Network Technologies (TCP/IP Suite)

Umar Kalim
Dept. of Communication Systems Engineering

umar.kalim@niit.edu.pk
http://www.niit.edu.pk/~umarkalim

15/06/2007

Outline

- Simple Network Management Protocol (SNMP)

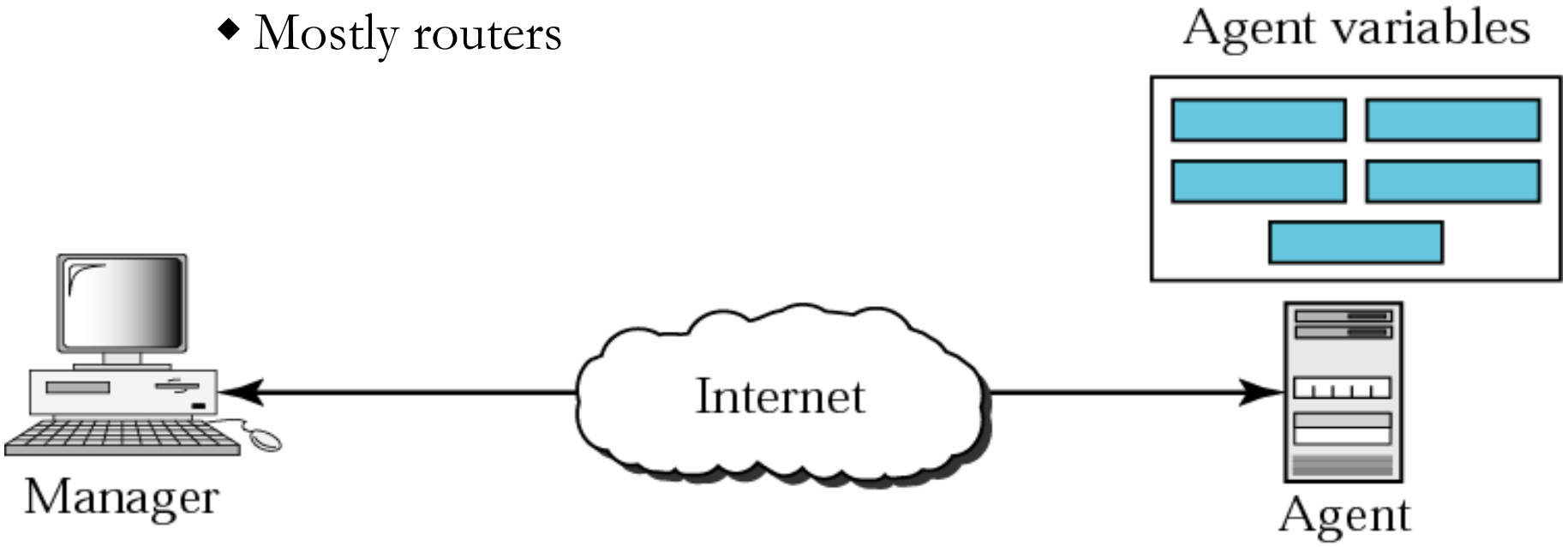
Simple Network Management Protocol

- A framework for managing devices in an internet using the TCP/IP protocol suite
  - It provides the fundamental operations for monitoring and maintaining an internet

- Application level protocol
  - So that it may monitor devices made by different manufacturers, installed on different networks
    - Limitations of the type of the network as well as the type of the device is removed

Concept of SNMP

- Uses the concept of a manager and an agent
  - Manager is the host that controls/monitors a (set of) agent(s)
  - Agents are hosts that are being monitored
    - Mostly routers



Manager & Agents

- Manager runs the SNMP client
- Agent runs the SNMP server
- Management is achieved by the simple interaction b/w the client and the server
- Agents maintain information about parameters affecting performance
  - e.g. # of packets received, # of packets forwarded etc
- The Manager has access to these parameters
- Manager can also have the agent perform certain actions
  - e.g. Reboot the router

Manager & Agents

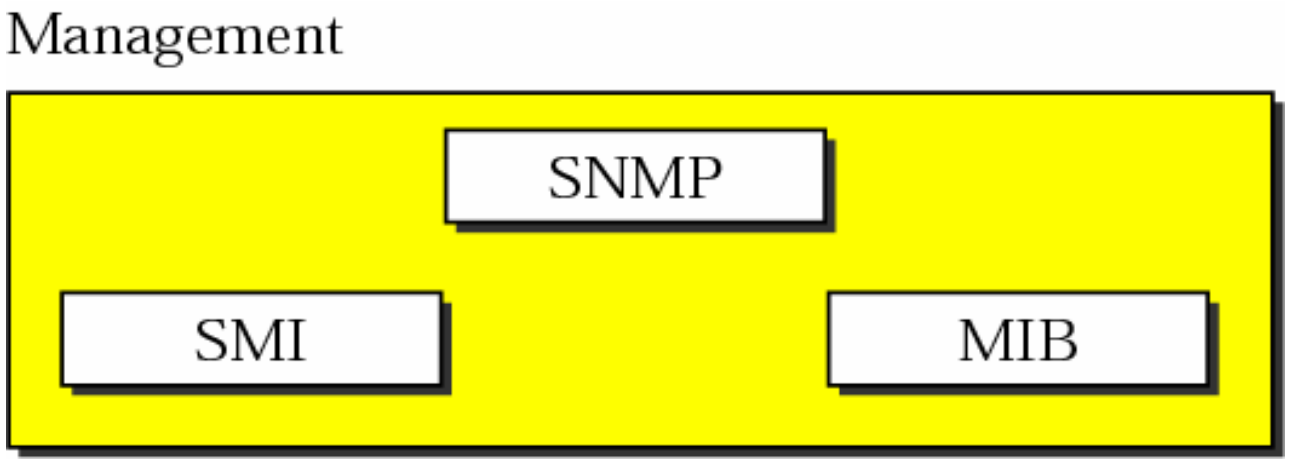
- Agents can also contribute to the management process
  - The server program checks the parameters for certain defined thresholds
  - If a threshold is reached, or there is some unusual activity, the agent sends a warning message (called a *trap*) to the manager

