

Network management services using a temporal information model

T. K. Apostolopoulos, V. C. Daskalou

Dept. of Informatics, Athens Univ. of Economics and Business

76, Patission Street, 104 34 Athens, Greece

Tel: +30 1 8203173, Fax: +30 1 8226204,

E-mail: {thodoros, dxv}@aueb.gr

Abstract

In this paper we address the issue of time as an attribute of the network management information. More precisely, we incorporate the temporal dimension in the management information model proposed by the Internet Engineering Task Force (IETF), as it is described by SMI. The core of the proposed network management information model is the *Temporal Management Information Base (TMIB)*, a conceptual representation of the diachronic behaviour of network resources. We define the architecture as well as the services supporting our view. Finally, we give the exact definition of one key service using ASN.1 formulation.

Keywords

temporal management information base, temporal network management services, modelling and interpretation of management information

1 INTRODUCTION

The today's picture of networking is composed by a very heterogeneous environment supporting multivendor applications upon a variety of underlying switching systems and transmission facilities. The need to control these complex and heterogeneous networks has introduced the concept of network management. A lot of research effort has been given in order to solve problems arising in this area and to establish standards that could be used across a broad spectrum of product types (e.g. hosts, routers, bridges, switches, telecommunication equipment) in a multivendor environment.

The general architecture of a network management system is based on a client-server architecture, where the server is called agent, while the client is the manager. Each network component has an agent which maintains a local Management Information Base (MIB). The MIB is a conceptual representation of the network resources that

provides the network manager with the ability to observe and control the current behaviour of network elements. The manager and the agents can communicate through a network management protocol such as the SNMP (Case, 1990) or CMP (ISO 9596) protocols. The interaction between the manager and the agents admits the retrieval and/or update of the MIB information in a way enabling the implementation of various network management functions.

Time is an attribute of most real world phenomena, and in this paper we address the issue of time as an attribute of the network management information. More precisely, we incorporate the temporal dimension in the management information model proposed by the Internet Engineering Task Force (IETF), as it is described by SMI (Rose, 1990). A similar approach is presented elsewhere (Shvartsman, 1993). The core of the proposed network management information model is the *Temporal Management Information Base (TMIB)*, a conceptual representation of the diachronic (past and current) behaviour of network resources. The temporal network management model and the TMIB design are presented in detail in section 2. The architecture that implements the proposed information model as well as the general services are described in section 3. In section 4 we present an analytical view of one significant temporal management service. In section 5 we discuss the main issues of our approach.

2 TEMPORAL NETWORK MANAGEMENT INFORMATION MODEL

Two different information models were defined in order to represent the network management information: one from ISO (ISO 10165) and another one from IETF (Rose, 1990). Both models are centered around the so-called managed objects, which represent an abstraction of network resources and form the Management Information Base (MIB). The IETF's model classifies the managed objects in two types: *scalar objects* and *table objects*. The table objects are two dimensional arrays of scalar objects and at a given time they consist of multiple row entries. The scalar objects are simple MIB variables which can have at most one instance at a given time. The management information contained in the MIB can be classified according to the various criteria as it is described in (Apostolopoulos, 1996) and (Harista, 1993). In particular, taking into account the frequency that management information variables change their values, we conclude to two broad classes of objects (Apostolopoulos, 1996):

- *Quasistatic objects*, which describe the current network configuration (e.g. the number of host interfaces, the routing table, etc.) and their values do not change very often. For example, in MIB-II (McCloghrie, 1991) we can characterise as quasistatic objects the information contained in the `system group`, the `ifTable` (information about the interfaces on a host), the `ipRouteTable` (routing table), etc.
- *Dynamic objects*, which are related to network events (e.g. the transmission of packets) and their values change very often during time. This information involves mainly the objects that are of type `Counter` and `Gauge`.

