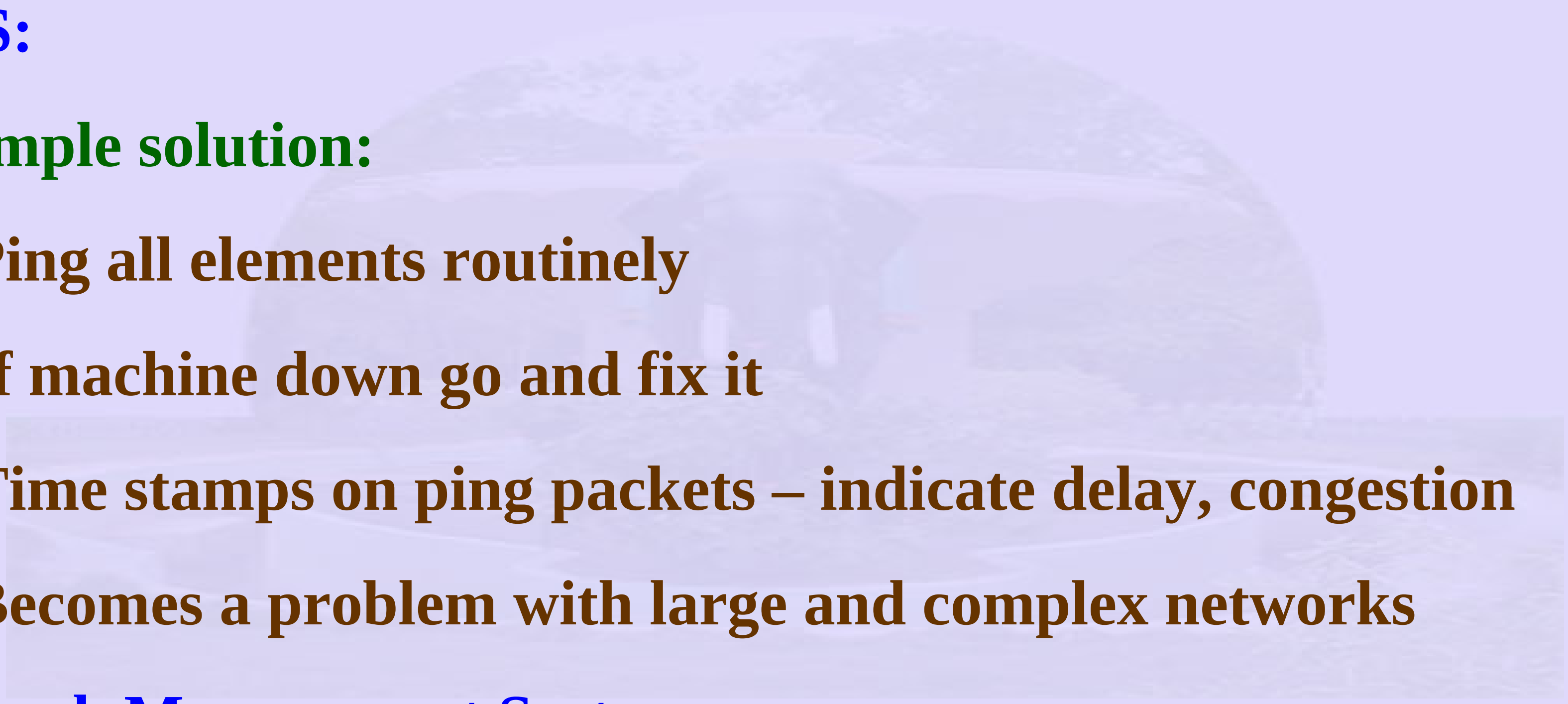
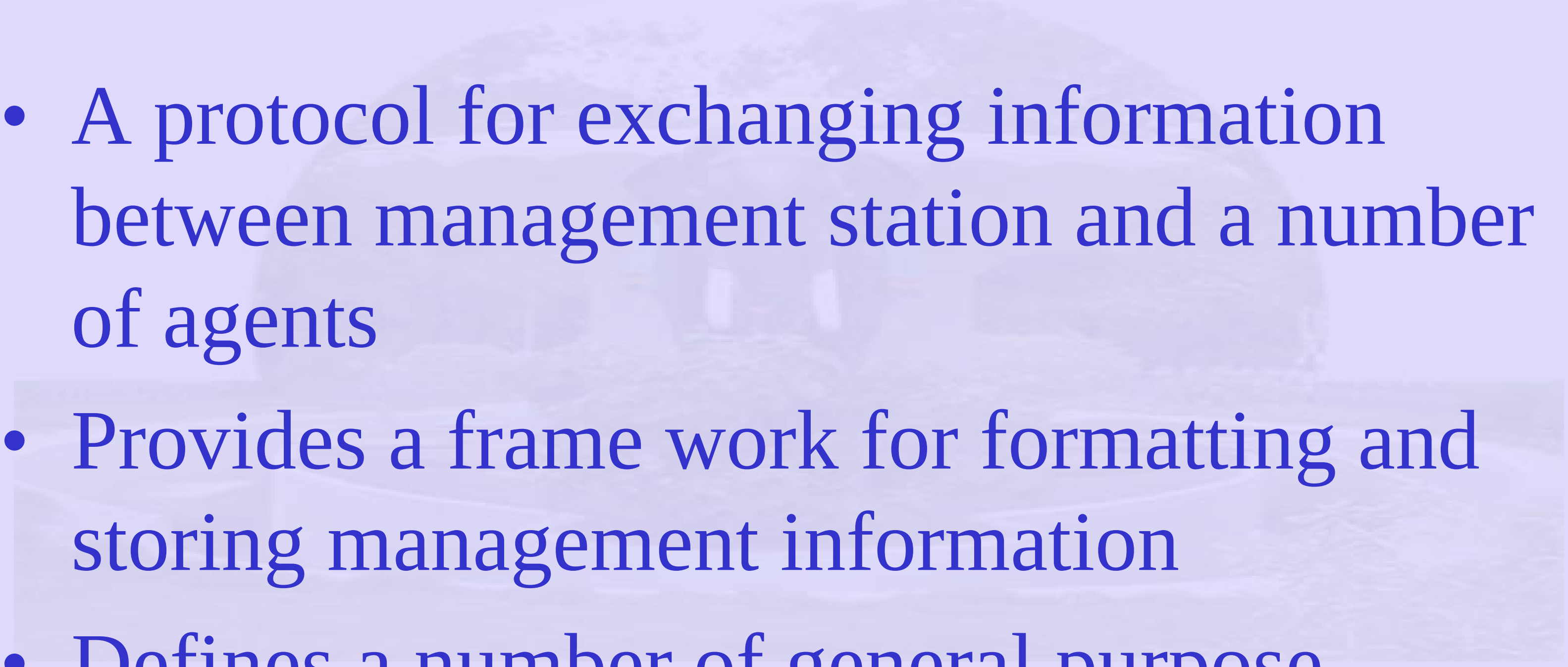
Network Management System

