Abstract

In this paper we show that an ATM network equipped with Connectionless Servers (which are interconnected by a CLS overlay network) can adequately support SMDS multicasting as well as other key SMDS features. We argue that multicast traffic is most efficiently transmitted on multicast trees, and propose a scheme for the computation and maintenance of such trees exploiting the underlying routing procedure. We then review a number of CLS network design options and variables, namely, encapsulation, on-the-fly transmission, packet dropping and virtual topology. We evaluate the impact of these choices on the multicast service.

Keywords: ATM; SMDS; Multicast service; Connectionless server

1. Introduction

In today's data market, more and more customers are requiring Local Area Networks (LANs) with large bandwidth capabilities. There is also a growing need to interconnect LANs across wider geographical areas. The Switched Multimegabit Data Service (SMDS) has been designed to satisfy this need and to offer a high-speed connectionless public packet-switched data service.

According to the specification activity ongoing in US and Europe, SMDS can be provided either through the interconnection of regional operators by means of carrier networks as in US [17] or through a net of directly interconnected public telecommunication operator networks as in Europe [18].

The first SMDS implementations were based on DQDB (IEEE 802.6). Recent studies have addressed implementation over ATM [3].

The main challenge of SMDS on top of ATM stems from the fact that ATM is a connection-oriented network, while SMDS is a connectionless service. Particularly critical is the support, within ATM, of the SMDS multicast service, which enables end users to send a single data unit to multiple destinations without explicitly identifying the individual address of each recipient. This is needed in several applications commonly found in internet environments, such as: discovery of routes in remote bridging applications; routing information exchange in support of OSI protocols.
connectionless routing; file server discovery; and mapping of Internet addresses to E.164 addresses using a similar procedure performed, in a LAN environment, by the Address Resolution Protocol (ARP) to discover physical addresses.

The main goal of this paper is to address the multicast service for SMDS in an ATM network. Connectionless support is also extensively discussed, since it is essential for multicasting.

Several topics related to multicasting are addressed. First, a scheme for multicast tree computation is presented. Then, several CLS overlay network design variables and options (encapsulation, on-the-fly forwarding, packet dropping and virtual topology) are reviewed. The impact of these choices on the performance of the SMDS multicast service is evaluated.

2. ATM architecture supporting SMDS

The traditional SMDS service, based on DQDB (IEEE 802.6) standards, is implemented by a three layered architecture. This architecture specifies the Subscriber Network Interface based on DQDB (SNI₁), through which the CPE₁ accesses the network, the DQDB Inter-Switching System Interface (ISSI₁) and the DQDB Inter-Carrier Interface (ICI₁) between network nodes [15,17]. As SMDS provides interconnection between different networks, it must be supported by (and made to interwork with) different technologies as shown in Fig. 1.

In order to maintain SMDS customer investments, the recently proposed implementation of SMDS over ATM should guarantee full compatibility of the service with the DQDB based implementation, as required in Ref. [16]. This is achieved by maintaining the same level 3 protocol functionality and formats. SMDS is offered to the users either through DQDB or ATM based interfaces. In the case of ATM, the Broadband Subscriber Network Interface (SNI₂) corresponds to the User Network Interface (UNI); moreover, the Broadband Inter-Switching System Interface (ISSI₂) and the Broadband Inter-Carrier Interface (ICI₂) correspond to the Network Node Interface (NNI) [14].

![Fig. 1. Reference architecture.](image-url)
InterWorking Units (IWUs) are needed to interface ATM and DQDB environments to each other. These devices operate on a cell basis, up to the SAR sub-layer (handling MID and VPI/VCI allocation) in order to support the transfers from DQDB to ATM and vice versa. The SMDS layer 3 functionalities, including routing and management operations, are provided in the B-ISDN environment by the ConnectionLess Service layer (CLS-layer).

