Abstract

This paper describes considerations for potential future public data communication services interconnecting users on private local area networks belonging to different administrative domains. The relevant management issues are addressed in an architecture consisting of distributed software modules organized in a hierarchical structure supporting network administration functions. We describe an experimental internetworking unit being built to explore the functional capabilities needed to interface ethernet networks to systems operating at OC-3 SONET/SDH rates of 155.22 Mb/s with basic Asynchronous Transfer Mode (ATM) transport capabilities. We describe bandwidth allocation, address screening, address translation, and traffic monitoring service aspects in the experimental prototype.

1.0 Introduction

High-speed connectionless data services may fulfill the demand for broadband services from customers who need to extend facilities and performance of Local Area Networks to distant areas. Metropolitan Area Networks promise to satisfy the current demand quite soon, thanks to the advanced stage of technology in this field and the large shared bandwidth available on the transmission medium. Taking into account this demand and the service characteristics of Switched Multimegabit Data Service (SMDS) [1], we are exploring open management issues in the context of a connectionless data service using a MAN prototype. The experimental data service can be modeled by an intermediate system consisting of a subnetwork interconnecting two or more end systems or customer subnetworks [2]. The proposed service has the following features:

- Intra-customer (IC) communication enabling bridging of remote LANs belonging to the same customer (a concept similar to closed user groups in ISDN[3]); and inter-customer communication allowing a user with outgoing access to communicate with users in different closed user groups (CUG).
- Security provision based on distributed source and destination address screening.
- Controlled quality of service based on access class; and dynamic bandwidth management.
- Service management functions efficiently exploiting network resources; CUGs maintenance; and traffic data collection and statistical analysis to forecast bandwidth allocation.
- Service management facilities: set up of CUGs and membership handling; flexible bandwidth negotiation; and customer controlled service capabilities.

Interworking in OSI [2], management of closed user groups in ISDN[3], address screening in SMDS, and virtual path connections in B-ISDN[4] were previously described, but, have not been fully implemented in the past. However, these concepts and their interrelation may become more important as network integration requires internetworking of heterogeneous networks and services. The current project re-examines these services concepts in the light of new VLSI devices, high speed subnetworks including large customer networks, and more complete signalling[5] and management[6] protocols. We have demonstrated communication among users within closed user groups, and communications among users of different closed user groups in high speed networks using network management command protocols such as SNMP (Simple Network Management Protocol[6]) rather than ISDN signalling, which would not be fast enough. However, ISDN protocols were used to set up a telecommunication management network (TMN) interconnecting the different network management entities needing to exchange network management commands.

We presume a public backbone MAN switching system offering services to several customers spread among various sites in a single metropolitan area. The resources available to a customer utilizing nodes of the MAN switching system constitute a Virtual Private Metropolitan Area Network (VP-MAN). The experiments are being carried out in the Bellcore METROCORE experimental prototype network [7].

* This work was done at Bellcore as part of a Bellcore-Telettra Collaborative Research Agreement.
2-0 Internetworking Data Services

The current experiment explores a variety of issues related to the definition and implementation of a connectionless data service on a public MAN in a multi-customer environment. There are three main subjects.

First, different data transfer mechanisms have been built to offer a set of different connectivity modes among users communicating by means of the MAN prototype. The SMDS concept of communication security and privacy has been provided by using address screening and validation techniques.

The basic transfer feature of the proposed Data Service consists of two controlled mechanisms for communication among remote LAN users belonging to the same Closed User Group, by means of a connectionless transfer mode. Members of a CUG can either belong to the same customer (Customer CUG, C-CUG) or to different customers (Inter-Customer CUG, IC-CUG). Within a C-CUG a "bridging" mechanism operates as already defined in [8] on the basis of which the METROCORE backbone only works as a remote bridge among those LANs belonging to the same customer. The transfer mechanism ruling IC-CUG communication, which operates alongside the bridging mechanism but implies a more controlled and sophisticated addressing policy, is described in a following section.

More general, four different modes to interconnect CUG users can be obtained by different combinations of the two basic transfer mechanisms and three network destination address types. The four modes are shown in Table 1.