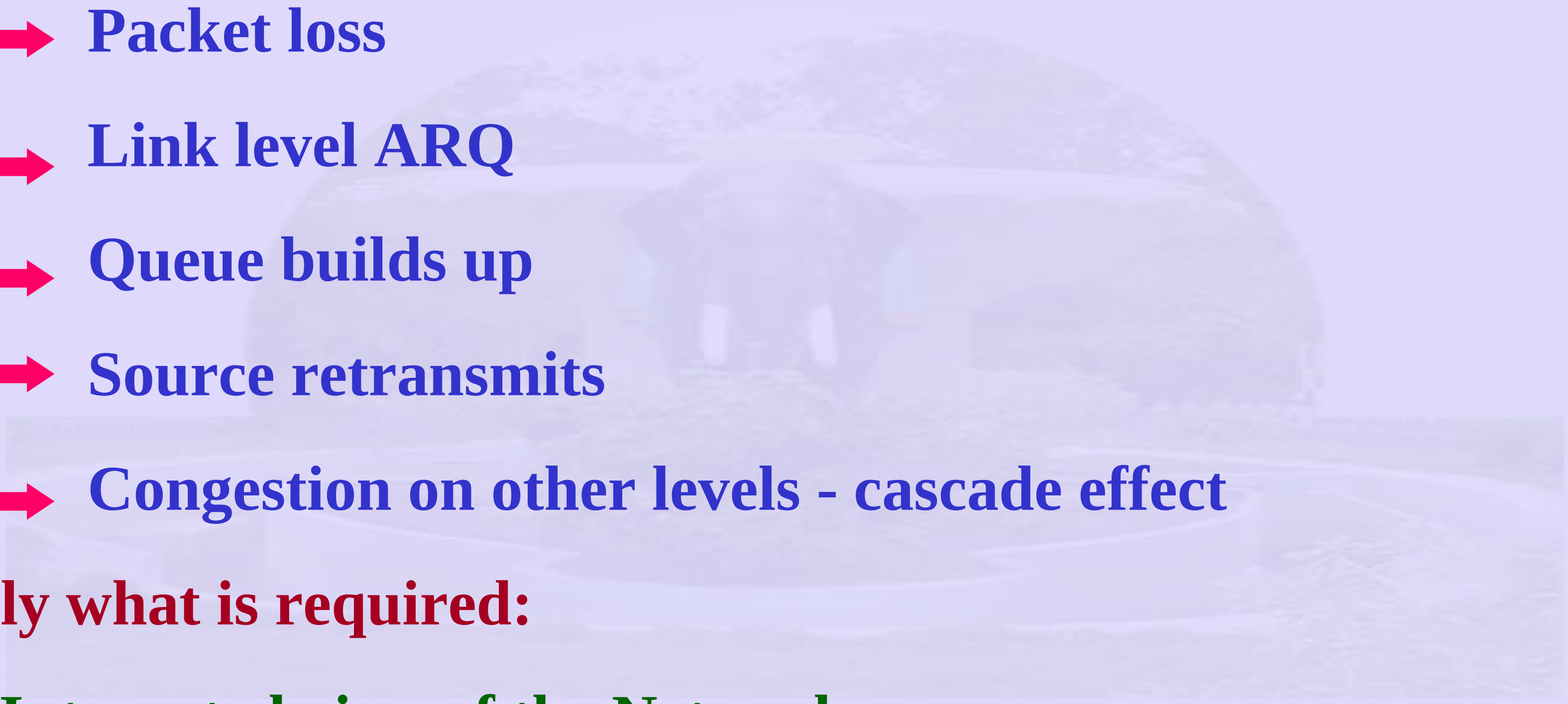


- **NMS:**
- **Simple solution:**
  - **Ping all elements routinely**
  - **If machine down go and fix it**
  - **Time stamps on ping packets – indicate delay, congestion**
  - **Becomes a problem with large and complex networks**
- **Network Management System:**
  - **Remote monitoring and control of the network**
  - **Complex Network – failure in one part can affect the rest of network, for example Network storms**



Simple Network Management Protocol

- A protocol for exchanging information between management station and a number of agents
- Provides a frame work for formatting and storing management information
- Defines a number of general purpose management information variables, objects



Network Management System

* **Example: Noise on a link**

→ **Packet loss**

→ **Link level ARQ**

→ **Queue builds up**

→ **Source retransmits**

→ **Congestion on other levels - cascade effect**

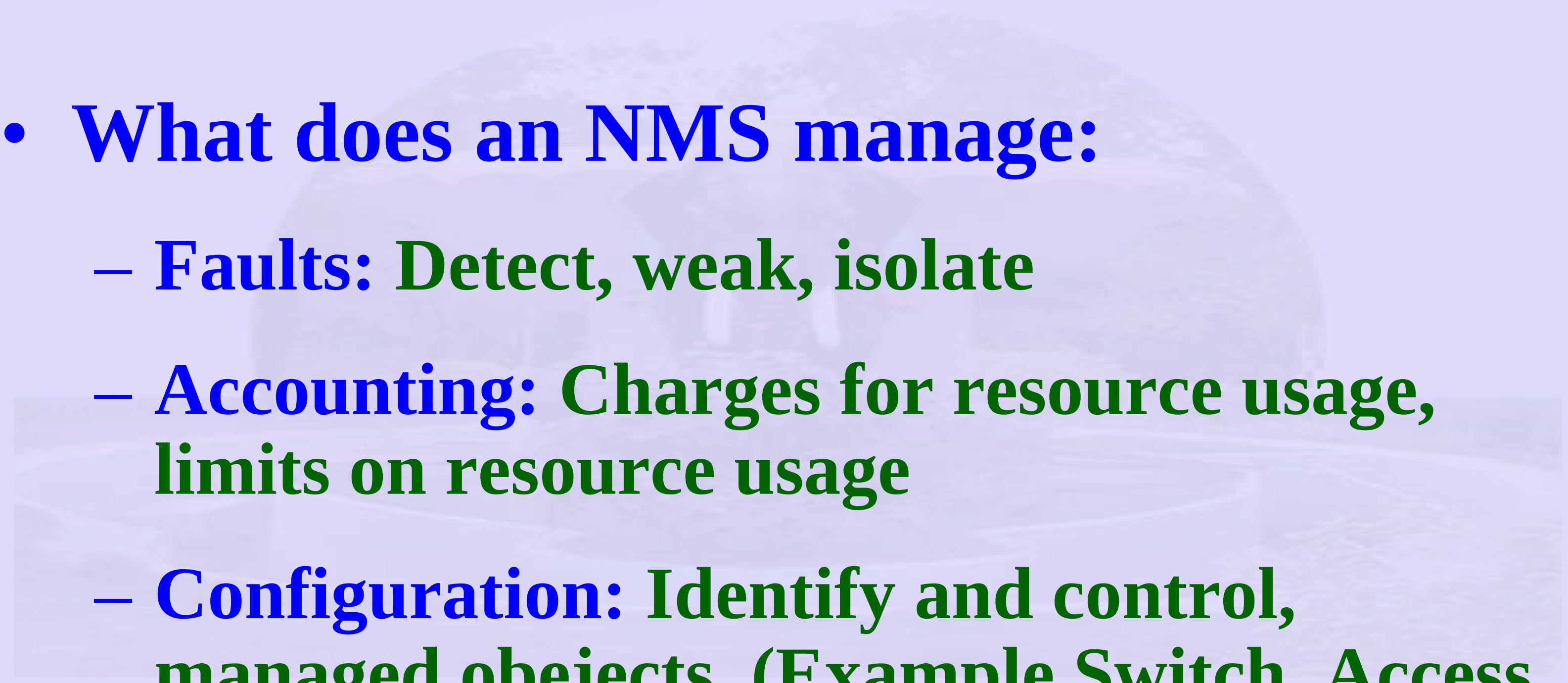
Clearly what is required:

- An Integrated view of the Network

Network Management:

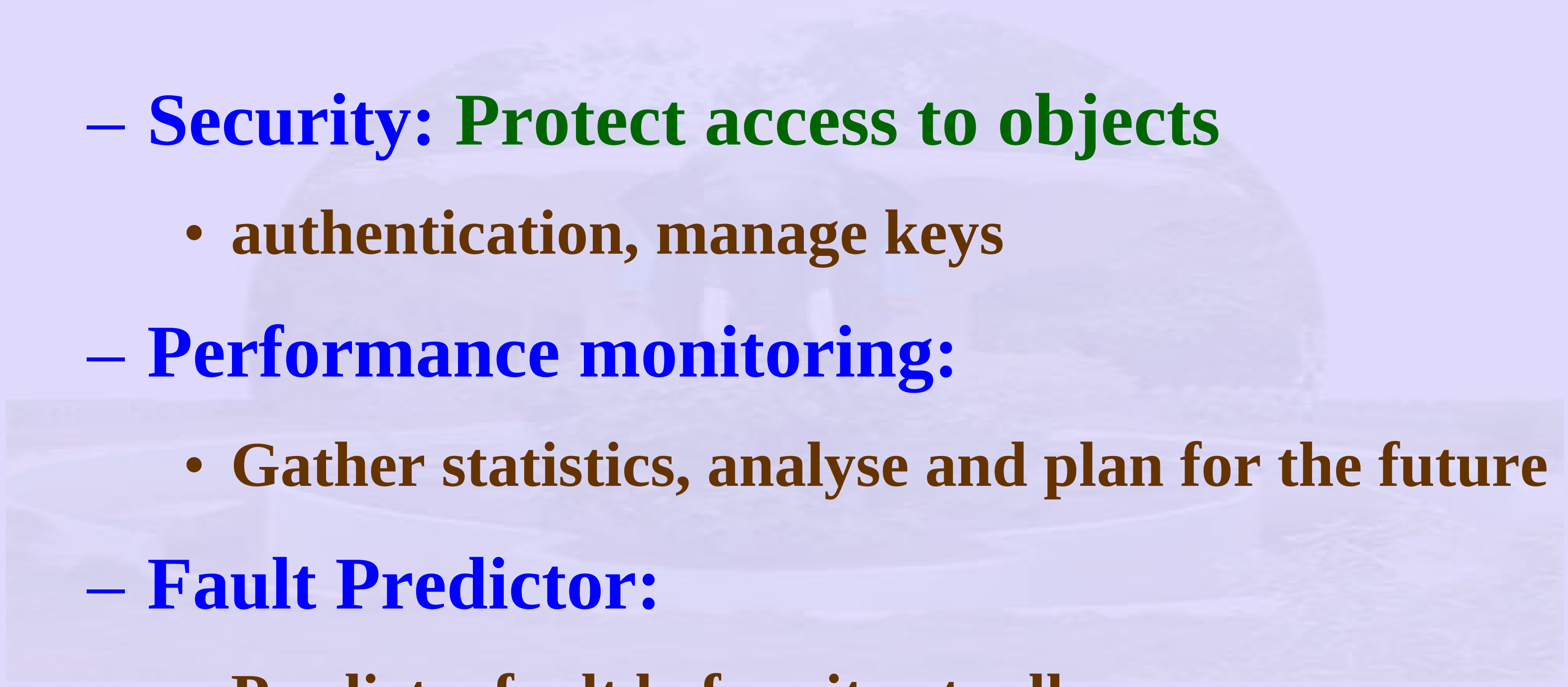
Monitoring and control of a heterogeneous, geographical distributed NEs

