIMA00 Hálózati technológiák integrációja

6 Hálózatmenedzsment

6b SNMP

Jakab Tivadar
jakab@hit.bme.hu

Budapest,
2020.06.12.



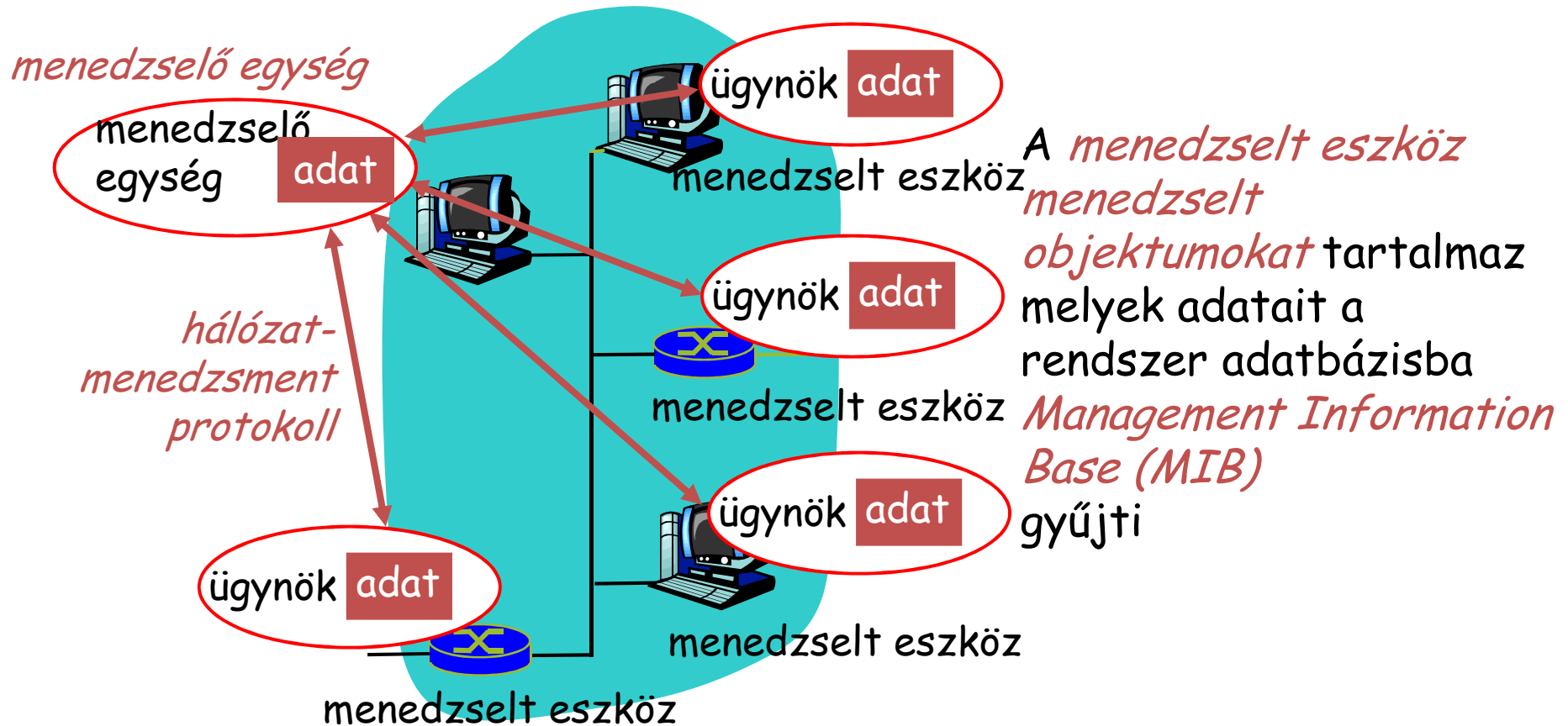


SNMP ÁTTEKINTÉS

Kurose: Computer Networking 9. Fejezet Network Management alapján

Addison-Wesley ISBN-10:0321497708 ISBN-13:9780321497703

definíciók:



OSI CMIP

- Common Management Information Protocol
- a 1980-as években tervezték: *az egységesítő szabvány*
- nagyon lassú volt a szabványosítási folyamat

SNMP: Simple Network Management Protocol

- Internet-gyökerek (SGMP)
- egyszerűnek indult
- gyors alkalmazás és adaptáció
- növekedése: méret, komplexitás
- aktuális változat: SNMP V3
- *de facto* hálózatmenedzsment szabvány