Concept of SNMP

1. Manager checks an agent by requesting information that reflects the behaviour of the agent
2. A manager can force an agent to perform a task (by resetting certain values in the agents database)
3. An agent can help in the management by generating trap messages/warning in case of an unusual activity

Management Components

- SNMP uses two other protocols to achieve its objectives
  - Structure of Management Information (*SMI*)
  - Management Information Base (*MIB*)



Role of SNMP

- It defines the format of the packet to be sent from a manager to an agent and vice versa
  - Packets exchanged contain object names and their status
  - SNMP is responsible for *reading* and *changing* these values
- It interprets the results and creates statistics (often using other s/w)

Role of SMI

- To use SNMP we require rules
  - To name objects using a uniform convention for open access
  - To define what type of objects can be used (simple or structured, which simple types are available, their sizes etc) again for open access
- SMI defines such rules
  - How to define objects
  - How to list their types

Note:

SMI defines the general rules for naming objects, defining object types (including range and length), and showing how to encode objects and values.

SMI defines neither the number of objects an entity should manage, nor names the objects to be managed nor defines the association between the objects and their values.

Role of MIB

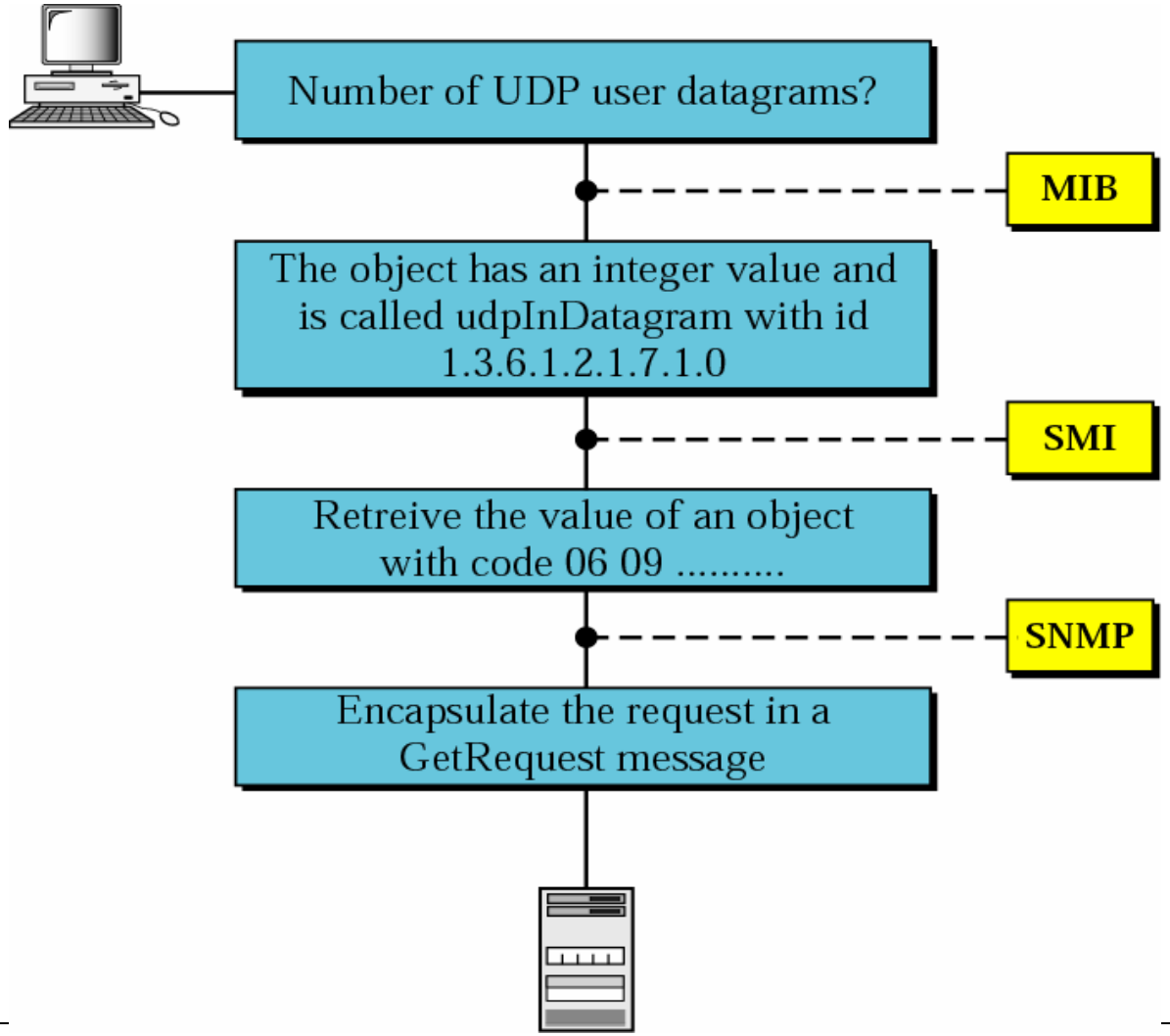
- For each entity to be managed, we must define
  - The number of objects
  - Name them
  - Associate a type with each

- MIB creates a collection of named objects, their types, and their relationships to each other in an entity to be managed