Network Manage System (contd.)



- **What does an NMS manage:**
  - **Faults: Detect, weak, isolate**
  - **Accounting: Charges for resource usage, limits on resource usage**
  - **Configuration: Identify and control, managed obejects (Example Switch, Access centre, router)**

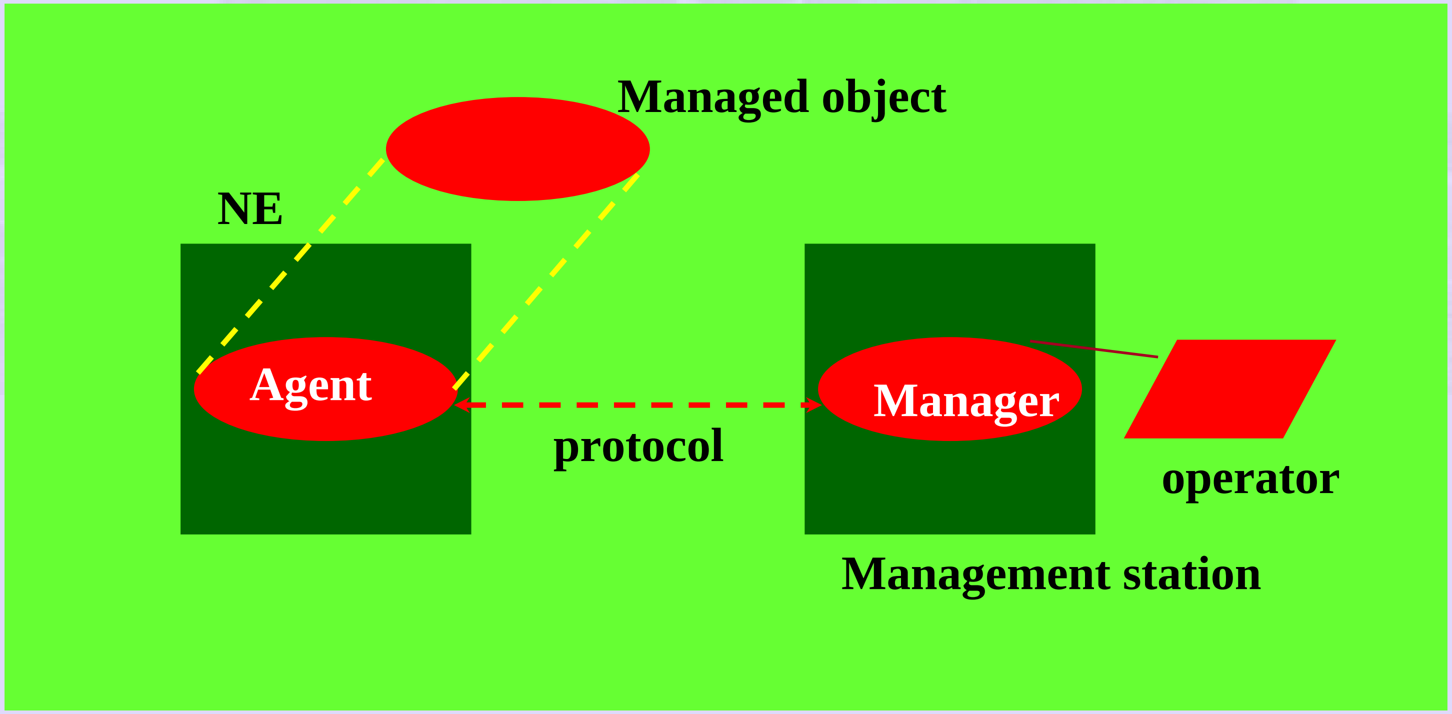
Network Management System (contd.)



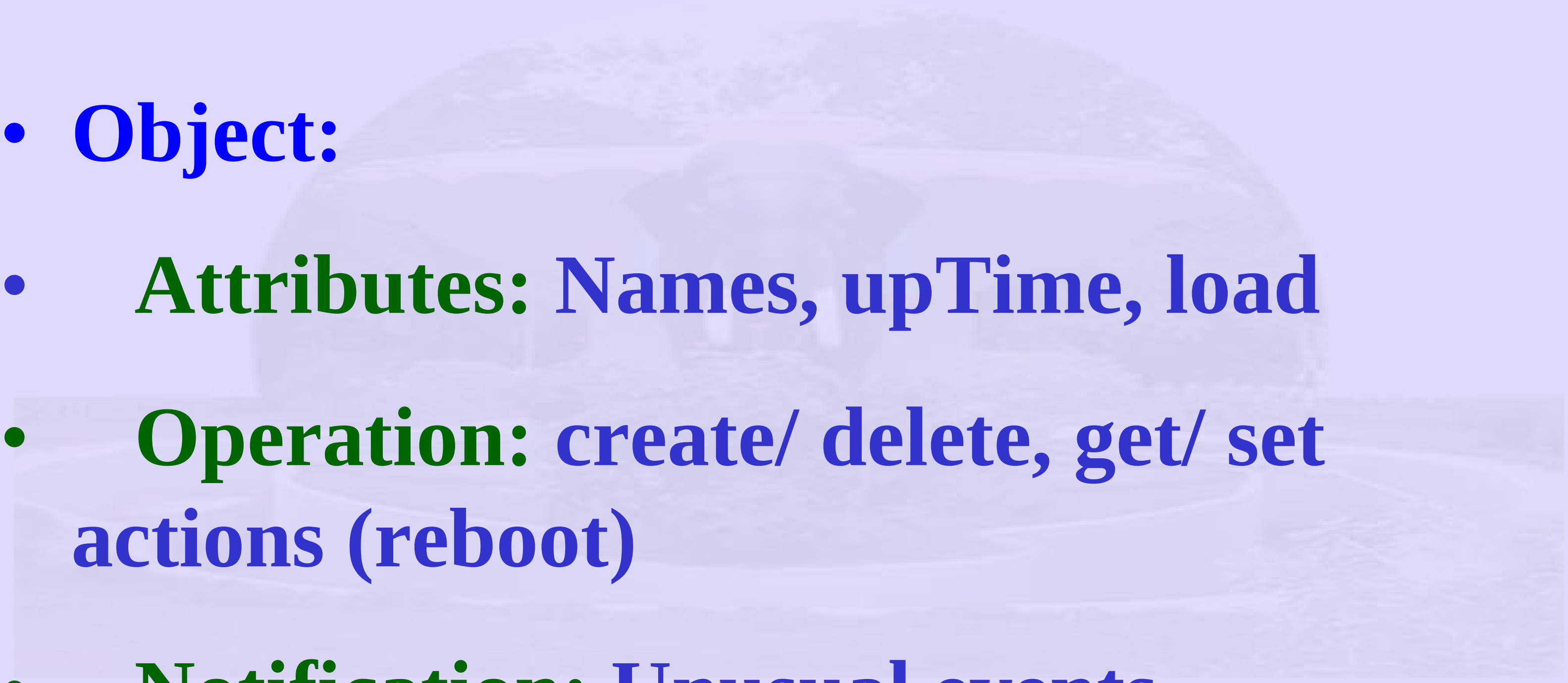
- **Security: Protect access to objects**
  - **authentication, manage keys**
- **Performance monitoring:**
  - **Gather statistics, analyse and plan for the future**
- **Fault Predictor:**
  - **Predict a fault before it actually occurs**

Network Management System (contd.)