In this paper we propose a model that provides a diachronic representation of the network management information. The model uses a Temporal Management

Information Base, TMIB in order to represent the past and current behaviour of network resources. This new information model is based on the IETF's model and it extends it in order to include the temporal nature of the management information. The incorporation of the temporal dimension in the IETF's model admits the adoption of the TRDM temporal database model. The TRDM, that is presented in (Snodgrass, 1987), extends the standard relational data model by timestamping each row.

According to the new temporal management information model proposed in this paper, the TMIB is a collection of historical table objects that represent a diachronic view of the management information. The TMIB tables represent the past and current network state. The TMIB consists of two types of historical tables: *interval tables* and *event tables*. Interval tables consist of a set of explicit columnar management information objects and of two implicit time objects. These time columnar objects, *validFrom* and *validTo*, represent the time interval [validFrom, validTo) during which the state of the management information is valid. Event tables consist of a set of explicit columnar management information objects and of one implicit time object *validAt*. This time object refers to the instant that an event described by the explicit columnar objects took place. In order to extend the IETF's management information model we should transform the scalar and table objects that constitute the IETF's MIBs into TMIB tables. For this purpose we use the following rules:

1. The MIB table objects are mapped onto corresponding TMIB tables. The explicit non-temporal columnar objects of the derived TMIB table represent the columnar objects that constitute the MIB table object.
2. The scalar objects of MIB groups are mapped onto non-temporal columnar objects of TMIB tables.

nodeID	ifIndex	ifType	ifAdminStatus	ifOperStatus	ifLastChange	Valid time	
						(from)	(to)
pegasus	1	6	1	1	500	570	1100
pegasus	2	15	1	2	800	850	∞
pegasus	3	6	1	2	900	950	1200
pegasus	1	6	1	2	1050	1100	1800
pegasus	3	6	1	1	1180	1200	∞
pegasus	1	6	1	1	1700	1800	∞

Figure 1 Part of the quasistatic ifTable TMIB table.

Taking under consideration the criterion that classifies the network management information into quasistatic and dynamic managed objects, the TMIB tables (interval and event tables), according to the type of managed objects they represent, are characterised as:

1. *Quasistatic tables*: Tables that represent **only** quasistatic management information. The non-temporal columnar objects of a quasistatic TMIB table represents **only** the quasistatic part of the corresponding MIB table or quasistatic scalar objects of MIB groups.
2. *Dynamic tables*: Tables that represent only dynamic management information. The non-temporal columnar objects of a dynamic TMIB table represents **only** the dynamic part of the corresponding MIB table or dynamic scalar objects of MIB

groups. The explicit columnar objects of a dynamic TMIB table represent delta values of the corresponding MIB objects.

The reason for the characterisation of a TMIB table as quasistatic or dynamic is based on the type of operations that can be performed upon each type of table. The information included into quasistatic TMIB tables can be the argument in a retrieve, append, delete or replace database operation, while the information included in dynamic TMIB tables can be an argument only to a retrieve operation.

The TMIB tables (interval and event tables) are historical tables that can be classified in two categories: 1) *simple tables* and 2) *capitalised tables*.

nodeID	ifIndex	ifInOctets	ifOutOctets	Valid time	
				(from)	(to)
pegasus	1	2000	1200	100	130
pegasus	2	4250	3000	100	130
pegasus	3	550	630	100	130
pegasus	1	2050	1320	130	160
pegasus	2	4700	3200	130	160
pegasus	3	400	850	130	160
.....

Figure 2 Part of the dynamic ifTable TMIB table.

Simple tables are constructed and maintained by using the temporal database model augment by the appropriate network monitor and network executor procedures. Our model provides monitor procedures with user-definable polling interval for quasistatic and dynamic TMIB tables. Examples of simple TMIB tables are illustrated in Figures 1 and 2. Figure 1 illustrates a part of the quasistatic ifTable TMIB table. It is an interval table that represents the quasistatic part of the MIB-II ifTable managed object. In this table we can see the history (the past and the present) of the values taken by each columnar managed object. For example, we can see that during the time interval [570, 1100) the interface with ifIndex = 1 was up, during the interval [1100, 1800) it was down and since time 1800 until now, e.g. during the interval [1800, ∞), is again operational. Each row lists the values of the columnar objects during the time interval [*validFrom*, *validTo*). The current information is represented with rows having the *validTo* time object equal to ∞ . Figure 2 illustrates a part of the dynamic ifTable TMIB. It is an interval table representing a dynamic part of the MIB-II ifTable for node “pegasus”. As this is a dynamic table, each row represents delta values of the columnar managed objects during the interval [*validFrom*, *validTo*). For example, the first row in this table says that during the interval [100, 130) the node “pegasus” at the interface with ifIndex =1 received 2000 octets and sent 1200 octets.