We have selected the use of ConnectionLess Servers (CLSs) as this is the most promising way to support SMDS over the ATM network. Connectionless Servers interconnected by dedicated VPs form a connectionless overlay network. The reasons that motivated this choice are extensively discussed in Ref. [3].

The architecture, which we propose to examine, contains the Broadband SMDS Switching Systems (SSSB) as network nodes providing SMDS over an ATM network. The SSSB is realised with a ConnectionLess Server (CLS) associated with one ATM node. Both ATM node and CLS are needed to build a SSSB. However, it is not required that every ATM node have a CLS.

Note that the link between two SSSB may consist of an ATM Virtual Path (VP) traversing several ATM nodes, which are not involved in the SMDS service and just provide for ATM connectivity.

In our proposed architecture the CLS and the ATM node are connected via an ATM interface. This makes it easy to upgrade a single ATM node to a full SSSB. Moreover, it makes for a smooth, easy and cost effective growth of SMDS over the B-ISDN network.

The linking of several SSSB over the ATM network can be viewed as a “CLS overlay network” on top of the ATM transport network. The CLS overlay network design issues are presented in the next section.

3. CLS overlay network

In the connectionless overlay network, ConnectionLess Servers are installed at various ATM nodes. The CLSs are interconnected with each other by a “virtual overlay network” consisting of several permanent VPs with pre-allocated bandwidth [20]. The overlay network does not necessarily include a VP for each CLS pair (i.e. full mesh).

In the design of the CLS overlay network, several parameters/functions must be selected. Among them we include:

(a) **Number and location of CLS facilities.** These should be chosen to match connectionless traffic volume and pattern. In general, the fewer the CLS nodes, the higher the load on each CLS, and the higher the load on each VP in the overlay network. Higher CLS load is generally not desirable, since CLS throughput capacity is well inferior to that of an ATM switch, thus making the CLS a potential bottleneck. On the other hand, the concentration of several different traffic sources on a few CLS overlay links makes the bandwidth allocation more efficient.

(b) **Bandwidth allocation.** This function adjusts the bandwidth of overlay network VPs to the connectionless traffic changing demands, making the best possible use of the residual bandwidth available on the ATM trunks.

(c) **Overlay topology connectivity.** The topology can range from fully connected (one VP between each CLS pair) to sparsely connected (a CLS maintains VPs only to a very few neighbouring CLSs). The connectivity plays an important role on CLS strategy performance. The basic trade-offs for low and high connectivity are:

(i) low connectivity generally leads to higher link traffic concentration and thus to more stable and predictable traffic. On the negative side, the datagram must travel through several CLSs, thus causing higher CLS loads.

(ii) high connectivity makes congestion easier to manage especially if most traffic travels just one hop. On the other hand, bandwidth is not efficiently utilised, because of fragmentation and traffic burstiness.

The main advantages of the CLS strategy are: ability to scale up to large network sizes; good
traffic concentration on overlay network VPs; dynamic routing capabilities; and centralised management of the bandwidth allocated to the connectionless service.

On the negative side, the potential drawbacks of the CLS strategy are: cost of the CLS facilities; liability to CLS congestion; and mis-sequencing of datagrams if dynamic routing is used.

From the above considerations, it appears that the CLS strategy is well suited to handle large user populations, exchanging relatively short bursts, without strict QOS requirements.

The CLS strategy runs into problems if users are transmitting sustained bursts at high rate and with specific requirements on cell loss, delay and sequencing. To overcome this problem, a hybrid solution which uses the CLS network for short bursts and end-to-end semipermanent VPs for stream type traffic can be sought [3]. Note that, when using end-to-end semipermanent VPs (which involves only the ATM layer and below), the SMDS service cannot be offered since it must be supported by upper layers. However, some of the SMDS features (address validation, address screening function and QOS selection), described in the next section, are already implicitly provided when an end-to-end connection is established. The hybrid solution may be very attractive for multimedia applications where the real time traffic is served by end-to-end VPs and the data traffic by the CLSs network.