How is management done?

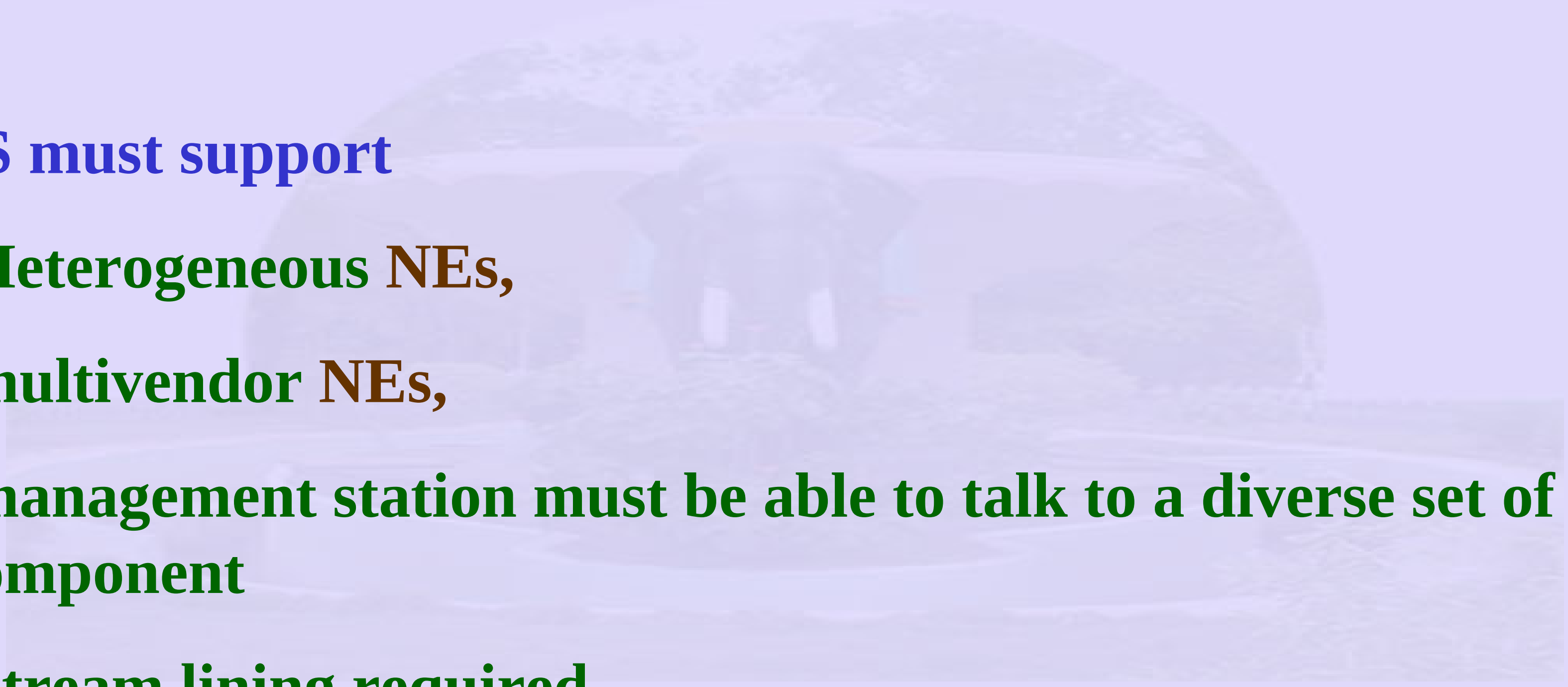


Network Management System (contd.)



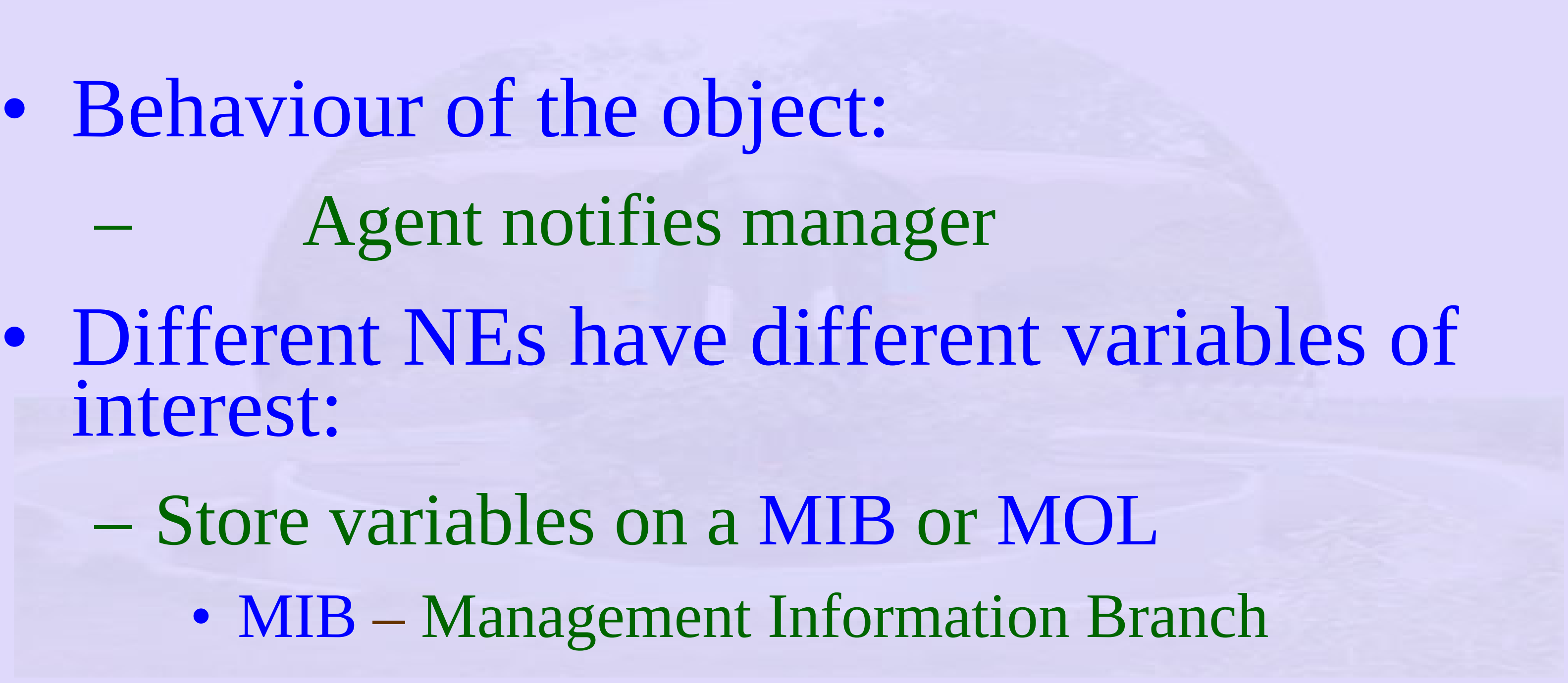
- **Object:**
- **Attributes: Names, upTime, load**
- **Operation: create/ delete, get/ set actions (reboot)**
- **Notification: Unusual events**

Network Management System (contd.)