- **Management information base (MIB):**
 - a hálózatmenedzselési adatok elosztott adatbázisa
- **Structure of Management Information (SMI):**
 - Adatleíró nyelv a MIB objektumokhoz
- **SNMP protocol**
 - szállítás menedzselő<->menedzselt objektum adatok, parancsok
- **Biztonsági és adminisztrációs képességek**
 - légyeges új funkciók az SNMPv3-ban

Cél: a menedzsel adatok szintaxisának és szemantikájának egyértelmű, jól definiált leírása

- alapvető adattípusok:
 - ...
- objektum típus
 - adattípus, státusz, a menedzsel objektumok szemantikája
- modulazonosító
 - csoportok objektumok a MIB modulba

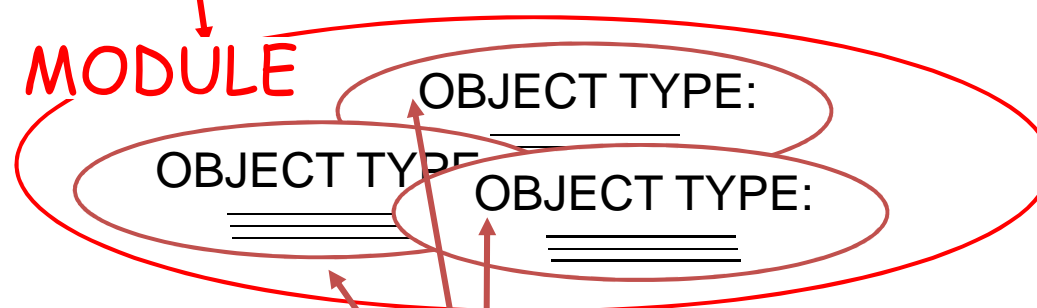
Alapvető adattípusok

INTEGER
Integer32
Unsigned32
OCTET STRING
OBJECT IDENTIFIED
IPAddress
Counter32
Counter64
Gauge32
Time Ticks
Opaque

MIB module : SMI alapján specifikálva

MODULE-IDENTITY

(100 szabványosított MIB, többnyire gyártó-specifikus)



Az objektumok SMI alapján
specifikálva
OBJECT-TYPE construct



OBJECT-TYPE: ipInDelivers

ipInDelivers 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}

MODULE-IDENTITY: ipMIB

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}

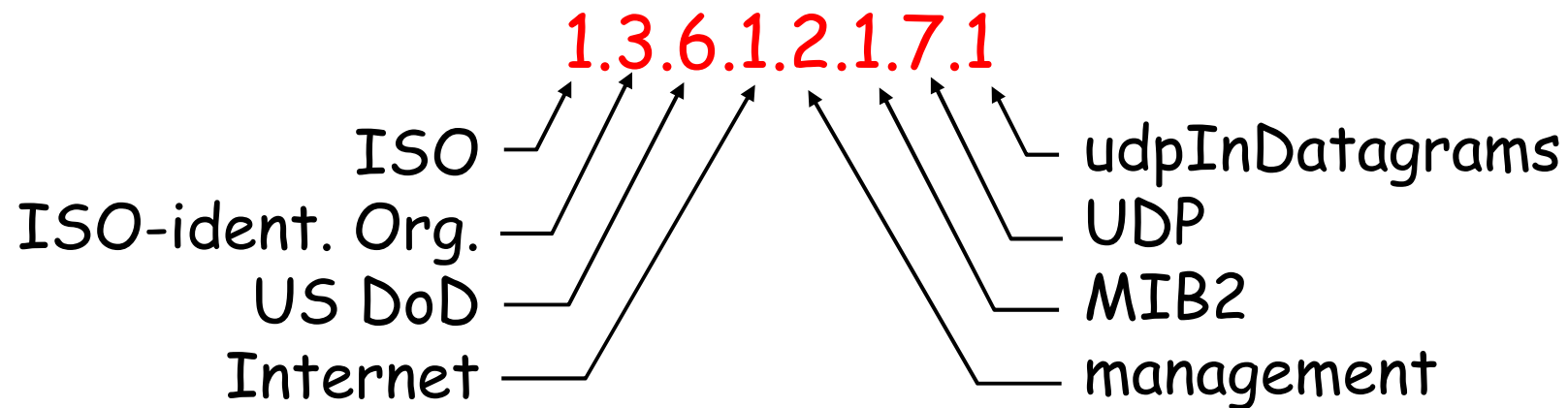
MIB PÉLDA: UDP MODUL

<u>Object ID</u>	<u>Name</u>	<u>Type</u>	<u>Comments</u>
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 port!
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