4. ConnectionLess Servers

The ConnectionLess Servers (CLSs) are in charge of storing, routing and forwarding datagrams across the ATM network. They support the ConnectionLess Service layer (CLS-layer) functions as defined in Ref. [14].

Basically the CLS-layer provides the connectionless service on top of an ATM network. In fact, it complements the ATM Adaptation Layer (AAL) connectionless data transfer with additional user plane functionality. Namely, it provides for routing and addressing of connectionless, variable length packets, transferred from one source to one or more destinations without the need of establishing a connection on a per packet basis. The CLS-layer must also support the selection of QOS parameters for PDUs loss probability and transit delay, and the selection of higher layer protocols. In order to make the service more suitable to the needs of LANs, other functionalities are supported by the CLS, namely: source address validation, source and destination address screening, access class enforcement and multicast [4,14].

Source address validation insures that the sender of a connectionless data unit will not use an illegal source address. The source address of every data unit is therefore validated.

Address screening enforces restrictions on the transmission of connectionless data units to particular destinations, and the delivery of connectionless data units generated by particular sources. This feature makes it possible to create a Closed User Group. In this case, each user is allowed to send and receive packets only within a selected list of users.

Multicast is the ability of a customer to send the same data unit to several intended recipients. This means that inside the network there is the ability to replicate the multicast data unit up to the number of the desired destinations. The destinations are indicated in the multicast data unit by the use of a group address which the network has the ability to resolve. A Closed User Group can exploit the multicast ability to improve the communication efficiency among its members.

The routing of each datagram is performed on the basis of the E.164 destination address, which is mapped into the various VP Identifiers along the overlay network path. The user plane of the connectionless virtual overlay network interacts with the management plane in order to cope with resource allocation, dynamic routing and congestion [3].

In a CLS, congestion may occur when the traffic directed to a specific outgoing VP exceeds the VP peak rate allocation. In this case, incoming cells must be dropped. Preferably, the congested CLS will drop cells from the same datagram, rather than dropping cells randomly from different datagrams in transit.

Multiple paths between CLSs can be estab-
lished to provide load splitting capability. In order to guarantee the frame sequencing required by SMDS the load splitting is performed based on the destination address as suggested in Ref. [15]. More effective load balancing procedure which dynamically takes into account the network load condition could be used. However, such methods do not guarantee the frames sequencing, which, in this case, should be recovered by end-to-end protocols operating outside the public network.

5. SMDS multicast service requirements

An important service required by SMDS users is the multicast service, which allows a user, through the transmission of a single data unit, to reach a predefined set of destinations. Such a service is provided in each network domain, supporting SMDS, through the Group Address Agent network. Each Group Address is administered by a Group Address Agent (GAA), unique among all SMDS networks. The user requiring multicast service will forward his data unit to the corresponding Agent. Note that each customer of SMDS can address this Group, even if he does not belong to it.

The Agent tasks are:
(a) maintenance of the list of users belonging to the group;
(b) forwarding the multicast frame to all group members;
(c) preservation of the Group Address identification in the delivered data units.

In our view, the ConnectionLess Server overlay network is ideally suited to act as the Group Address Agent network. The use of a single CLS as replication point may cause the transmission on the VPs, interconnecting the CLSs, of several replicated data units, all with the same payload but different destination addresses. This unnecessary overhead can be avoided by organising the data unit transmissions according to a tree, where the root and nodes correspond to the CLSs. The leaves of the tree represent the members of the Group or the interface point towards other SMDS domain. The root CLS in this case must only generate a number of data units equal to the number of its outgoing branches in the multicast tree. Similarly the CLSs down the path duplicate the data unit until all leaves are reached. If this procedure is exploited in full, no duplicate data units will be transmitted on the same link. This approach, of course, requires additional procedures for multicast tree maintenance and user group lists updating in the CLSs.