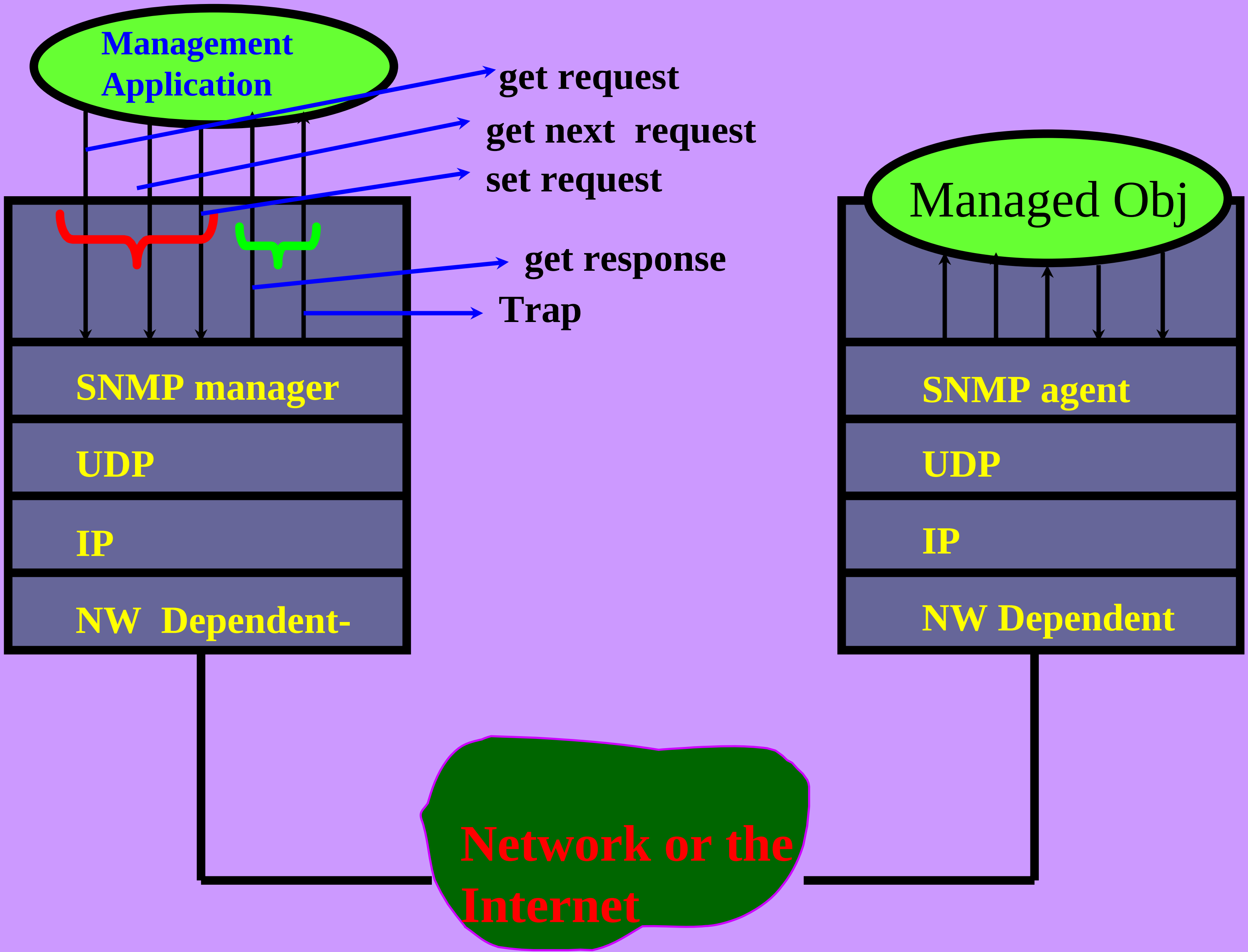
- NMS must support
  - Heterogeneous NEs,
  - multivendor NEs,
  - management station must be able to talk to a diverse set of component
  - Stream lining required
  - Specify information maintained by different devices rigidly

Network Management System (contd.)

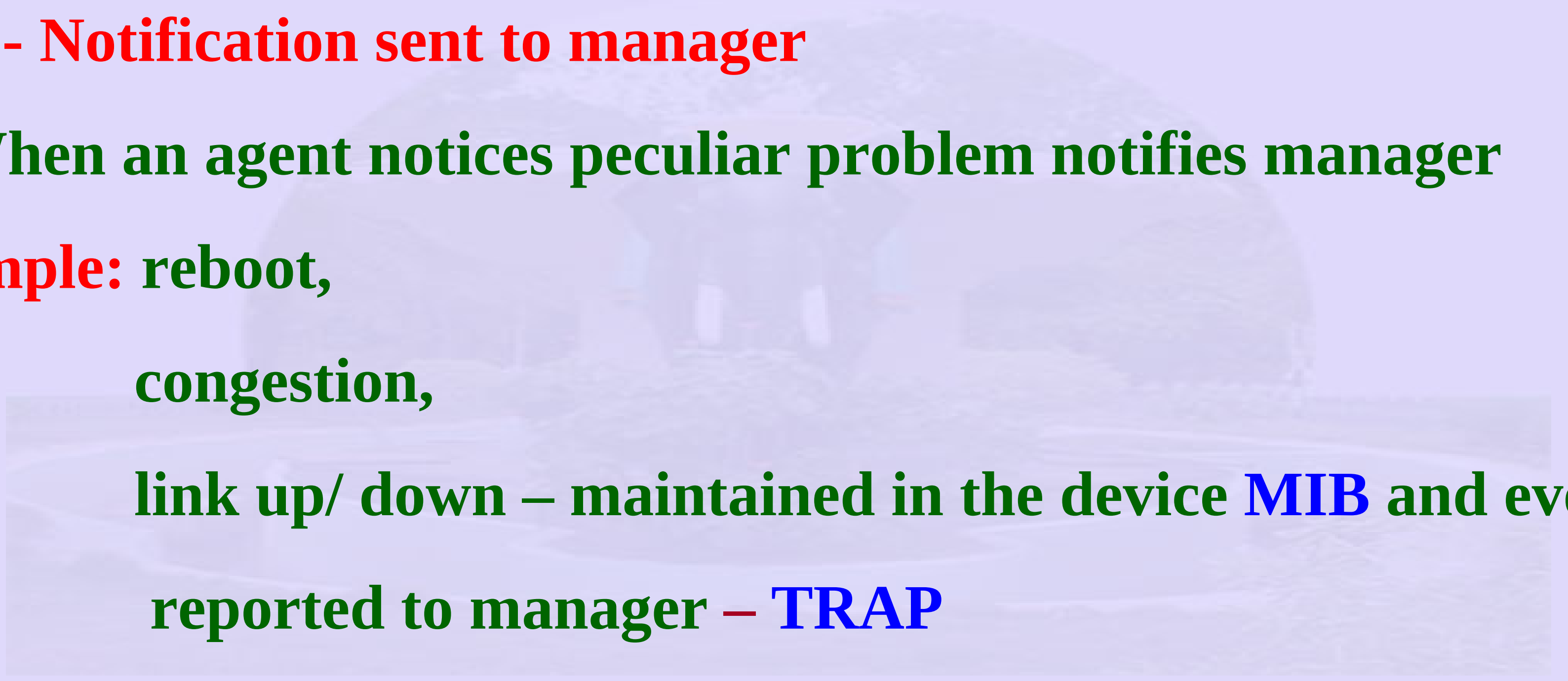


- Behaviour of the object:
  - Agent notifies manager
- Different NEs have different variables of interest:
  - Store variables on a MIB or MOL
    - MIB – Management Information Branch
    - MOL – Management Object Library
- Protocol: Message (PDU) for operations and notification

A typical view SNMP for management



SNMP (contd.)



Trap - Notification sent to manager

When an agent notices peculiar problem notifies manager

Example: reboot,

congestion,

link up/ down – maintained in the device MIB and event reported to manager – TRAP