Capitalised tables serves the need for capitalisation of the network management information because of the vast amount of data included in simple tables (especially in dynamic tables). This type of tables is constructed by applying the appropriate aggregate operations on simple TMIB tables. Based on the simple historical table of Figure 2, an example capitalised TMIB table representing the history of the total number of ifInOctets received by all the interfaces of node “pegasus” is illustrated in Figure 3. In order to produce this capitalised table we determine the periods of time

for which no new rows were inserted in the table presented in Figure 2. For each such period we calculate the sum of `ifInOctets` from the rows that were valid during that period. For example, during the interval [100, 130) the total number of `ifInOctets` is 6800 because it is calculated only from the rows {`pegasus`, 1, 2050, 1320, 100, 130} and {`pegasus`, 1, 4700, 3200, 175, ∞} that were valid during that interval.

nodeID	sumTotalIfInOctets	Valid time	
		(from)	(to)
pegasus	6800	100	130
pegasus	7150	130	160
....

Figure 3 Part of a dynamic capitalised TMIB table.

3 ARCHITECTURAL MODEL

In this section we will describe the architectural model and the services that incorporate directly the proposed network management perspective. In order to present this architectural model we define two new objects: the *temporal network management system (TNMS)* and the *temporal agent (t-agent)*. We define as TNMS an object that manages a *Temporal Management Information Base (TMIB)* which represents diachronically the global network state. A t-agent is an object physically located in a specific network node which has the responsibility of collecting network management information of all the network elements belonging to its *area of responsibility* and storing this information in a number of TMIB tables. The proposed network management architecture consists of one TNMS and a number of t-agents as illustrated in Figure 4.

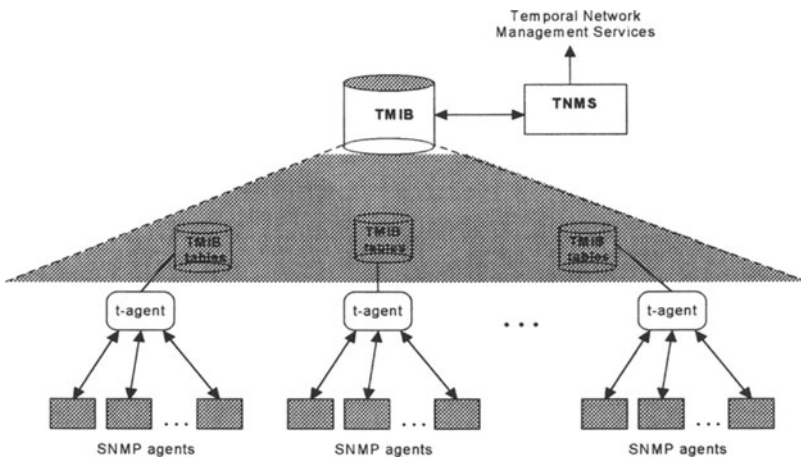


Figure 4 The proposed network management architecture.

The TMIB design follows the distributed model because it consists of the total number of TMIB tables that reside in the t-agents constituting the entire managed network. The communication between the TNMS and the t-agents does not need a specific network management protocol. This can be accomplished directly because the TNMS can consider that the TMIB tables residing on every t-agent are tables constituting a distributed database. All the well known techniques and protocols borrowed from the distributed database era may be used in order to achieve a consistent global state of the overall network.