In principle, multicasting of SMDS traffic across the ATM network could also be supported through the ATM multicast facilities already existing in most of the proposed ATM switches [19]. In practice, this solution is improper, for the reason that ATM requires individual bandwidth allocation for each multicast tree. Since a large SMDS system across a nation-wide network could conceivably have thousands of user groups and therefore thousands of multicast trees, it is obvious that the a priori dedication of bandwidth would be extremely costly. Besides, bandwidth requirements of multicast users tend to be bursty, thus are difficult to predict.

Furthermore, the protocols needed to provide multicast are too complex to be implemented in the ATM switching fabric. For instance, the modifications of the cells of the multicast frame require a processing hardly implementable in an ATM copy network. In fact, such modifications involve not only the ATM layer, but also both AAL 3/4 and the Connectionless layers.

On the contrary, the CLS multicast solution has the capability to perform all involved layer modifications and turns out to be quite efficient in terms of bandwidth utilisation.

6. SMDS multicast implementation in the CLS overlay

The next problem that must be addressed is the development of a multicast scheme, which can be efficiently implemented on top of the CLS overlay network described in Section 3.

Many techniques have been proposed for multicasting in a generic packet switched network. The typical performance measure (to minimise) was the network load caused by the replicas of
the original packet. Thus, most solutions propose to compute Steiner trees of minimum total length which span the multicast destinations [8,9]. The source issues only one copy of the message. Each internal node in the tree broadcasts the message on all outgoing branches.

Generally, the existing solutions assume that the topology is static (except for failures) and that the number of multicast groups is rather limited. In the SMDS over ATM case, the problem is quite different. The overlay topology can vary in time, based on the fluctuations of the SMDS connectionless traffic as well as of the connection oriented (ATM based) traffic. VPs between CLS nodes can be established, expanded, contracted and released quite dynamically. Furthermore, in a nation-wide SMDS configuration, thousands of multicast groups may need to be supported.

Because of the above unique characteristics of the SMDS/ATM environment, the existing schemes cannot be efficiently applied. A new scheme must be developed, which satisfies the following requirements:
- bandwidth efficiency (i.e., minimum number of transmissions);
- low processing overhead (i.e., simple multicast tree computation);
- robustness (to changes in topology);
- low multicast table overhead.

In the next section, we propose a solution to this problem.

7. Multicast tree algorithm

The challenging aspect of multicast is the computation of the multicast routing tree. Thus it is natural that, in our proposal, we will try to take advantage of the existing routing algorithm, in order to reduce the computational overhead. Two schemes, however, have prevailed in recent years: the Distance Vector scheme and the Link State scheme [13]. A discussion of these options is beyond the scope of this paper. We must point out, however, that the Distance Vector scheme would exhibit poor performance (possibly looping) in a dynamically changing CLS topology. The Link State scheme, on the other hand, is able to track more rapidly the change in topology. That is, each CLS node immediately broadcasts to all other CLS nodes any significant change in connectivity. This permits each node to maintain an up-to-date picture of the current topology and bandwidth allocation.

Based on these considerations, we will assume that CLS routing is of Link State type and that routes are chosen so as to minimise path length. We also assume that each CLS node has the global picture of the topology, and therefore can compute shortest routes.

The proposed multicast routing scheme is built on top of the CLS routing scheme. The basic idea is that the source sends the multicast packets to a designated CLS node. This node will then distribute the packet to the group members via a multicast tree. The internal nodes of this tree are called SubAgents (SAs), and have the task of replicating the packet on the outgoing branches. The computation of the multicast tree relies on the existing CLS routing solution, as we shall see later.

The decision to designate a particular CLS node as the root of the multicast tree, was inspired by the approach recommended by the SMDS specifications for the handling of Group Addressing [15]. Another possible implementation of multicasting could have chosen the entry CLS (i.e., the CLS connected to the source) as the root of the tree: this reduces the number of packet transmission. However, it dramatically increases the processing and memory overhead.