An Analogy to understand the relationship

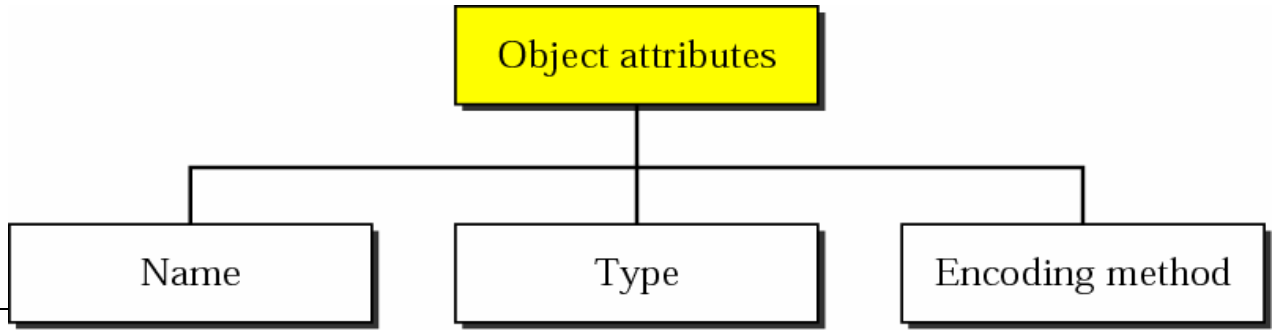
- Before we write a computer program, the syntax must be defined. Also, the available data types must be defined etc [**SMI**]
- Declaration of variables in the program [**MIB**]
- Write code to read, change the variables, interpret their meaning etc [**SNMP**]

Operational overview



Structure of Management Information

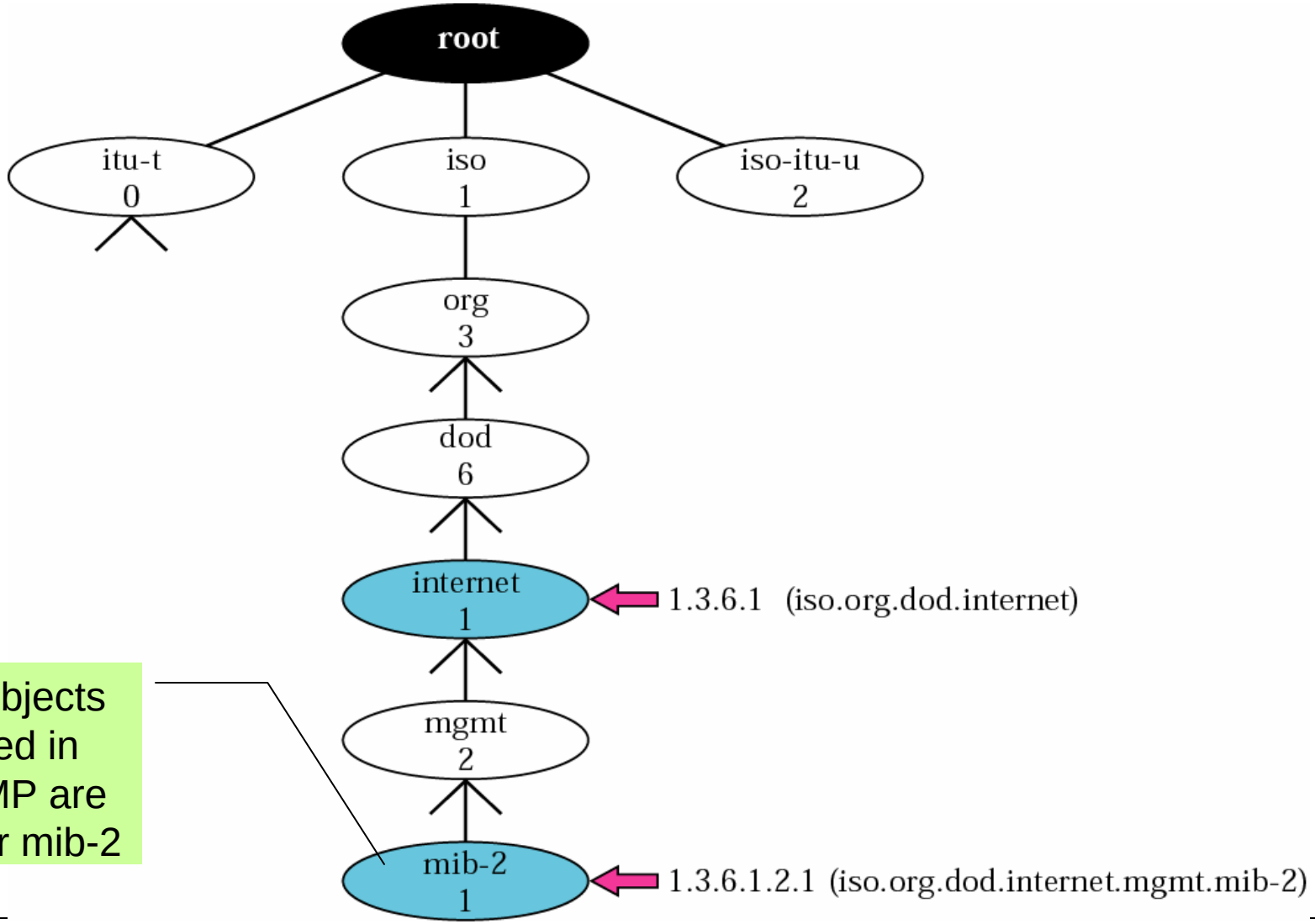
- SMI (SMIv2) has the following functions
  - To name objects
  - To define the type of data that can be stored
  - To show how to encode data for transmission over the network
- Correspondingly SMI emphasizes three attributes to handle an object
  - Name
  - Type
  - encoding



Name

- SMI requires each managed object (such as a router, a variable in a router or a value etc) to have a unique name
- Thus SMI uses an *object identifier* which is a hierarchical identifies based on a tree

Object Identifier



All objects used in SNMP are under mib-2

Type

- To define the data types SMI uses *Abstract Syntax Notation 1* (ASN.1) and adds some new definitions
- SMI has two broad categories for types
  - Simple
  - structured