get – Enables manager to retrieve inform of object at agent

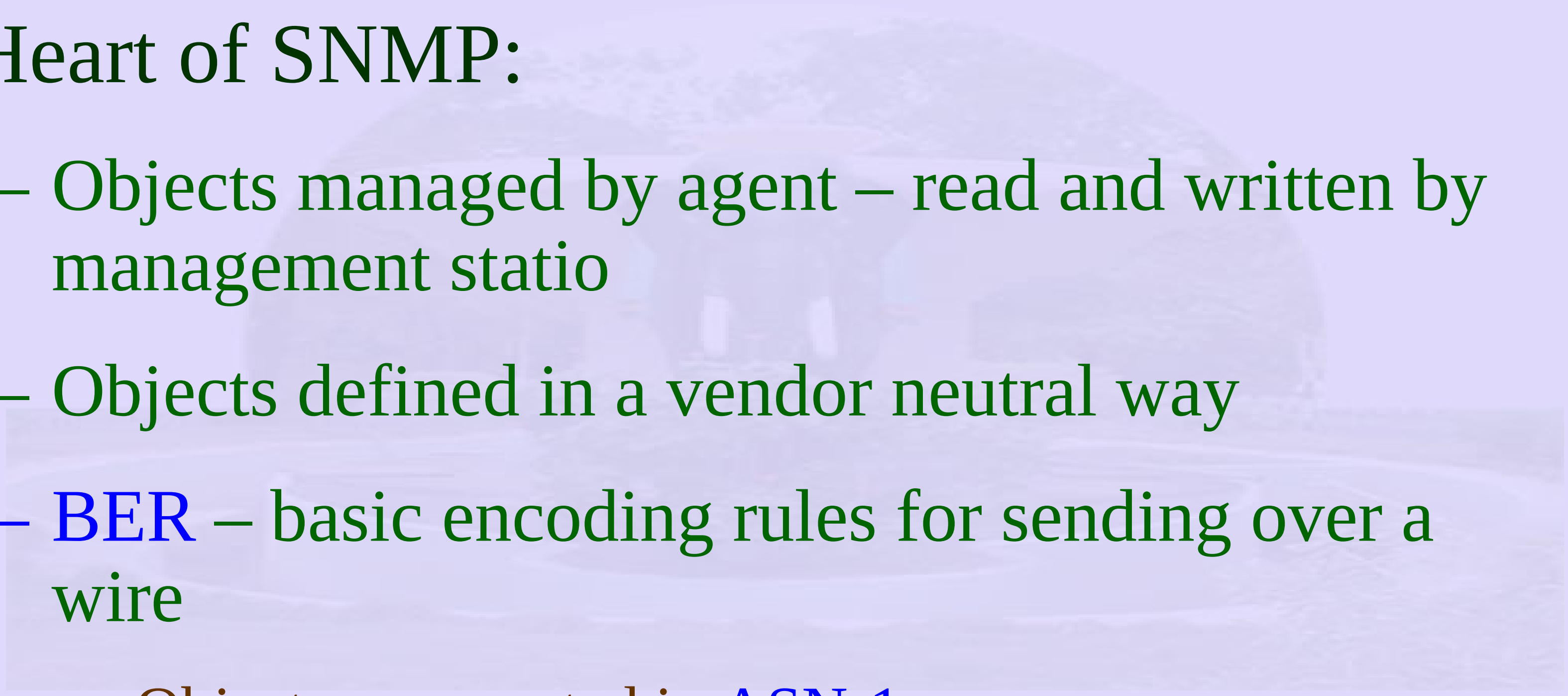
SNMP (contd.)

Proxy agents: SNMP based NMS assume SNMP agent is running on all NEs

Older devices – do not support SNMP

- Support proxy agent, who communicates with manager on behalf of a device

SNMP (contd.)



- Heart of SNMP:
  - Objects managed by agent – read and written by management statio
  - Objects defined in a vendor neutral way
  - BER – basic encoding rules for sending over a wire
    - Objects represented in ASN-1
      - DDL: ISO 8824
      - BER: ISO 8825
      - Data = <type, value>

SNMP (contd.)

Basic Data types allowed in SNMP:

INTEGER: arbit length – Integer

BITSTRING: A string of 0 or more bits

OCTETSTRING: A string of 0 or more unsigned bytes

NULL: A place holder

OBJECTIDENTIFIER: An officially defined type

Count INTEGER ::= 100

STATUS ::= INTEGER {up(I), down(Z), unknown(I)}

OBJECTIDENTIFIER: Provides ways of identifying object

- A standard tree, every object is placed at a unique place in the tree

SNMP (contd.)

Every object in every standard represented by an OID

Construction of new type from basic types:

SEQUENCE – ordered list of type – structure in C

SEQUENCE of - a 1–D array of a single type

Tagging: Creating new types by tagging old ones

Count 32 ::= [APPLICATION 1] INTEGER(0….. 2^{32} – 1)

Gauge32 ::= [APPLICATION 2] INTEGER(0….. 2^{32} – 1)

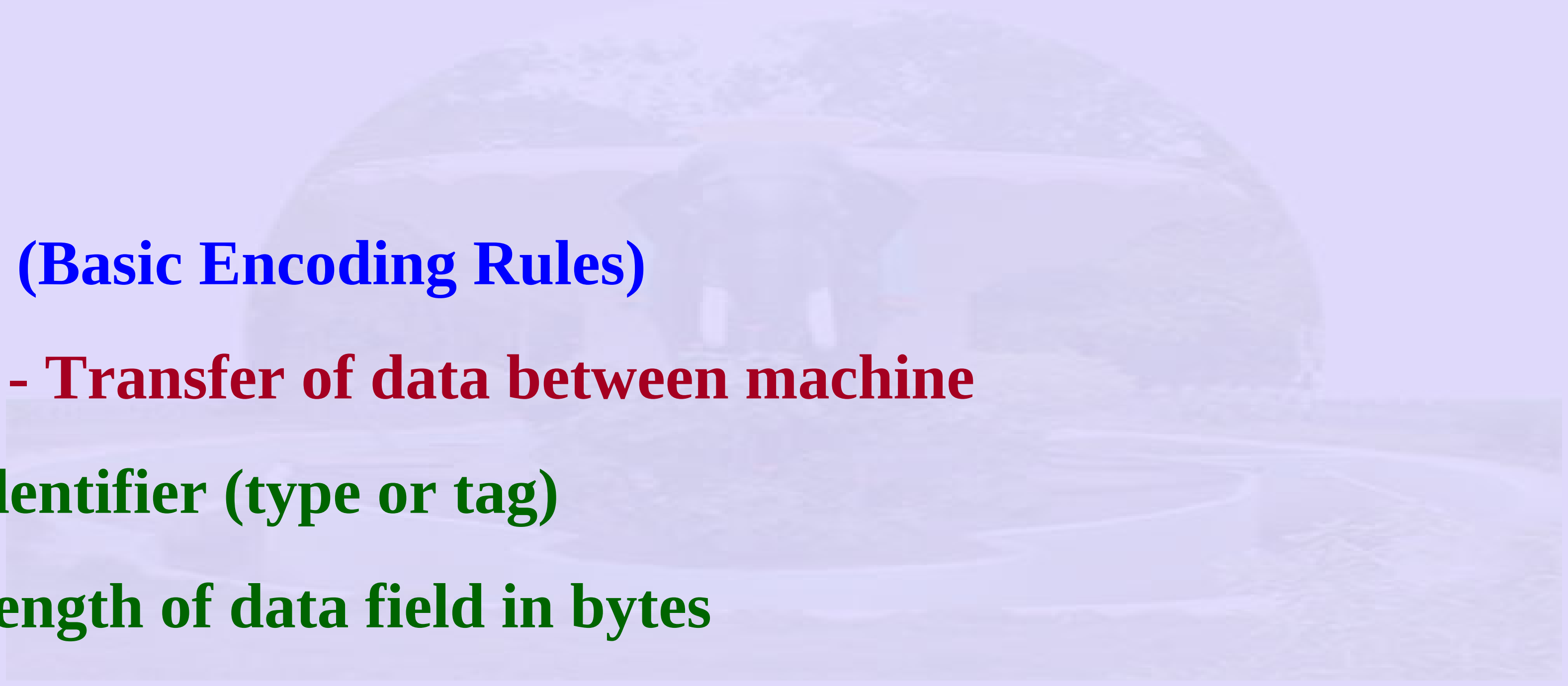
Tags: 4 types

Universal, application wide, context specific and private

ASN 1 Transfer Syntax:

- Define how values of ASN 1 types can be unambiguously converted to a sequence of bytes for transmission

SNMP (contd.)