TABLE 1. Internetworking Communication Services

<table>
<thead>
<tr>
<th>Service communication mode</th>
<th>User mobility</th>
<th>Network destination addresses</th>
<th>Mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inter-Location</td>
<td>0</td>
<td>NID</td>
<td>Bridge: yes, IC: yes</td>
</tr>
<tr>
<td>Intra-Customer</td>
<td>1</td>
<td>GID</td>
<td>Bridge: yes, IC: no</td>
</tr>
<tr>
<td>Inter-Customers</td>
<td>2</td>
<td>GID</td>
<td>Bridge: yes, IC: yes</td>
</tr>
<tr>
<td>Intra-Service</td>
<td>3</td>
<td>BID</td>
<td>Bridge: yes, IC: yes</td>
</tr>
</tbody>
</table>

Each mode is given its own network destination address, such as Node Identifier (NID), Group Identifier (GID) or Broadcast Identifier (BID), and the transfer mechanisms it uses, namely IC-CUG ruler (IC) and/or (Bridge). Moreover, the "user mobility" column in the table gives a degree of possibilities that a user has got to move over the network without requiring any destination address modification in data packets. For instance, within the inter-location mode if a station is moved over another network node it will be impossible to send it packets without substituting the NID in them. On the contrary, within the intra-service mode a station can always be reached all over the network transparently by the packet sender station. In-between mobility degrees are possible for the remaining two transfer modes.

Secondly, we have explored and experimented with some management strategies and operations in order to guarantee an efficient exploitation of network resources, such as available bandwidth and screening tables, according to the SMDS concept of Access Path. In particular, an attribute named access class is associated with data exchanges between customers in order to manage bandwidth allocation aspects. The bandwidth management mechanisms allow each customer to negotiate its access class with the public service administrator, so that it can subsequently use its own discretion to re-allocate bandwidth previously allocated among the nodes making up its VP-MAN. Furthermore the same attribute determines a priority parameter to be used in the bandwidth reallocation algorithms of the Data Service, which continuously tries to match network resource availability and customers' requests.

Finally, some added value service functions have been defined as an integral part of the basic concepts of the service, so that each customer can find it easier to access the described service capabilities and adopt flexible inner mechanisms to manage the resources it is assigned.

3-0 Services Management Architecture

The Experimental Data Service Management has been integrated into the service management architecture defined in [6] and depicted in Figure 1. It is organized in a hierarchical structure in which two distinct domains are defined: the public and the private one. The main aim of this organization is to offer a high degree of flexibility in the implementation of service management. Management functions can, in fact, be distributed between the two domains, following specific optimization criteria. One of these would be to give the public domain the task of performing any "customer independent" management functions concerning overall administration of the service, and to assign the private domain the task of executing specific management strategies that each customer can choose for his own users. In this way the public domain is not overburdened by the management choices of individual customers.

The administrator of each domain, whether public or private, consists of a running process and a human agent. The public domain is subdivided into a Delivery Sector in which the Network Administrator (NA) has direct control of network resources, and a Service Sector in which various Service Administrators (SAs) are presumed to manage the various services supplied by the network, each providing the Customer Administrators (CAS) with the set of primitives (Customer Service Management Primitives-CSMPs) relating to the specific type of service it offers.

The private domain is made up of a set of multi-user customers, each of which has a Customer Administrator which virtually manages one or more nodes in the network (Virtual Private MAN, VP-MAN), to provide its users with network services. In reality, the Customer Adminis-
tractor asks the Service Administrators to provide the services, through the set of management primitives.

**FIGURE 1. Management Domain Architecture**

![Management Domain Architecture Diagram]

**3.1 Service Machine Model**

In order to provide each software module previously shown in the Service Management Architecture with a unique and simple development framework, a Service Machine (SM) Model has been defined. It supplies a specific template which allows every experimental service to manage information exchanges with other service machines, so that specific implementations of SM can play, for example, the NA, DSA or CA role.

From the point of view of a single SM any other SM can only assume two different roles: client or resource. A client is a SM asking for an added value service and a resource is a SM providing an added value service. Furthermore, the one who asks for the generation of a SM is called owner and each implemented SM is a part of a hierarchical family, according to the specific role. A particular SM, called Directory Server, manages the directory of a family. The main advantage in using the SM model consists of minimizing the efforts to implement and integrate new management services in a modular and recursive architecture.

Figure 2 shows the functional model of the SM. It highlights four main functions which we will refer to as entities: Resource Management; Client Management; Management Information Base and Core Functions.

- The resources/clients management functions are specific for managing every kind of dialogue with higher/lower level service machines of the hierarchical architecture. These functions include transfer control mechanisms of information and recovery mechanisms of connection failures. A useful subset of these functions works as communication interfaces to higher or lower level.

- The Management Information Base is an entity local to each SM domain, which collects information regarding management network aspects. It is also supplied with maintenance functions of data.

- The core function is a template which an owner can customize by modifying it and adding new specific functions. It is an empty container where new service functionality can be developed by using the basic services offered by template functions and other SM entities.