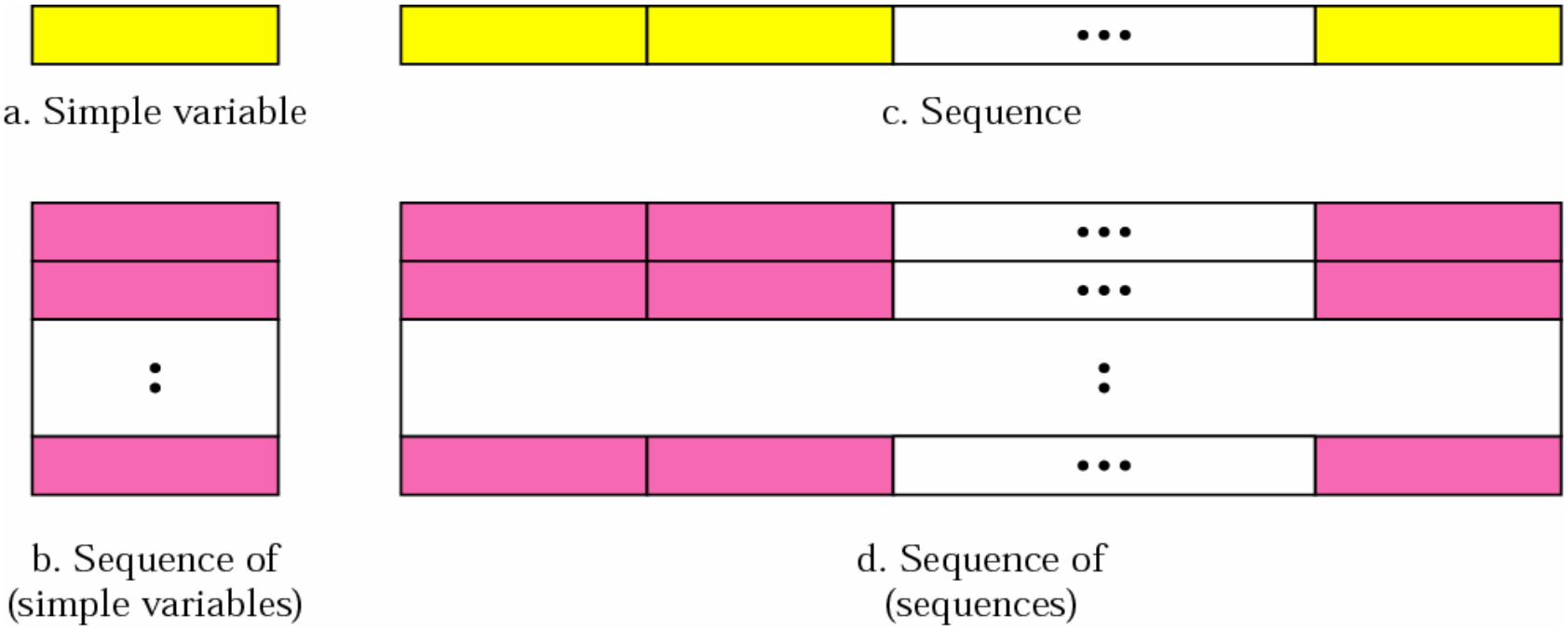
Simple types

Type	*Size*	*Description*
INTEGER	4 bytes	An integer with a value between -2^{31} and $2^{31}-1$
Integer32	4 bytes	Same as INTEGER
Unsigned32	4 bytes	Unsigned with a value between 0 and $2^{32}-1$
OCTET STRING	Variable	Byte-string up to 65,535 bytes long
OBJECT IDENTIFIER	Variable	An object identifier
IPAddress	4 bytes	An IP address made of four integers
Counter32	4 bytes	An integer whose value can be incremented from zero to 2^{32}; when it reaches its maximum value it wraps back to zero
Counter64	8 bytes	64-bit counter
Gauge32	4 bytes	Same as Counter32, but when it reaches its maximum value, it does not wrap; it remains there until it is reset
TimeTicks	4 bytes	A counting value that records time in 1/100ths of a second
BITS		A string of bits
Opaque	Variable	Uninterpreted string

Structured types

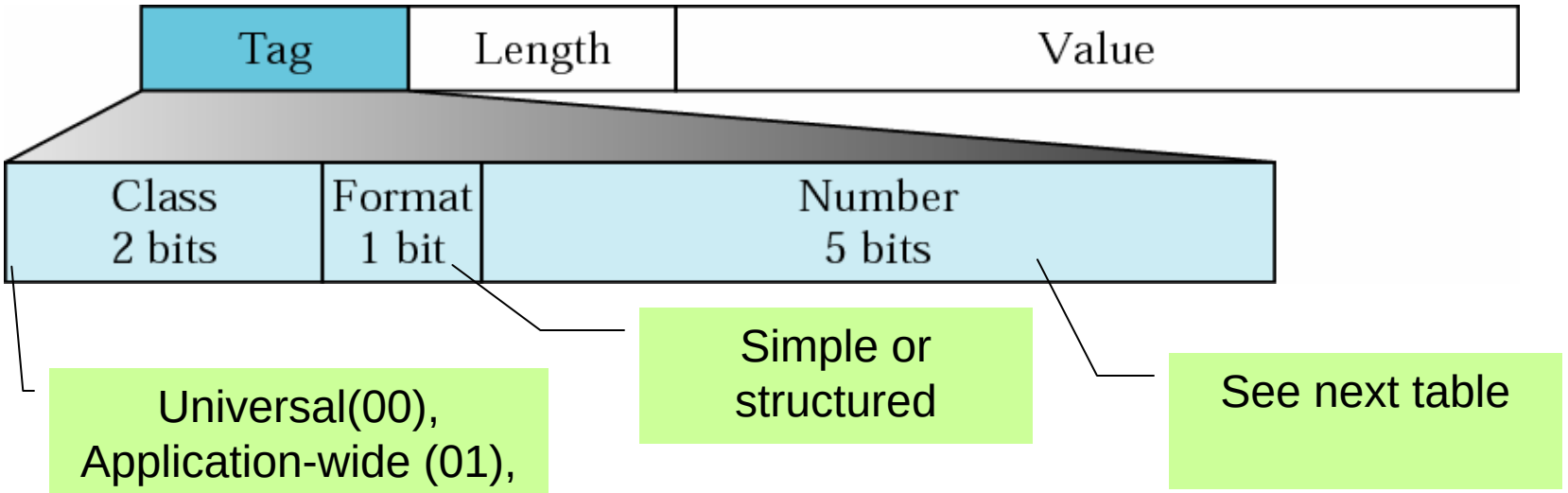
- By combining simple and structured types
- *Sequence*
  - Combination of simple/primitive data types
  - Not necessarily of the same type
    - e.g. struct
- *Sequenceof*
  - Combination of a simple data types, all of the same type
    - e.g. array