The architectural model admits a user-definable set of t-agents that should be established in the network management environment. More precisely, the network administrator should establish each t-agent implementation in a specific network node and define its area of responsibility. In order to define the area of responsibility of each t-agent, the network manager should divide all the network components and/or the networks that he wants to manage in a number of non-intersecting sets of SNMP agents. Each t-agent maintains a number of TMIB tables. These tables describe the diachronic network state that reside in the SNMP MIBs of all the network elements in each own area of responsibility (see Figure 4).

Each t-agent is associated with a unique object identifier that characterises the agent in the entire network. Each t-agent, in order to collect the network management information stored in the TMIB table, uses polling procedures, as well as event-driven mechanisms, that are based on the SNMP protocol.

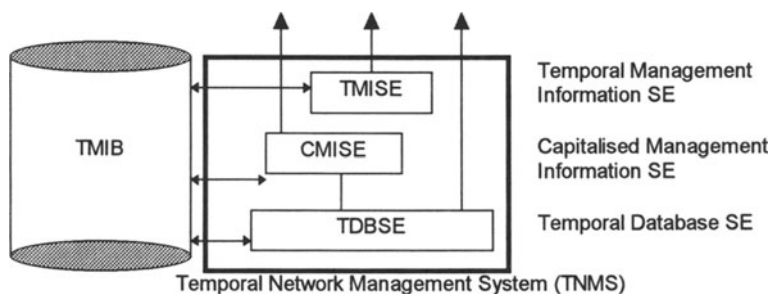


Figure 5 The structure of the Temporal Network Management System.

One important parameter of the TNMS design is to have the network administrator interact solely with the TMIB, so that from the user's point of view the TMIB embodies diachronically the network. This can be done by offering to the user a set of temporal network management services for the monitoring and control of the past and current behaviour of network resources. The TNMS, in order to provide these services, is constructed by the following service elements, as illustrated in Figure 5:

1. *Temporal Management Information Service Element (TMISE)*: It provides services for the definition of objects needed by the TNMS operation. It provides also services for the TMIB tables creation and manipulation.
3. *Capitalised Management Information Service Element (CMISE)*: It provides services for the capitalisation and aggregation of the historical network management information.

4. *Temporal Database Service Element (TDBSE)*: It provides services for the monitoring and control of the historical behaviour of network resources which is stored in the appropriate quasistatic and dynamic simple TMIB table objects.

In the sequel, we will describe the proposed services. The TMISE provides the following services:

Definition services:

- *TM-DEFINE-TABLE*: Service used to define an object that will describe the structure of a TMIB table. More precisely, it is used to define the explicit columnar objects and the implicit time objects that will constitute a TMIB table used by the TDBSE or the CMISE.
- *TM-DEFINE-EVENT*: Service used to define an object that will describe an event and the causes that can trigger this event.
- *TM-DEFINE-FILTER*: Service used to define an object that represents the constrains that can be applied on the values of the explicit columnar objects of a TMIB table (selection constrains) and the constrains that can be applied on the values of the implicit time objects (temporal constrains).

Table creation and manipulation services:

- *TM-CREATE*: Service used to create a TMIB table object associated with the TDBSE or the CMISE respectively. The structure of the TMIB table should have been already defined by the user with the *TM-DEFINE-TABLE* service.
- *TM-COPY*: Service used to create a copy of a TMIB table object as a whole, or to create a copy of some of its columnar objects, or to create a copy of some of its rows.
- *TM-REMOVE*: Service used to drop a TMIB table object as a whole, to delete a number of its explicit columnar objects or a number of table rows.

The CMISE provides the following services:

- *CM-RETRIEVE*: Service used to retrieve the capitalised information stored into TMIB tables objects related to the CMISE.
- *CM-EXECUTE*: Service used to control the procedure that fills an already created CMISE TMIB table with capitalised information coming from the aggregation of past or current information included in TDBSE TMIB tables.