**FIGURE 2. Service Machine Model**

![Service Machine Model Diagram]

**4.0 Data Service Operational Model**

The Data Service Administrator (DSA) software module of the Service Management Architecture is globally responsible for the creation and maintenance of CUGs, the activation and control of address screening, and the administration of bandwidth assigned to users according to the quality of service required by their customers. For this purpose the DSA processes the request made by customers and, on the basis of optimization algorithms and information contained in the MIB, applies policies of attribution/redistribution of resources. The resources referred to, in
relation to the critical nature of management, are essentially the bandwidth and the memory tables present in each node which store the addresses for forwarding and screening operations related to CUGs. In order to control physical resources the DSA uses, with the aid of the Network Administrator, dedicated local agents through which it can set and/or modify service-related parameters in the access nodes. The DSA also provides customers with value added management functions (CSMPs), thus obtaining a wider distribution of management responsibilities.

Since the DSA has been designed according to the Service Machine Model, it is possible to emphasize every described operation within the entities of its operational model by pointing out the most significant processes and the logical relations among them (Figure 3):

FIGURE 3. Data Service Operational Model

CLIENTS SERVICE MACHINES

CUSTOMER/SERVICE INTERFACE

COMMUNICATION RESOURCES

MIB

CM

BA

RUM

CORE FUNCTIONS

DATA SERVICE MACHINE

RESOURCES SERVICE MACHINES

... Service Request Primitive

... Service Response Primitive

... Database Access

MIB Management Information Base

RUM Resource Usage Monitoring

BA Bandwidth Allocation

CM Closed-User-Group Management

requests to and receive responses from the Network Administrator.

The Management Information Base (MIB) contains information about the amount of Data Service Resources distributed to each customer. The MIB also contains data concerning the extent to which the assigned Data Service Resources are being utilized. A subset of information contained in the MIB is made available to the customers through the CSI. This information provides a Customer with knowledge of traffic statistics for its VP-MAN, a directory of the active CUGs, and so on.

The Core Functions entity contains customized processes needed to implement the new service functionality:

- The Resource Usage Monitoring (RUM) is a running process which continuously monitors the use of resources by each customer and updates the MIB with the relevant data. This entity receives the traffic data from the Network Administrator through the correspondent interface. In this way the RUM process uses a collection service, offered by the NA thanks to the traffic measurement performed by local agents distributed among the network access nodes. Traffic monitoring functions also include processing of statistics data about traffic and errors provision.

- The Bandwidth Allocation (BA) functions receive the bandwidth requests coming from different customers and match them with the DSA resources. It prepares bandwidth requests to send to the NA through the appropriate interface. This function implies a resource allocation strategy.

- The CUG Management (CM) functions is responsible for the creation and maintenance of the CUGs, activation and control of address screening and bandwidth assignment according to the required quality of service.

5.0 Data Service Activities

In this section a more detailed explanation will be given about high-level strategies and operations adopted in designing CM and BA core functions.

5.1 CUG Management operations

As far as the IC-CUGs management is concerned the main phases of the negotiation between public and private domains are represented by the requests for the creation of IC-CUGs, the variation of their membership composition and a directory service.

The relevant CSMPs, which adopt a Request-Response dialogue scheme, are:

1. SET_CUG_ENTRY_REQ/RESP (parameters)
2. MODIFY_CUG_ENTRY_REQ/RESP (parameters)
3. ADD_MEMBER_REQ/RESP (parameters)
4. DEL_MEMBER_REQ/RESP (parameters)
5. CUG_DIR_REQ/RESP (parameters)
The usual information flows among Data Service modules, that rules the provision of IC-CUG service, works according to the following scheme.

A customer can ask the DSA either to create an IC_CUG with one or more other customers or to associate to an already formed IC-CUG (CSMP 1, REQ type). The customer also specifies the amount of resources, namely the amount of memory in the nodes' tables, which is required to satisfy its service needs. The DSA can also refuse the request in case the availability of resources is not as much as needed or if the other customers do not accept the new membership (CSMP 1, RESP type). Once the request is accepted, the customer can ask the DSA to associate one or more of its users to the IC-CUG, managing the entrance rules in its own organization (CSMP 3, REQ type). A similar pattern is followed when the customer wants to remove users from an IC-CUG (CSMP 4, REQ type). The remaining two primitives are used to vary the membership composition of groups (CSMPs 2) and to offer a directory service showing such compositions to customer (CSMPs 5). Insertion and removal operations imply the updating of a Data Service database which duplicates the information contained in the memory tables of the network node, so as to optimize reconfiguration and maintenance tasks. Read and write operations in the local memory tables are executed with the aid of the Network Administrator module, the only one which knows the network configuration of the nodes and can split the request coming from the DSA towards the proper local agents, as it will be described in the next section.