Conceptual data types



Encoding Method

- SMI uses the *Basic Encoding Rules* (BER) to encode data
- Each piece of data is encoded as



Codes for data types

Data Type	*Class*	*Format*	*Number*	*Tag (Binary)*	*Tag (Hex)*
INTEGER	00	0	00010	**00000010**	**02**
OCTET STRING	00	0	00100	**00000100**	**04**
OBJECT IDENTIFIER	00	0	00110	**00000110**	**06**
NULL	00	0	00101	**00000101**	**05**
Sequence, sequence of	00	1	10000	**00110000**	**30**
IPAddress	01	0	00000	**01000000**	**40**
Counter	01	0	00001	**01000001**	**41**
Gauge	01	0	00010	**01000010**	**42**
TimeTicks	01	0	00011	**01000011**	**43**
Opaque	01	0	00100	**01000100**	**44**

Management Information Base (MIB)

- MIB2
- Each agent
  - Collection of objects to be managed

System objects – information about the node – name, location, lifetime etc

Interface objects – information about the interfaces – number, physical address, ip etc

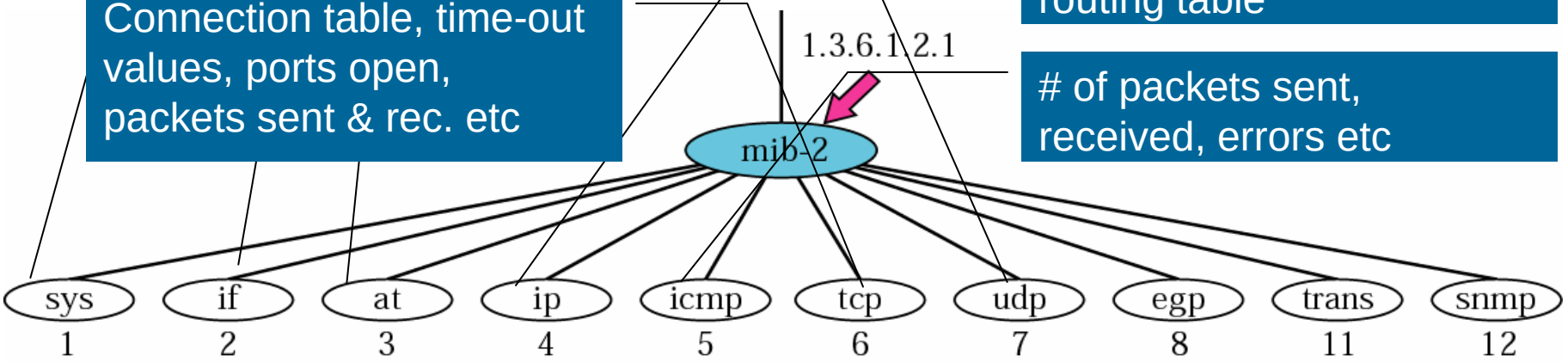
Address translation – information about the ARP table

Ports used, packets sent & rec. etc

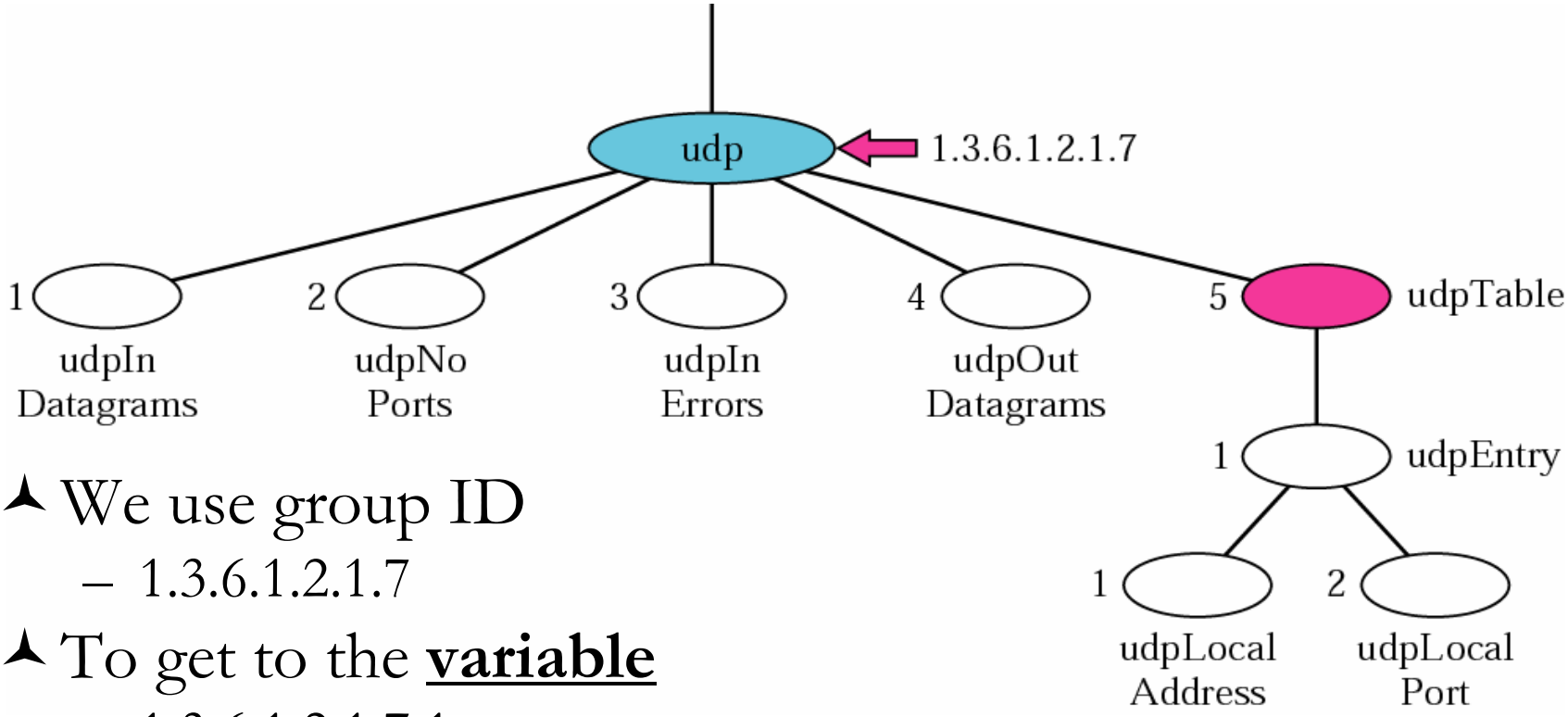
Information about the IP & routing table