The TDBSE provides the following services:

- *TD-MONITOR*: Service used to control the polling mechanism of network monitors that store the historical network management information in TDBSE TMIB tables.
- *TD-RETRIEVE*: Service used to retrieve the historical information stored into TMIB tables objects. This service allows the user to select past and current views of the network behaviour.
- *TD-APPEND, TD-DELETE, TD-REPLACE*: Services used to change the current state of network resources by creating, deleting and updating rows to the quasistatic TMIB tables.
- *TD-EVENT-REPORT*: Service used to inform the user that an event has occurred. This event can be an already user-defined event concerning thresholds upon values of TMIB table objects or a SNMP trap event.

- *TD-ACTION*: Service used to perform simultaneously a set of operations to different quasistatic TMIB tables. An action is atomic that is, either all or none of the operations have to be done as a unit in order to maintain network configuration integrity.

4 THE TEMPORAL NETWORK MANAGEMENT SERVICES

In this section, due to lack of space, we will present only the TM-DEFINE-TABLE service of the TMISE. This service can be used to define the structure of a TMIB table associated with one of the TDBSE or the CMISE. This service can create the structure for two types of tables:

1. *simple tables*: these TMIB tables can contain simple (not capitalised) past or current network management information and can be associated with the TDBSE.
2. *capitalised tables*: these TMIB tables can contain capitalised past or current network management information and can be associated with the CMISE.

The TM-DEFINE-TABLE operation takes as argument a TMIBStructure which can be of type TMIBSimpleStructure, if the user wants to define the structure of a simple table, or of type TMIBCapitalizedStructure, if the user wants to define the structure of a capitalised table. The result of the TM-DEFINE-TABLE operation is the identifier of the object that defines the table structure. The ASN.1 definition of this service is the following:

```
tm-define-table OPERATION
    ARGUMENT TMIBStructure
    RESULT TMIBStructureID
    ERRORS {invalidStructure, noSuchTableType}

 ::= 1
invalidStructure ERROR
    PARAMETER TMIBStructure
 ::= 1
noSuchTableType ERROR
    PARAMETER TableType
 ::= 2
TMIBStructure ::= CHOICE {
    simple [0] TMIBSimpleStructure,
    capitalised [1] TMIBCapitalisedStructure }
TMIBStructureID ::= ObjectDescriptor
TMIBSimpleStructure ::= SEQUENCE {
    tableType TableType,
    tableColumnarObjects SEQUENCE OF MIBObject }
MIBObject ::= SEQUENCE {
    mibObjectID [0] OBJECT IDENTIFIER,
    mibColumnarObjects [1] SEQUENCE OF OBJECT IDENTIFIER
    OPTIONAL }
TableType ::= ENUMERATED {quasistatic (0), dynamic (1)}
TMIBCapitalisedStructure ::= SEQUENCE {
    tableType TableType,
    tableColumnarObjects SEQUENCE OF TMIBColumnarObject}
```



```

TMIBColumnarObject ::= CHOICE {
    simple      ObjectID,
    aggregate   AggregateObject }
TMIBTableObjectID ::= ObjectDescriptor
ObjectID ::=SEQUENCE {
    tmibTable    TMIBTableObjectID,
    tmibColumn   ObjectDescriptor }
AggregateObject ::= CHOICE {
    count      [0] AggregateTerm,
    countU     [1] AggregateTerm,
    sum        [2] AggregateTerm,
    sumU       [3] AggregateTerm,
    avg        [4] AggregateTerm,
    avgU       [5] AggregateTerm,
    stdev      [6] AggregateTerm,
    stdevU     [7] AggregateTerm,
    any        [8] AggregateTerm,
    min        [9] AggregateTerm,
    max        [10] AggregateTerm,
    first      [11] AggregateTerm,
    last       [12] AggregateTerm,
    var        [13] SEQUENCE {
        aggregateAttribute   EventExpression,
        parameters            AggregateParameters},
    rate      [14] SEQUENCE {
        aggregateAttribute   EventExpression,
        parameters            AggregateParameters,
        per                   Span} }
AggregateTerm ::= SEQUENCE {
    aggregateAttribute   TMIBColumnarObject,
    parameters            AggregateParameters }
AggregateParameters ::= SEQUENCE {
    partition            [0] SET OF TMIBColumnarObject OPTIONAL,
    window               [1] Span OPTIONAL,
    aggregateFilter      [2] FilterObjectID OPTIONAL }

