with some of the most important Internet protocols. [RFC 5000] lists all of the standardized MIB modules as of 2008. While MIB-related RFCs make for rather tedious and dry reading, it is instructive (that is, like eating vegetables, it is "good for you") to consider a few MIB module definitions to get a flavor for the type of information in a module.

The managed objects falling under system contain general information about the device being managed; all managed devices must support the system MIB objects. Table 9.2 defines the objects in the system group, as defined in [RFC 1213]. Table 9.3 defines the managed objects in the MIB module for the UDP protocol at a managed entity.

9.3.3 SNMP Protocol Operations and Transport Mappings

The Simple Network Management Protocol version 2 (SNMPv2) [RFC 3416] is used to convey MIB information among managing entities and agents executing on behalf of managing entities. The most common usage of SNMP is in a **request-response mode** in which an SNMPv2 managing entity sends a request to an SNMPv2 agent, who receives the request, performs some action, and sends a reply to the request. Typically, a request will be used to query (retrieve) or modify (set) MIB object values

Object Identifier	Name	Type	Description (from RFC 1213)
1.3.6.1.2.1.1.1	`sysDescr`	OCTET STRING	"Full name and version identification of the system's hardware type, software operating-system, and networking software."
1.3.6.1.2.1.1.2	`sysObjectID`	OBJECT IDENTIFIER	Vendor-assigned object ID that "provides an easy and unambiguous means for determining 'what kind of box' is being managed."
1.3.6.1.2.1.1.3	`sysUpTime`	TimeTicks	"The time (in hundredths of a second) since the network management portion of the system was last re-initialized."
1.3.6.1.2.1.1.4	`sysContact`	OCTET STRING	"The contact person for this managed node, together with information on how to contact this person."
1.3.6.1.2.1.1.5	`sysName`	OCTET STRING	"An administratively assigned name for this managed node. By convention, this is the node's fully qualified domain name."
1.3.6.1.2.1.1.6	`sysLocation`	OCTET STRING	"The physical location of this node."
1.3.6.1.2.1.1.7	`sysServices`	Integer32	A coded value that indicates the set of services available at this node: physical (for example, a repeater), data link/subnet (for example, bridge), Internet (for example, IP gateway), end-to-end (for example, host), applications.

Table 9.2 ♦ Managed objects in the MIB-2 system group

Object Identifier	Name	Type	Description (from RFC 4113)
1.3.6.1.2.1.7.1	`udpInDatagrams`	Counter32	"total number of UDP datagrams delivered to UDP users"
1.3.6.1.2.1.7.2	`udpNoPorts`	Counter32	"total number of received UDP datagrams for which there was no application at the destination port"
1.3.6.1.2.1.7.3	`udpInErrors`	Counter32	"number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port"
1.3.6.1.2.1.7.4	`udpOutDatagrams`	Counter32	"total number of UDP datagrams sent from this entity"

Table 9.3 ♦ Selected managed objects in the MIB-2 UDP module

associated with a managed device. A second common usage of SNMP is for an agent to send an unsolicited message, known as a **trap message,** to a managing entity. Trap messages are used to notify a managing entity of an exceptional situation that has resulted in changes to MIB object values. We saw earlier in Section 9.1 that the network administrator might want to receive a trap message, for example, when an interface goes down, congestion reaches a predefined level on a link, or some other noteworthy event occurs. Note that there are a number of important trade-offs between polling (request-response interaction) and trapping; see the homework problems.

SNMPv2 defines seven types of messages, known generically as protocol data units—PDUs—as shown in Table 9.4 and described next. The format of the PDU is shown in Figure 9.4.

- The `GetRequest`, `GetNextRequest`, and `GetBulkRequest` PDUs are all sent from a managing entity to an agent to request the value of one or more MIB objects at the agent's managed device. The object identifiers of the MIB objects whose values are being requested are specified in the variable binding portion of the PDU. `GetRequest`, `GetNextRequest`, and `GetBulkRequest` differ in the granularity of their data requests. `GetRequest` can request an arbitrary set of MIB values; multiple `GetNextRequest`s can be used to sequence through a list or table of MIB objects; `GetBulkRequest` allows a large block of data to be returned, avoiding the overhead incurred if multiple `GetRequest` or `GetNextRequest` messages were to be sent. In all three cases, the agent responds with a `Response` PDU containing the object identifiers and their associated values.
- The `SetRequest` PDU is used by a managing entity to set the value of one or more MIB objects in a managed device. An agent replies with a `Response` PDU with the "noError" error status to confirm that the value has indeed been set.

SNMPv2 PDU Type	Sender-receiver	Description
`GetRequest`	manager-to-agent	get value of one or more MIB object instances
`GetNextRequest`	manager-to-agent	get value of next MIB object instance in list or table
`GetBulkRequest`	manager-to-agent	get values in large block of data, for example, values in a large table
`InformRequest`	manager-to-manager	inform remote managing entity of MIB values remote to its access
`SetRequest`	manager-to-agent	set value of one or more MIB object instances
`Response`	agent-to-manager or manager-to-manager	generated in response to `GetRequest`, `GetNextRequest`, `GetBulkRequest`, `SetRequest PDU`, or `InformRequest`
`SNMPv2-Trap`	agent-to-manager	inform manager of an exceptional event

Table 9.4 ♦ SNMPv2 PDU types

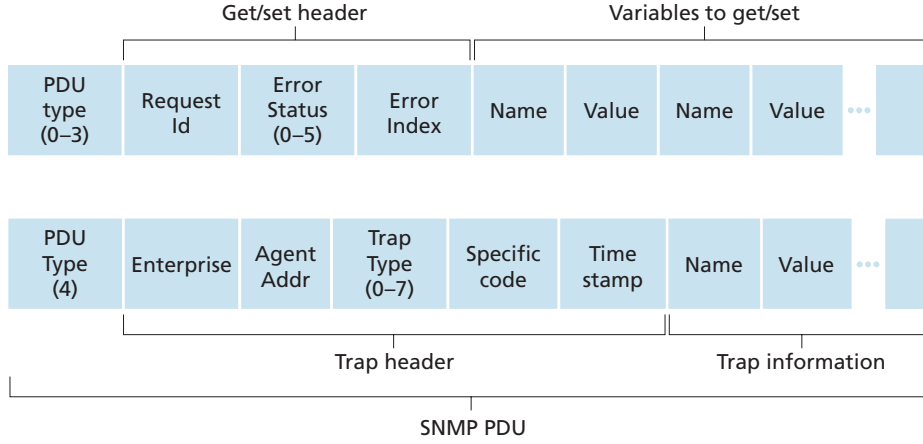

Figure 9.4 ♦ SNMP PDU format