5.2 Bandwidth Allocation Operations

Further DSA operations concern the assignment of bandwidth to each customer as well as the provision of facilities previously mentioned. One of these facilities allows each customer to optimize the sharing of the assigned bandwidth among its users which are connected through the nodes making up the VP-MAN.

The Customer/Service dialogue scheme is based on the following Request-Response primitives:

6. ALLOC_BWIDTH_REQ (BWmin; Pr_Cl)
7. ALLOC_BWIDTH_RESP [status]
8. SET_BWIDTH_REQ [list of (Node_id; BW; Pr)]
9. SET_BWIDTH_RESP [status]

We can split up the dialogue in two different parts. The first one, which uses the CSMP 6) and 7), concerns the request of the global bandwidth, BWmin, and the priority class, Pr_Cl, needed by the Customer to provide users with an assured minimum quality of service. Depending on the resources it owns the DSA decides whether or not to accept the above request (the response value is referred to as the Status parameter in the primitive 7). The second part, which uses the CSMP 8) and 9), allows the customer to use bandwidth management facilities, so that a customer can ask the DSA for dynamic bandwidth partitioning among its own nodes according to the established user priorities. In fact the request primitive 9) contains three specific parameters referred to each node: Node_id is the node identifier; BW is the bandwidth associated to that node; Pr is the priority level for that node. The response associated with the bandwidth partition request, primitive 9), contains a Status parameter whose value is used to indicate whether or not it has been possible to set the correspondent network parameters.

A specific algorithm of the RUM process uses statistical reports to generate request bandwidth parameters to send to the Network Administrator. This action causes the involved Local Agents to set the related node parameters concerning the maximum number of packets (Pmax) which can be transmitted for each data cycle [9].

6.0 Experimental Set Up

In our experiment (Figure 4) we have used the Bellcore METROCORE network research prototype based on a dual bus fiber optic architecture similar to IEEE 802.6. This prototype has programmable data and video service interfaces (Service Processors - SPs) interconnected by a high speed backbone operating at either 140 Mb/s or at 1.7 Gb/s. The Data Service operations are supported by some low level processes running on the LAN SP board [10], in particular on two bit-slice processors (Packetizer and Depacketizer) whose main tasks are to filter incoming and outgoing data packets and change their format. These activities are initialized, monitored and handled by the Local Agent management process running on a dedicated processor.

Three kinds of activities can be distinguished in the implementation of low level operations:

- management of memory tables used by bit-slice processors, supporting CUG Management operations, for packet checking, address screening, and translation;
- on-line traffic filtering and forwarding, which allow the DSA to transfer data packets among users according to the connectivity mode established through CM operations;
- collection of traffic data supporting Bandwidth Allocation operations.

6.1 Low level operations supporting CUG Management activities

Every time a user is admitted to a certain IC-CUG, the Packetizer and Depacketizer of partner organizations need to store their Ethernet and the MAN addresses in relevant tables, by using the information contained in command PDUs coming from the service administrator through the management software chain. The MAN address in our experimental set-up is the 16-bit MAC address used by the METROCORE network.

Each bit_slice processor has two banks of Content Addressable Memory (CAMs), which can perform parallel searches through its 48-bit memory locations very quickly, so as not to lose any Ethernet packet. CAM1 banks support "transparent wiring" operations. CAM2 banks store
For instance, whenever an ADD_MEMBER_REQ primitive is issued by a customer the CM process splits the request, through the aid of the Network Administrator, towards the local agents of the involved access nodes.
In order to perform the dynamic bandwidth allocation algorithms described in the previous section, the Data Service requires the collection of data on actual network traffic. For this purpose, every transmitted packet will automatically get the Packetizer to increment a related counter, so that the Local Agent can periodically collect the values of this counter and make them available to the RUM data collecting routines. Other counters are implemented and managed for statistics, debugging and billing purposes in both bit-slice processors.

6.2 Low level operations supporting BA management activities

In order to perform the dynamic bandwidth allocation algorithms described in the previous section, the Data Service requires the collection of data on actual network traffic. For this purpose, every transmitted packet will automatically get the Packetizer to increment a related counter, so that the Local Agent can periodically collect the values of this counter and make them available to the RUM data collecting routines. Other counters are implemented and managed for statistics, debugging and billing purposes in both bit-slice processors.

7.0 Conclusions

In this paper we have presented the design of an experimental Data Service implemented on the METROCORE service-oriented MAN prototype to explore internetworking services that could offered to customers network man-