```

The `TMIBSimpleStructure` type has two components: the `tableType`, that specifies the type of the TMIB table as quasistatic or dynamic and the `tableColumnarObjects` that specifies the TMIB table columnar objects. As mentioned in the previous section, the columnar objects of a simple historical TMIB table represent some MIB objects. So, for the `tableColumnarObjects` component we use the `MIBObject` type, which has the `mibObjectID` and `mibColumnarObjects` components. With the component `mibObjectID` we can specify the object identifier of the SNMP MIB table or group that its objects will be depicted on the TMIB table columnar objects. Moreover, with the `mibColumnarObjects` component, which is a sequence of the object identifiers, we can define explicitly which of the MIB objects included in the group or table with object identifier the `mibObjectID` will be depicted on the TMIB table columnar objects. For example, if we want to define a the TMIB table in Figure 2 we should write:

```

tmibTableStructure TMIBSimpleStructure ::=
{
  tableType      dynamic,
  tableColumnarObjects {
    {mibObjectID      1.3.6.1.2.1.2.2 ,
     mibColumnarObjects
      {1.3.6.1.2.1.2.2.1.1, 1.3.6.1.2.2.2.1.10,
       1.3.6.1.2.2.2.1.16}}}
}

```

The `TMIBCapitalisedStructure` type has two components. The first component is the `tableType` that specifies if the table is quasistatic or dynamic. The second component, the `tableColumnarObjects`, specifies the explicit columnar objects of the capitalised table. The `tableColumnarObjects` component is a set of objects of `TMIBColumnarObject` type. The `TMIBColumnarObject` type is a choice of two types of components. This means that a capitalised TMIB object can have as columnar objects the columnar objects of a simple TMIB table or aggregate objects that will contain aggregate information of simple TMIB tables. The `AggregateObject` type is a choice of fifteen different types of aggregate operators as specified in (Snodgrass, 1993). As we can see from the `AggregateObject` type definition, the service user can specify any of the following aggregates:

1. *Simple aggregates*: `count`, `sum`, `avg`, `any`, `min`, `max` and `stdev`. The operators `countU`, `sumU`, `avgU` and `stdevU` are used when we want unique aggregation, that is the aggregation is performed over the set of strictly different values in an columnar object.
2. *Temporal aggregates*:
 - `first`: returns, at each point in time, the oldest value of the given attribute, that is, the one associated with the first valid row
 - `last`: analogous to `first`. It returns at each point in time, the newest value of the given attribute, that is the one associated with the row with the latest *from* time.
 - `rate`: computes the average growth or decrease experienced by values of an attribute over time, and is applicable only to numeric attributes in event relations. The return value indicates growth or decrease per time unit, e.g. Kbits/sec, packets/sec.
 - `var`, `VARIability of time spacing`: computes the degree of inequality of the time spacing within a given set of events

For the definition of the `count`, `countU`, `sum`, `sumU`, `avg`, `avgU`, `stdev`, `stdevU`, `any`, `min`, `max`, `first` and `last` aggregates we use the `AggregateTerm` type. This type has two components: first the `aggregateAttribute` component that represents the columnar object on which the aggregate should apply, and second, the `parameters` component that represents the parameters that control an aggregate operation. The `parameters` component is of type `AggregateParameters` that includes three components:

1. `partition`: component used for the specification of the management objects whose values would be used for the partition of the rows that participate in the aggregate operation.
2. `window`: this component represents the moving window in cumulative aggregates. An aggregate may or may not take into account rows that are no longer valid. By

taking this criterion under consideration, we can divide the aggregates into two types: *cumulative aggregates*, whose values for each point t in time are computed from all rows that have been valid in the past, as well as those valid at t and *instantaneous aggregates*, whose values for each point t in time are computed only from the row valid at time t . For cumulative aggregates, we must specify how far in the past to include rows used to compute a value at time t . For instantaneous aggregates the window component has the value "instant". If all previous rows are to participate, the value "ever" is used. Intermediate cases, such as using only those rows valid during some interval in previous time, are specified using values like "second", "hour", "day", etc. Such aggregates are termed *moving-window aggregates*.