Connection table, time-out values, ports open, packets sent & rec. etc

# of packets sent, received, errors etc

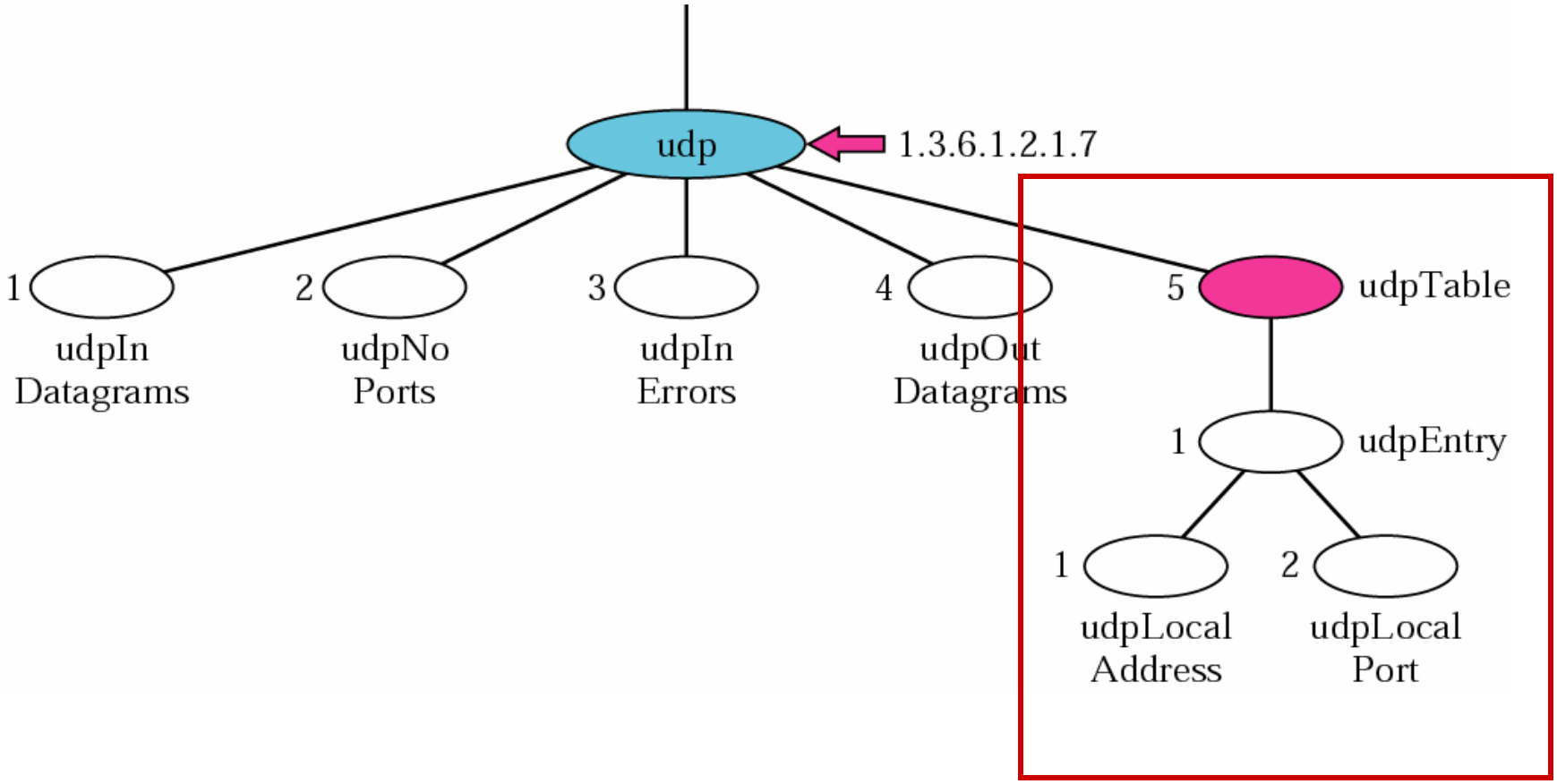


Accessing simple MIB variables



- We use group ID
  - 1.3.6.1.2.1.7
- To get to the **variable**
  - 1.3.6.1.2.1.7.1
- To get to the **instance**
  - 1.3.6.1.2.1.7.1.0
  - .0 is the instance suffix