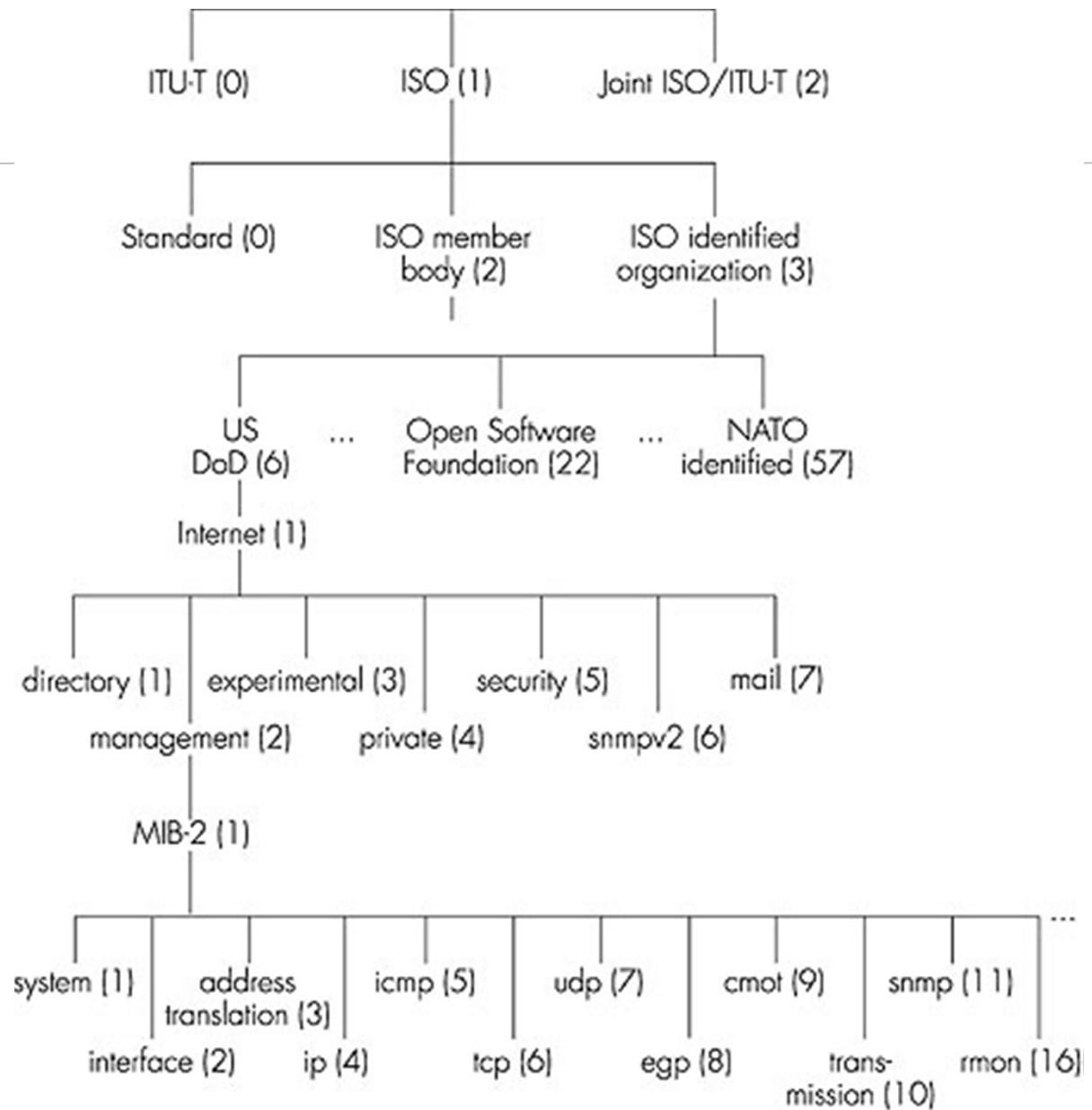
kérdés: hogyan nevezzük el az az összes lehetséges hálózati szabványban előforduló összes lehetséges szabványos objektumot (protokoll, adat, stb.)??

answer: ISO Object Identifier tree:

- Hierarchikus, fa alapú elnevezés
- minden elágazási pontnak neve (száma) van



OSI OBJECT IDENTIFIER TREE

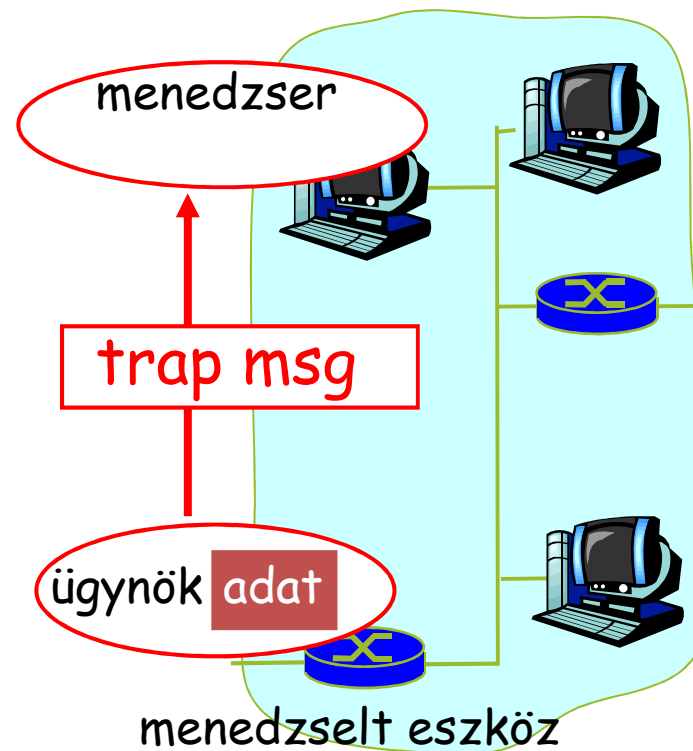


Check out www.alvestrand.no/harald/objectid/top.html

Két módszer a MIB adatok és parancsok szállítására:



request/response mode

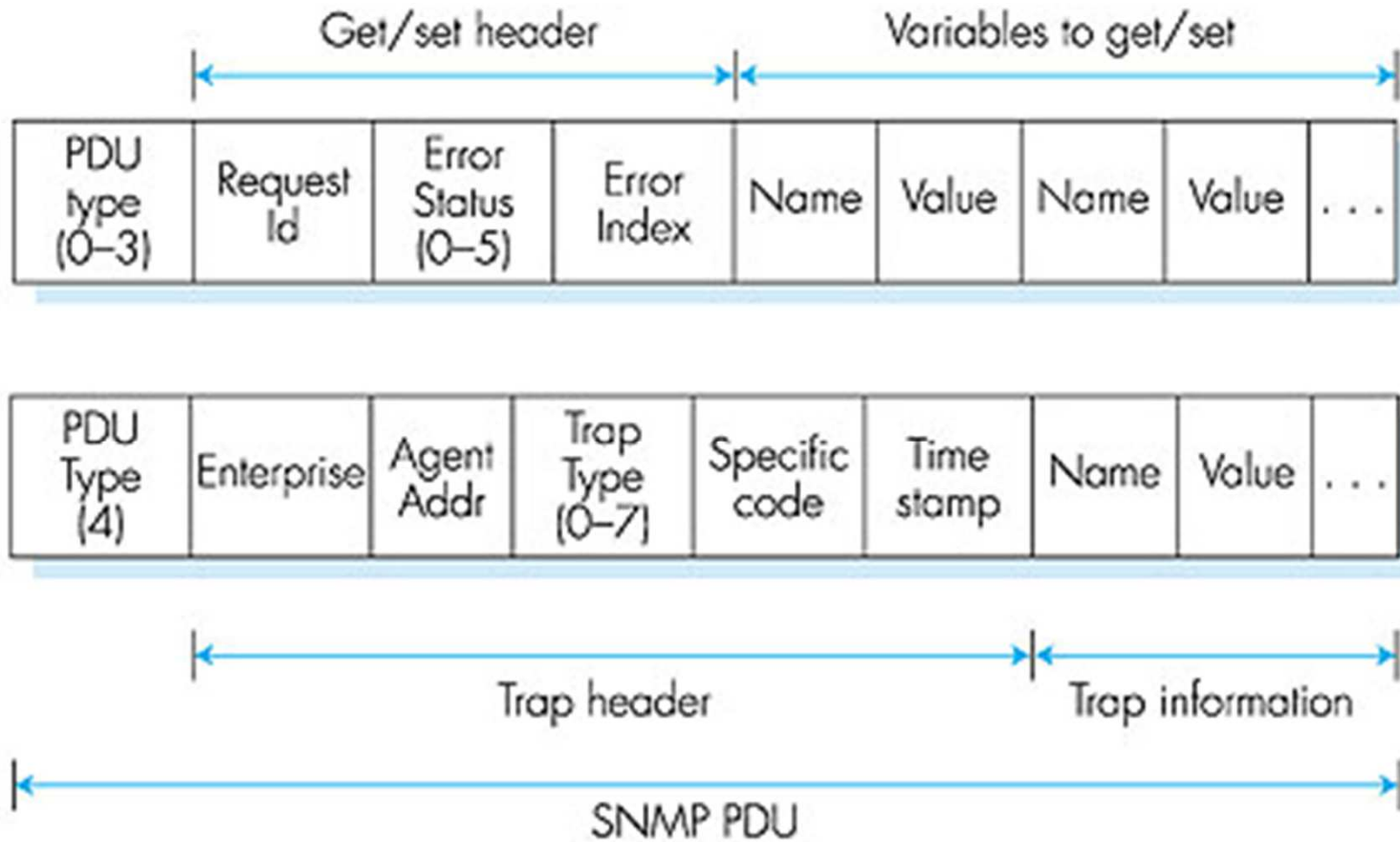


trap mode

SNMP PROTOCOL: MESSAGE TYPES

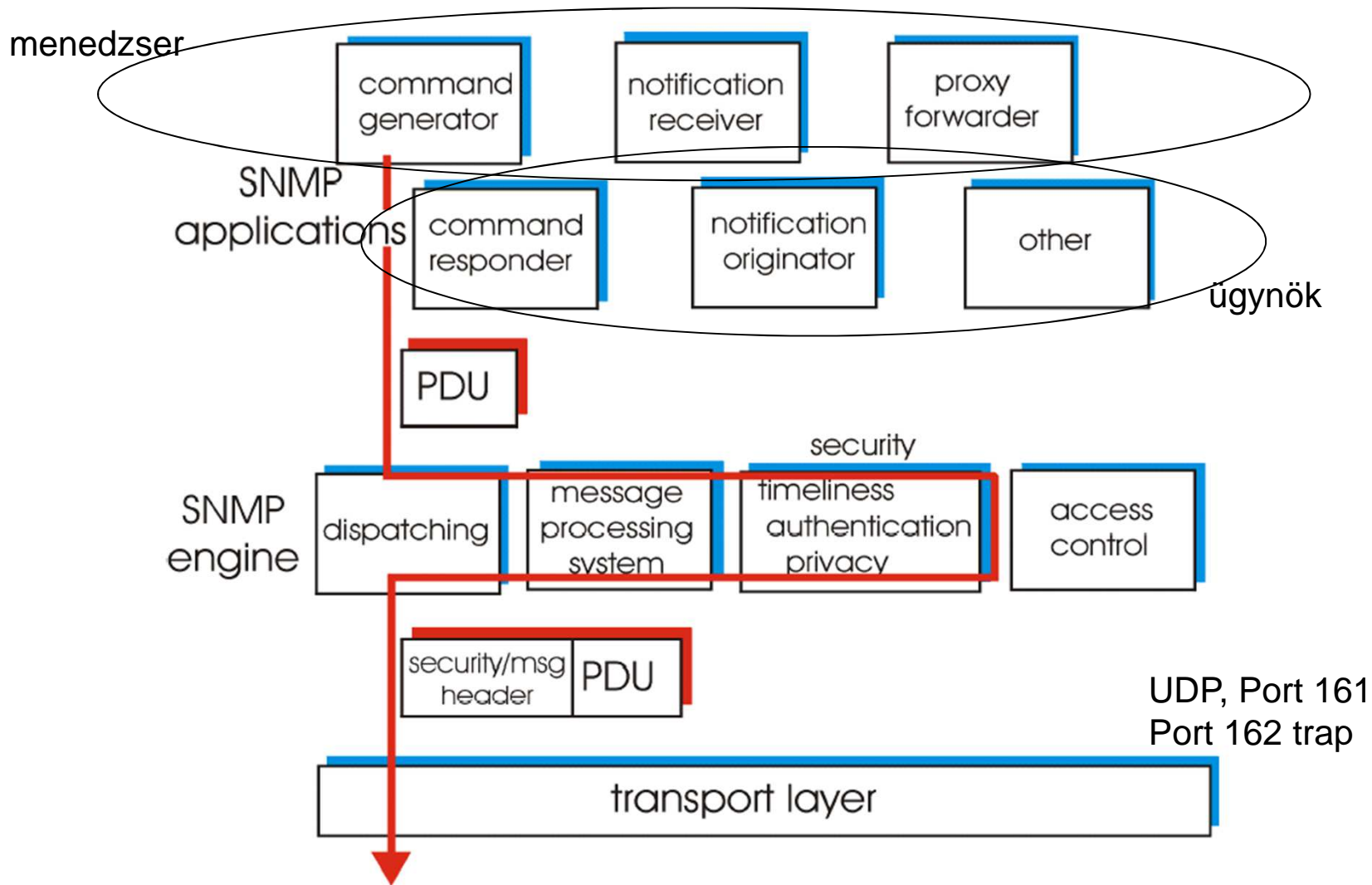
<u>Message type</u>	<u>Function</u>
GetRequest GetNextRequest GetBulkRequest	Menedzser ügynöknek: "küldj adatot" (egyedi, következő listaelem, blokk)
InformRequest	Menedzser menedzsernek: MIB adat
SetRequest	Menedzser ügynöknek : MIB adat írás
Response	Ügynök menedzsernek: adat, válasz Request-re
Trap	Ügynök menedzsernek: jelzés rendkívüli eseményről

SNMP PROTOCOL: MESSAGE FORMATS



- **encryption:** DES-encrypt SNMP message
- **authentication:** compute, send $\text{MIC}(m,k)$:
compute hash (MIC) over message (m),
secret shared key (k)
- **protection against playback:** use nonce
- **view-based access control**
 - SNMP entity maintains database of access rights, policies for various users
 - database itself accessible as managed object!