BER: (Basic Encoding Rules)

- Transfer of data between machine

1) Identifier (type or tag)

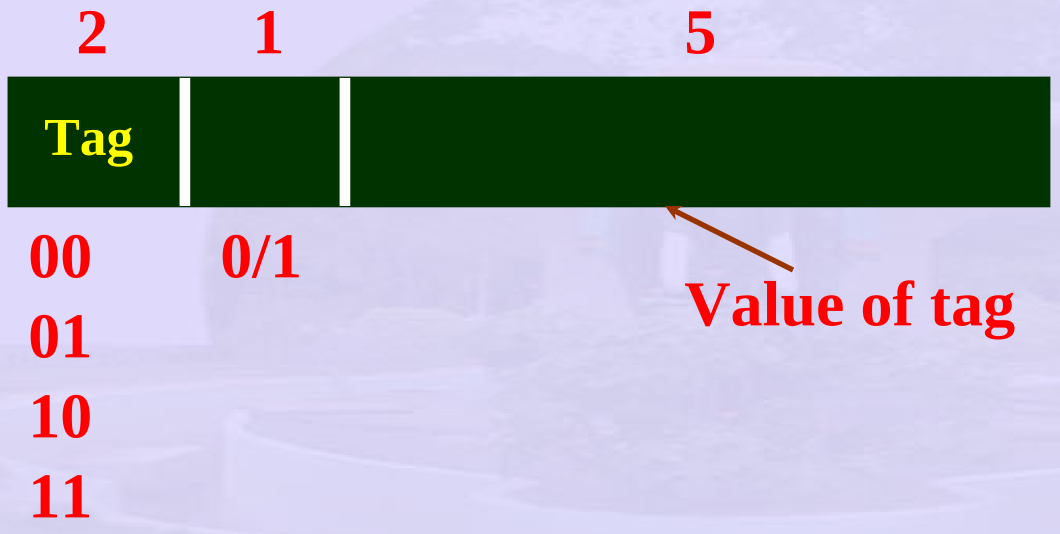
2) Length of data field in bytes

3) The data field

4) End of contents flag, if data length is unknown

SNMP (contd.)

SNMP message format:

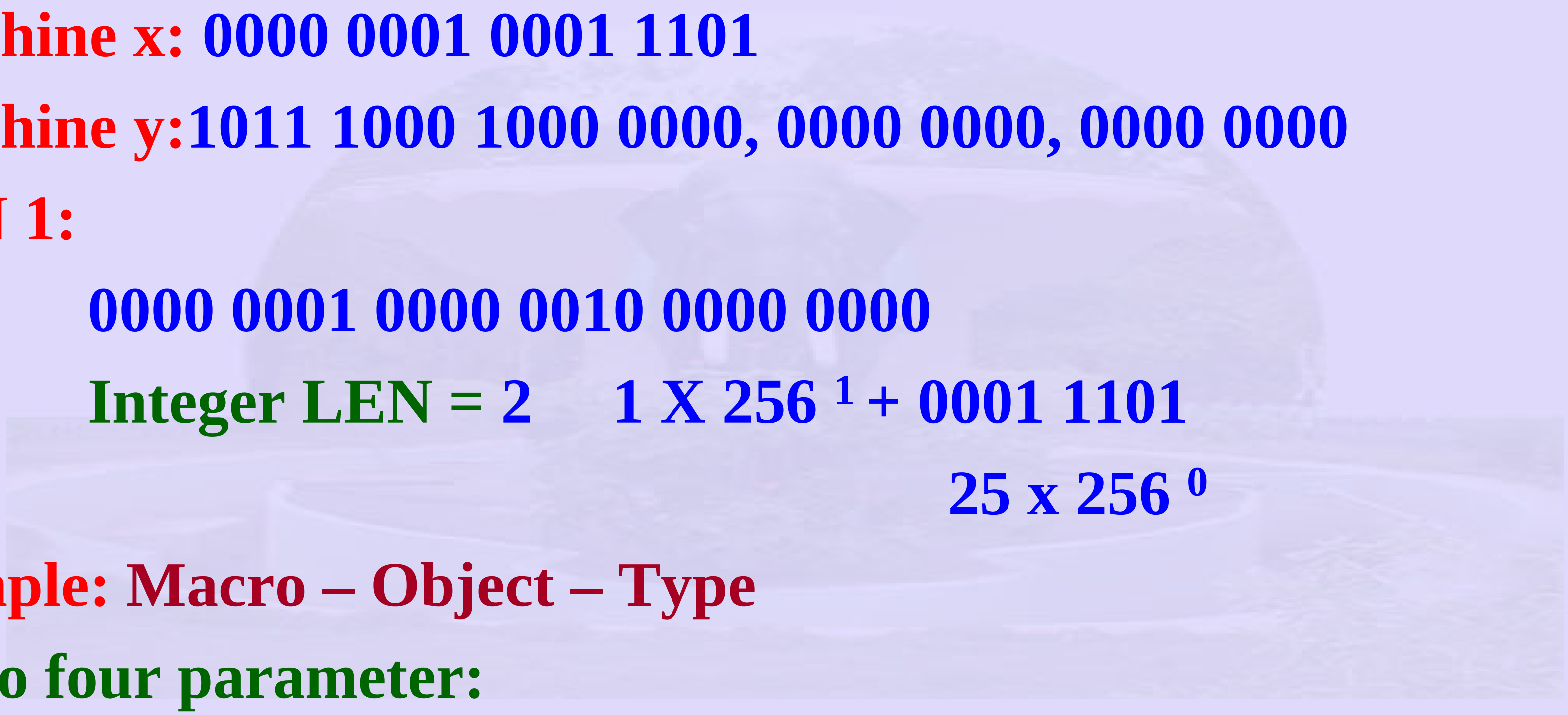


00 – Universal

01 – application wide each standard

10 – limited use in a standard context – specific

11 – not defined by only standard - private



SNMP (contd.)

Example: 285_{10}

Machine x: 0000 0001 0001 1101

Machine y:1011 1000 1000 0000, 0000 0000, 0000 0000

ASN 1:

0000 0001 0000 0010 0000 0000

Integer LEN = 2 1×256^{1} + 0001 1101

25×256^{0}

Example: Macro – Object – Type

Macro four parameter:

lostPackets OBJECT-TYPE

SYNTAX Counter 32 -32 bit counter

MAX-ACCESS Read-only – Cannot be changed by management station

SNMP (contd.)

STATUS current

- Conform with current SNMP

(Obsolete, deprecated, current)

DESCRIPRTION ::= { experimental 20}

position in tree

Representation of Internet object:

$\underbrace{000\ \ 00110}_{\text{OID}}\quad \underbrace{0000\ 0011}_{\text{3 bytes}}\quad \underbrace{00101\ 011}_{\text{40a+b}}\quad \underbrace{00000110}_{6}\quad \underbrace{00000}_{1}$

SNMP (contd.)

Structures of Management information:

- **Define SNMP DS**
- **Lowest level SNMP variable as defined as individual objects**
- **Related objects collected together into groups**
- **Groups collected together as new rules**
- **Uses macro to define new types**
  - **macro notation**
  - **macro definition**
  - **macro instance**

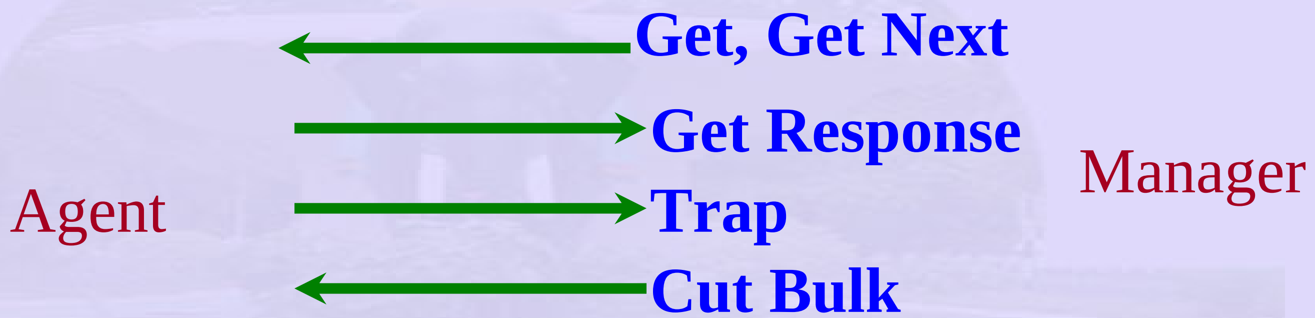
Pair-Integer ::= SEQUENCE (INTEGER, INTEGER,
OCTETSTRING)