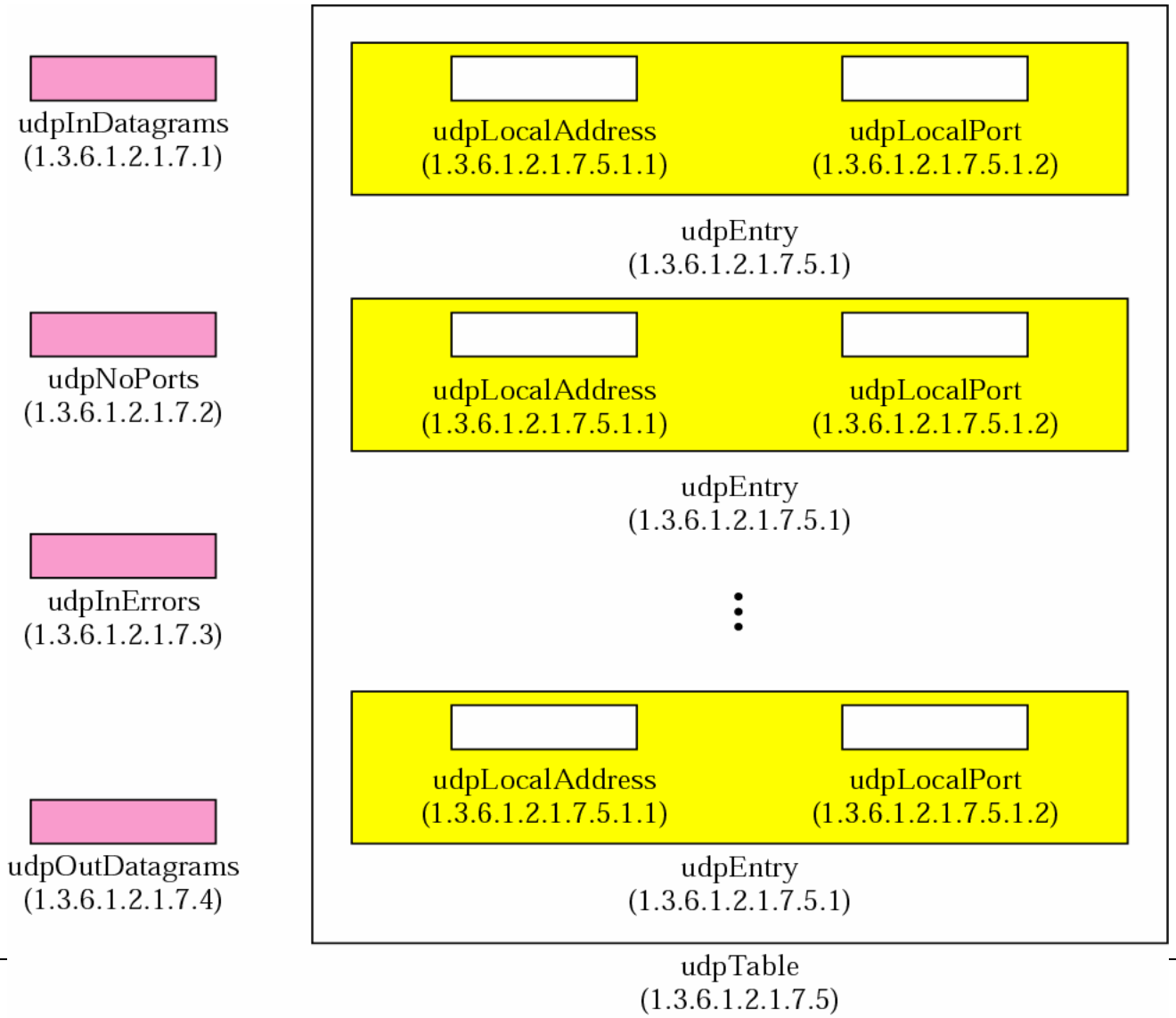
Accessing structured MIB variables



Accessing structured MIB variables

- Since the udpTable is not the leaf node, nor is the udpEntry, we cannot access them for instances
  - 1.3.6.1.2.1.7.5
  - 1.3.6.1.2.1.7.5.1
  - 1.3.6.1.2.1.7.5.1.1
  - 1.3.6.1.2.1.7.5.1.2