SNMPV3 ENGINE AND APPLICATIONS



THE PRESENTATION PROBLEM – ASN.1

Q: does perfect memory-to-memory copy solve “the communication problem”?

A: not always!

```
struct {
  char code;
  int x;
} test;
test.x = 256;
test.code='a'
```

test.code	a
test.x	00000001
	00000011

host 1 format

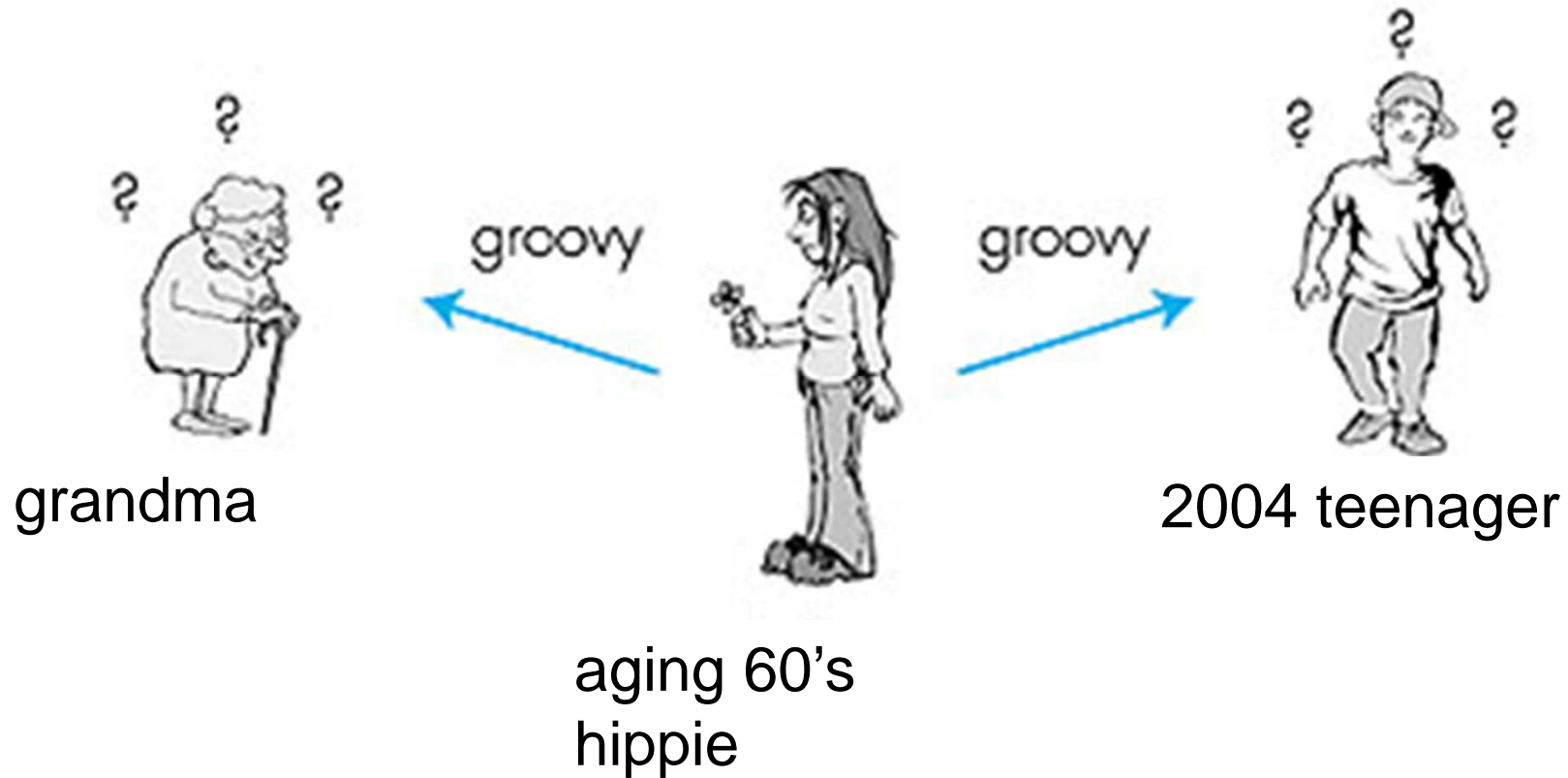
test.code	a
test.x	00000011
	00000001

host 2 format

problem: different data format, storage conventions

A REAL-LIFE PRESENTATION

PROBLEM:

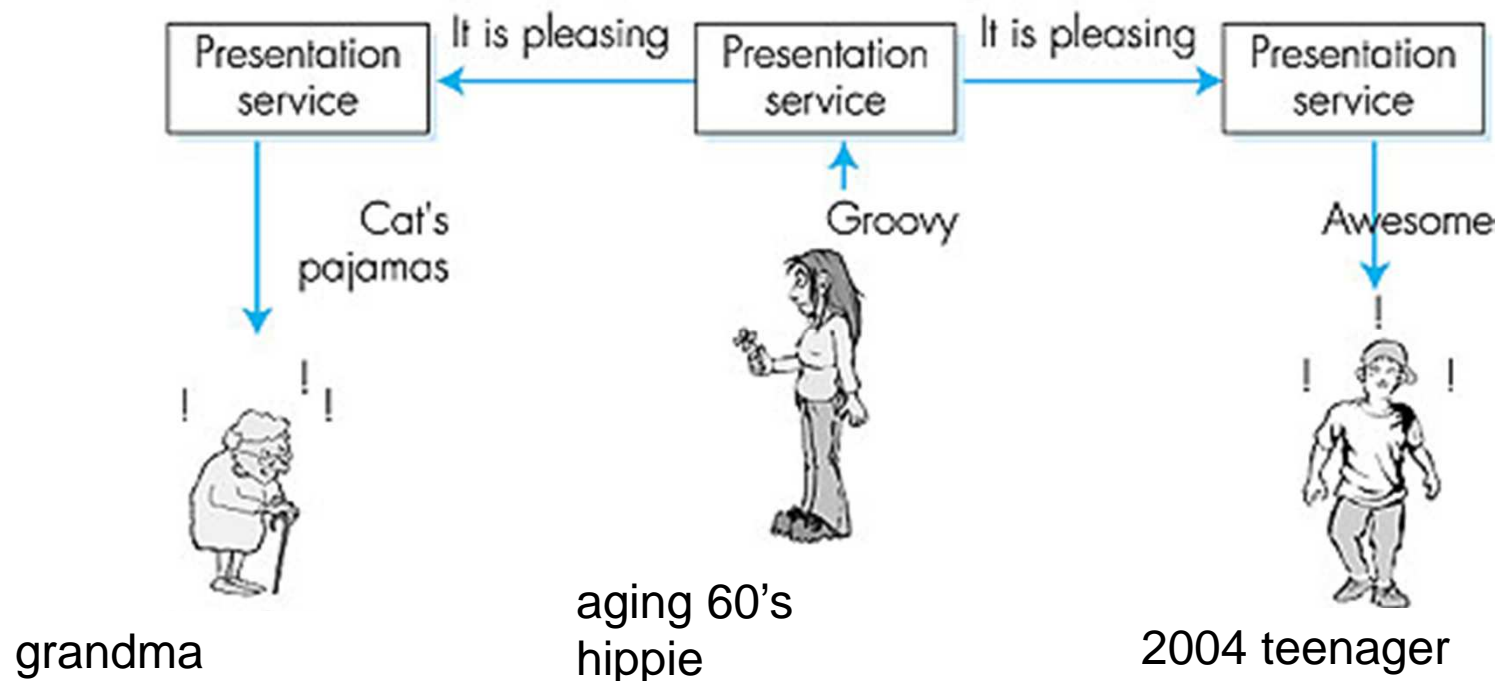


PRESENTATION PROBLEM: POTENTIAL SOLUTIONS

1. Sender learns receiver's format. Sender translates into receiver's format. Sender sends.
 - *real-world analogy?*
 - *pros and cons?*
2. Sender sends. Receiver learns sender's format. Receiver translate into receiver-local format
 - *real-world-analogy*
 - *pros and cons?*
3. Sender translates host-independent format. Sends. Receiver translates to receiver-local format.
 - *real-world analogy?*
 - *pros and cons?*

SOLVING THE PRESENTATION PROBLEM

1. Translate local-host format to host-independent format
2. Transmit data in host-independent format
3. Translate host-independent format to remote-host format



- **ISO standard X.680**
 - used extensively in Internet
 - like eating vegetables, knowing this “good for you”!
- **defined data types**, object constructors
 - like SMI
- **BER: Basic Encoding Rules**
 - specify how ASN.1-defined data objects to be transmitted
 - each transmitted object has Type, Length, Value (TLV) encoding