3. `aggregateFilter`: this component is the identifier of a filter object that the service user should have already defined by using the TM-DEFINE-FILTER service. It is used in order to apply specific constraints to the values of the explicit columnar objects and to the values of the implicit time objects that participate in the aggregate operation.

5 DISCUSSION

In this paper, we have proposed a temporal network management information model. Within this framework we have presented the primitive services provided by the model, as well as the architecture that supports them. The main aspects of our approach are:

- The direct incorporation of the time parameter as an attribute of the network management information, in a way that maintains the already existing semantics as far as the information management model is concerned.
- The adoption of the temporal database model for storing the historical network management information as well as the development of well-defined services for doing network management related operations.
- The adoption of a distributed database architecture, a fact that facilitates the easy implementation of the proposed approach. This can be done using well-known from the database era protocols, directly in the implementation of our approach.
- The formal definition of primitive network management services using ASN.1 formulation. This can result in a concrete and compact implementation of our view.
- The possibility for developing easily, complex network management applications based on the primitive services that have been defined.

Our view has been supported by the implementation of a prototype that partially implements the proposed model. In this simplified version we have focused on SNMP MIB-II. The experimental use of this prototype exhibits very positive results as far as the advantages of our approach is concerned. The description of the first version of the prototype is presented in (Apostolopoulos, 1996).

6 REFERENCES

- International Organization for Standardization, Open System Interconnection, *Common Management Information Protocol (CMIP)*, International Standard Number 9596.
- International Organization for Standardization, Open System Interconnection, *Management Information Model*, International Standard Number 10165.
- Apostolopoulos, T.K. and Daskalou, V.C. (1996) Temporal Network Management Model: Concepts and Implementation Issues. *Computer Communications* (in press).
- Case, J.D. Fedor, M.S. Schoffstall, M.L. and Davin J.R. (1990) *A Simple Network Management Protocol*. RFC 1157, DDN Network Information Center, SRI International.
- Haritsa, J. Ball, M. Roussoloulos, N. Baras, J. and Data, A. (1993) Design of the MANDATE MIB in Integrated Network Management III (ed. H-G. Hegering and Y. Yemini) IFIP Working Group 6.6., San Francisco.
- McCloghrie, K. and Rose, M.T. (1991) *Management Information Base Network Management of TCP/IP based internets: MIB-II*. RFC 1213, DDN Network Information Center, SRI International.
- Rose, M.T. and McCloghrie, K. (1990) *Structure of Management Information for TCP/IP based internets*. RFC 1155, DDN Network Information Center, SRI International.
- Shvartsman, A.A. (1993) Dealing with History and Time in a Distributed Enterprise Manager. *IEEE Network*.
- Snodgrass, R.T. (1987) The Temporal Query Language Tquel. *ACM Transactions on Database Systems*, Vol. 12, No.2, 247-298.
- Snodgrass, R.T. Gomez, S. and McKenzie E. (1993) Aggregates in the Temporal Query Language TQuel. *IEEE Transactions on Knowledge and Data Engineering*, Vol.5, No.5, 826-842.

7 BIOGRAPHIES

Theodore K. Apostolopoulos received his diploma in electrical engineering (1979) and his Ph.D. in Informatics (1983) from National Technical University of Athens. Since 1993 he is an Associate Professor in the Dept. of Informatics of Athens Univ. of Economics and Business. His research interests include telecommunication and computer networks, distributed systems and databases. He has more than 30 publications covering the above scientific areas.

Victoria C. Daskalou received her bachelor degree in Informatics from the Dept. of Informatics of Athens Univ. of Economics and Business in 1992, where currently she is a Ph.D. student since 1993. Her research interests include network management, databases and intelligent networks.