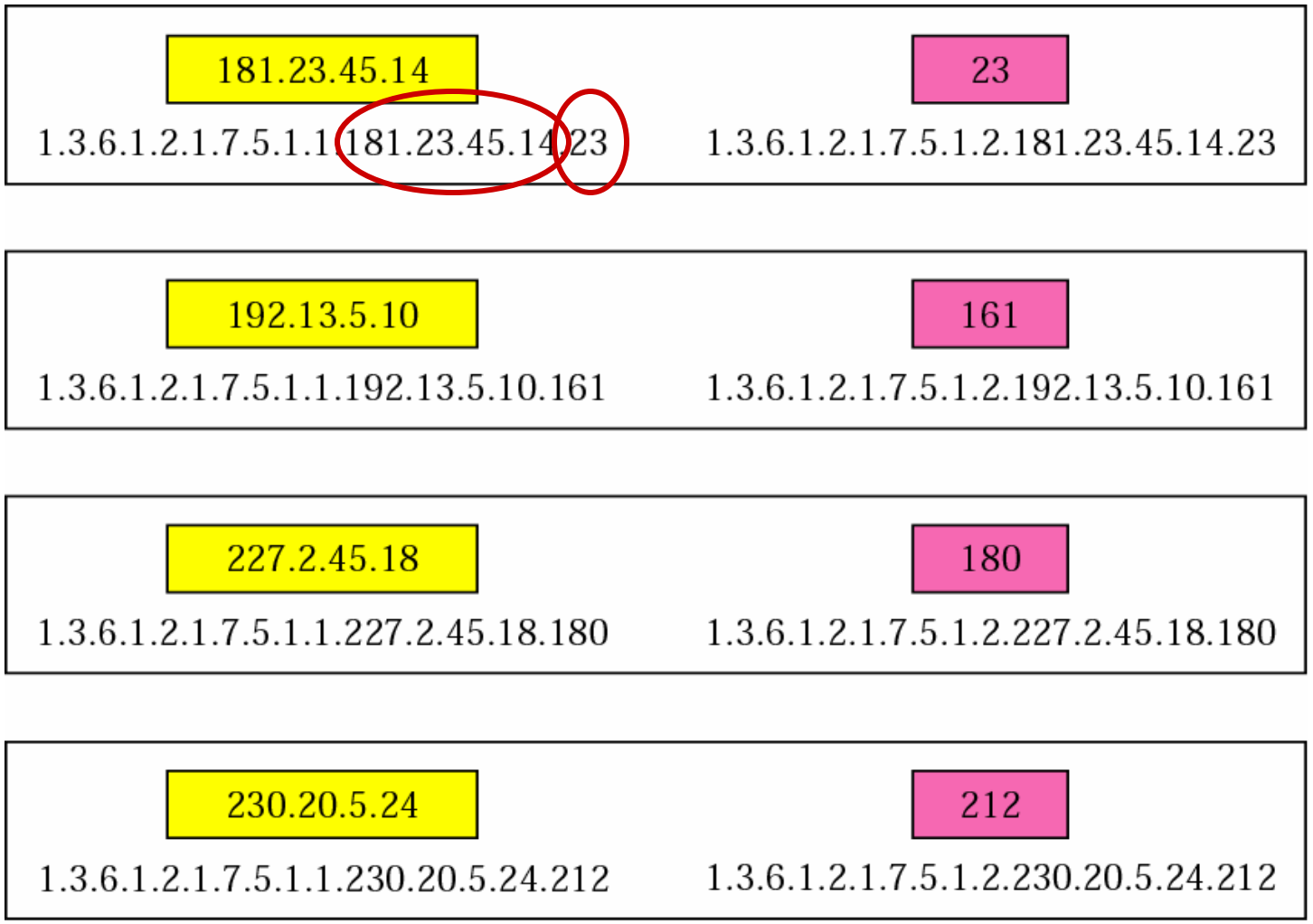
Accessing structured MIB variables



Accessing structured MIB variables

- Though we can access the instances, but we need to define *which* instance
  - We need the table index
    - Can be based on one, two or more fields

Accessing structured MIB variables

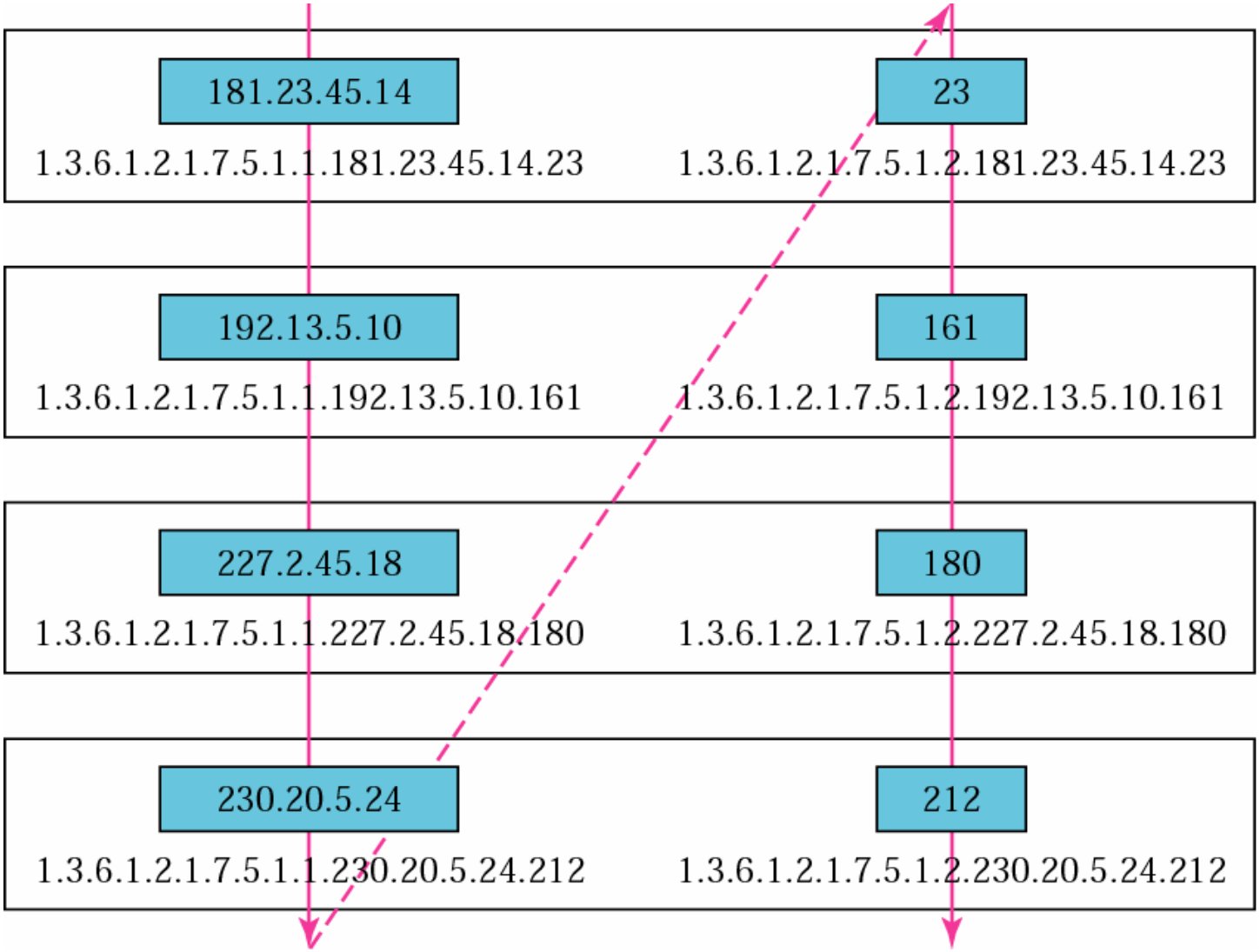


Lexicographic ordering

- Object identifiers follow lexicographic ordering
  - Tables are ordered according to column-row orders
- Use SNMP command `GetNextRequest`

lexicographical order, (a.k.a. **dictionary order** or **alphabetic order**)

Lexicographic ordering



Questions?

That's all for today!