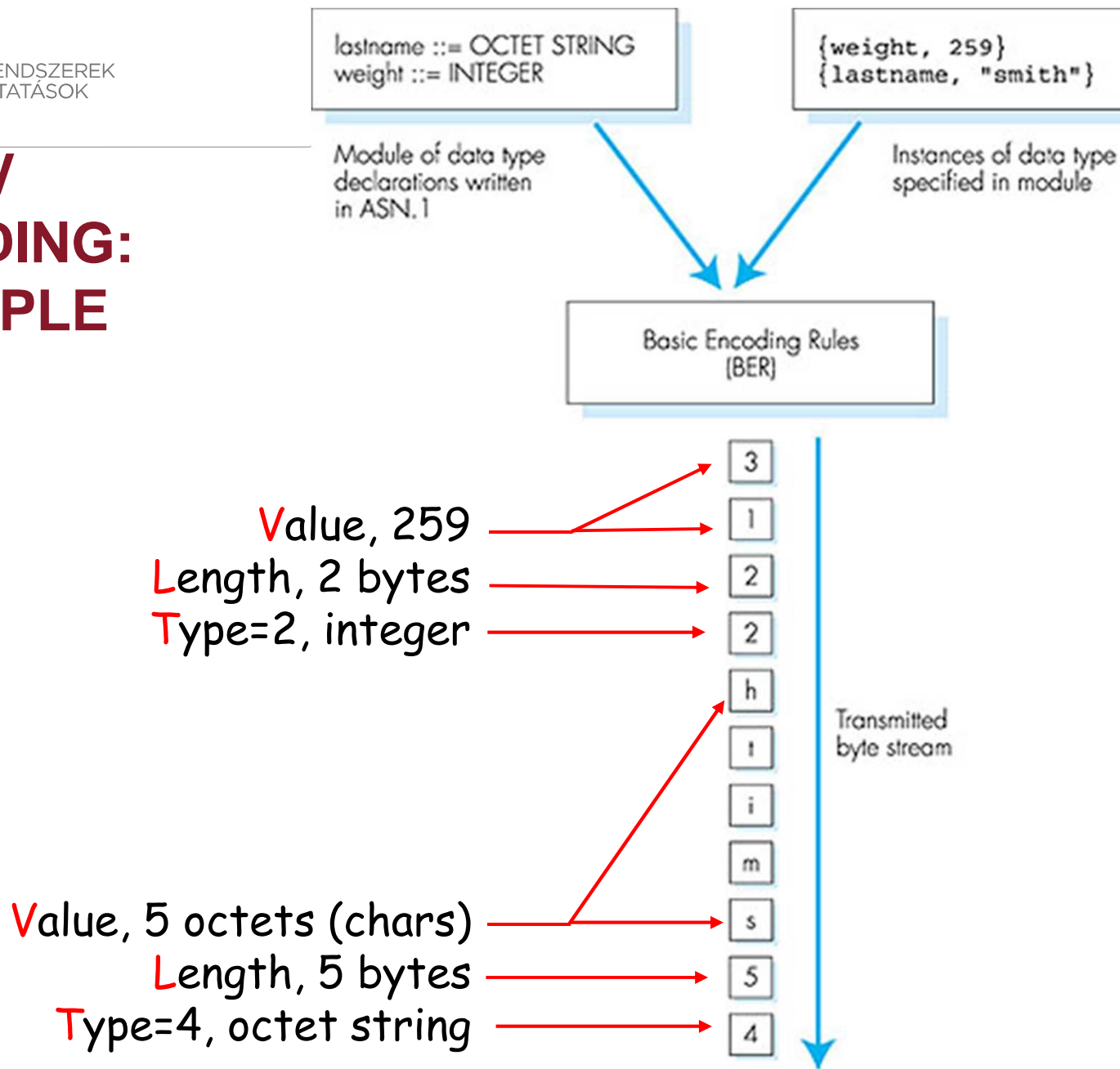


Idea: transmitted data is self-identifying

- T: data type, one of ASN.1-defined types
- L: length of data in bytes
- V: value of data, encoded according to ASN.1 standard

<u>Tag Value</u>	<u>Type</u>
1	Boolean
2	Integer
3	Bitstring
4	Octet string
5	Null
6	Object Identifier
9	Real

TLV ENCODING: EXAMPLE



- network management
 - extremely important: 80% of network “cost”
 - ASN.1 for data description
 - SNMP protocol as a tool for conveying information
- Network management: more art than science
 - what to measure/monitor
 - how to respond to failures?
 - alarm correlation/filtering?