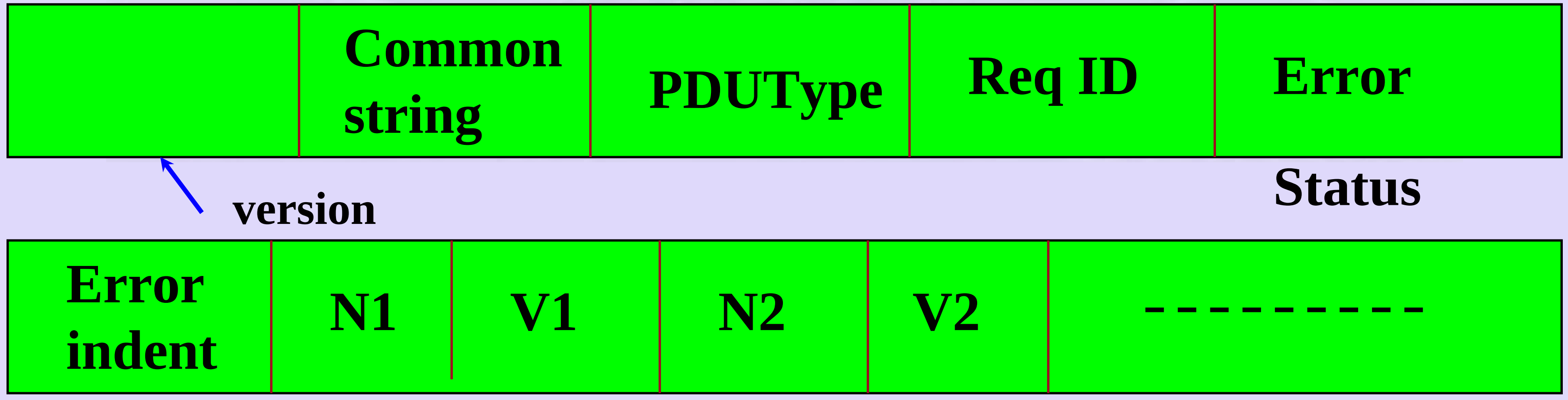
Combining a macro to include any such pair

SNMP PDU

Messages:

Agents and management station exchange PDUs



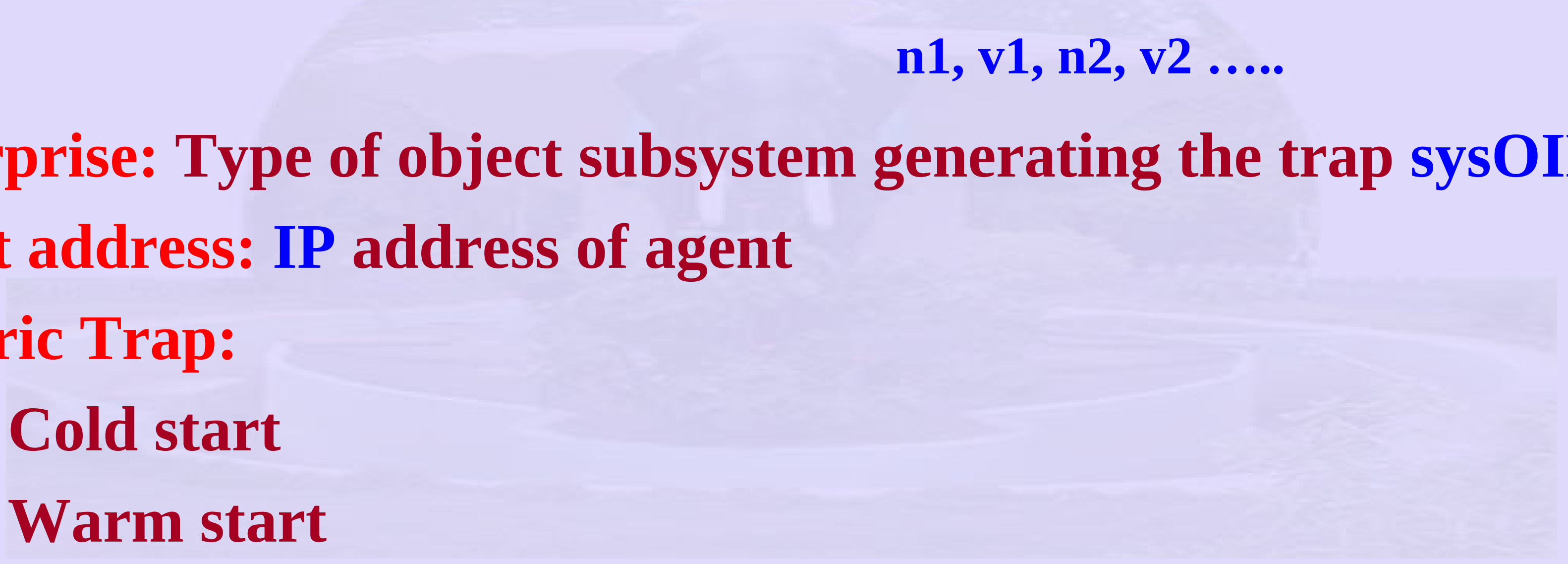


N# - Name, V#- value

SNMP TRAP PDU

PDUType	Enteprise	Agent address	Specific trap	time stamp

n1, v1, n2, v2 …..



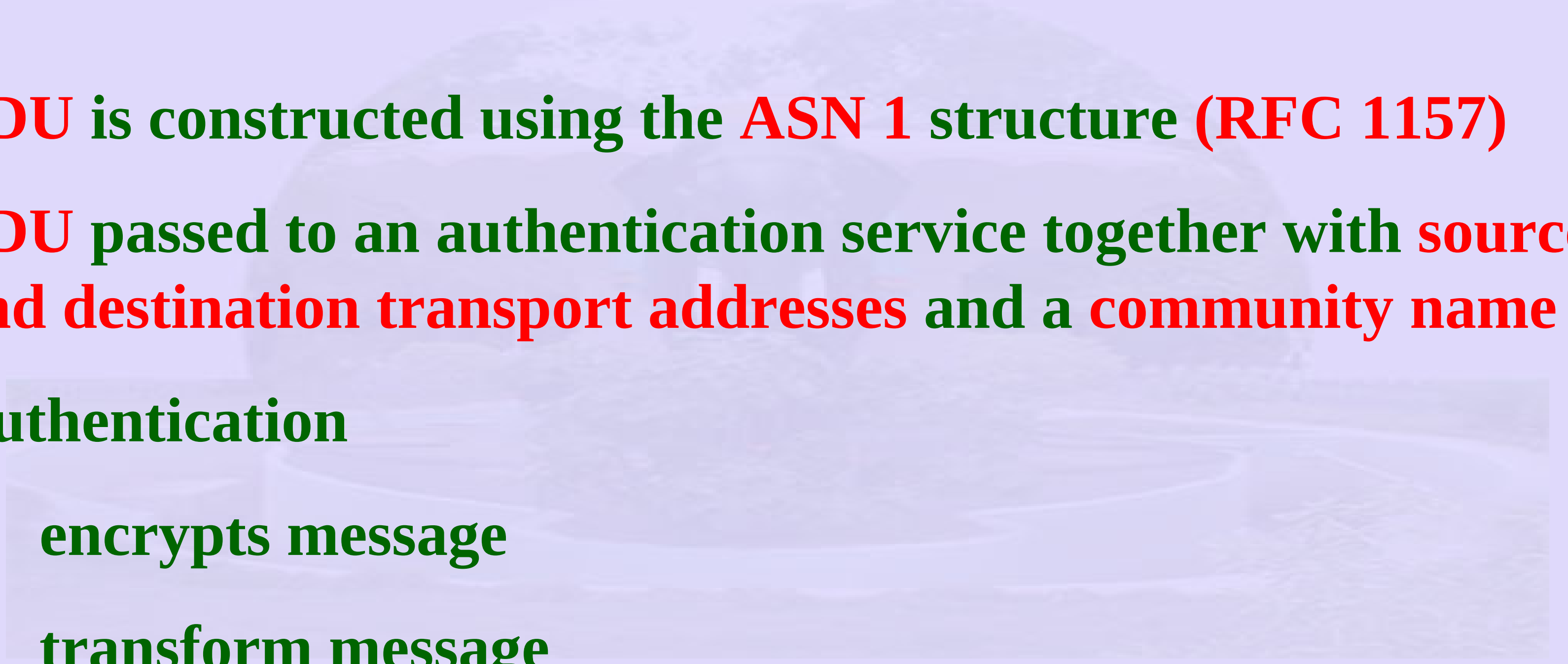
Enterprise: Type of object subsystem generating the trap sysOID

Agent address: IP address of agent

Generic Trap:

- 0 – Cold start
- 1 – Warm start
- 2 – Link down
- 3 – Link up
- 4 – Authorisation failure
- 6 – Enterprise specific

SNMP Message Transmission



- **PDU is constructed using the ASN 1 structure (RFC 1157)**
- **PDU passed to an authentication service together with source and destination transport addresses and a community name**
- **Authentication**
  - **encrypts message**
  - **transform message**
- **Protocol entity constructs a message – version field, community , ...**
- **Object then encoded using BER**