Referring to Fig. 2, we now illustrate the proposed multicast procedure.

(i) Source a sends a packet with destination address GA (Group Address) to CLS node V.

(ii) V recognises GA as a group address. It encapsulates the original packet in a new envelope, with source and destination address \((a, GA)\). It then forwards the packet to the CLS designated as root of the multicast tree.

(iii) The CLS receives the packet and resolves the GA using the table GA: \(g, G1, G2\). Namely, GAA issues three packets on its outgoing branches each with source and des-
Fig. 2. Multicast tree example. User data unit Payload Length.

Root

Connectionless Server (CLS)

P

T

G1

Q

G2

R

S

Fig. 2. Multicast tree example. User data unit Payload Length.

destination addresses \(<a, G1>\) to \(G1\), \(<a, G2>\) to \(G2\) and \(<a, GA>\) to \(g\). Note that \(G1\) and \(G2\) are the SubAgents of the multicast tree in CLSs \(P\) and \(Q\).

(iv) Subagent \(P\) resolves address \(G1\) using the table \(G1\): \(a, b, c\) and issues two packets each with source and destination addresses: \(<a, GA>\) to \(b\) and \(<a, GA>\) to \(c\). Note that the copy is not returned to the source (in this case, \(a\)).

(v) Only the frame originated by the source \(a\), and not the envelope, is delivered to every destination: \(b, c, d, e, f, g\).

The computation of the multicast tree basically reduces to the selection of the SAs, at the various levels and can be carried out recursively, as specified by the following algorithm.

**Multicast Tree algorithm:**

1. The GAA computes shortest paths to all members in the group (this is actually done as part of CLS routing). The superposition of these shortest paths is the multicast tree (see Fig. 2).
2. The intermediate children in the multicast tree are elected to be SubAgents if they have two or more children and more than \(K\) descendants. For the example in Fig 2, \(K\) was chosen to be 3. Thus, \(P\) and \(Q\) qualify, but \(T\) (only one child) and \(S\) (only 2 descendants) do not. In general, \(K\) is determined by the trade-off between table overhead and packet transmission overhead.
3. GAA dispatches the sublists to the Sub-Agents, e.g., GAA sends the sublist \(<a, b, c\) to \(P\).

Minor topology modifications will not strongly affect the structure of the multicast tree. They will be simply reflected in changes in routes between the nodes in the tree. For instance, in Fig. 2, if the \((\text{Root, } T)\) link is removed, the path from Root to \(G1\) will change from \(<\text{Root, } T, G1>\) to \(<\text{Root, } U, T, G1>\). In general, any modification of the topology may require a recalculation of the multicast tree. Note that even after a drastic change in topology (and underlying CLS routing pattern) looping cannot occur due to the tree structure of multicast routing.

In the proposed approach, we have heavily relied on the CLS routing scheme, to build the multicast tree. It should be noted that no mention was made of the ATM routing scheme, that is, the scheme that permits to find efficient routes for VCs and VPs. In fact, both CLS routing and SMDS multicast routing are totally independent of ATM routing. This property is very important, since it allows us to extend the multicast procedure to SMDS switching systems which include not only ATM, but also other subsystems (e.g. DQDB).

8. Impact of multicasting on CLS design options

The design of the CLS overlay network presents several options in the choice of topology, traffic control and protocol handling. The implementation of multicasting in the CLS network may influence some of these options.

In this section we examine several such options, and discuss implementation trade-offs.

8.1. Encapsulation

The transfer of a connectionless frame through the CLS overlay network in a broadband environment may be achieved by two methods: encapsulation and non-encapsulation.

The encapsulation method implies that the user data unit, when addressed to a remote SSS, is encapsulated as the payload of a new (encapsulating) data unit, whose header contains
proper addressing information to reach the destination SSS. A single Virtual Path between a pair of Sources is sufficient to carry all the SMDS traffic even if it pertains to multiple users. A decapsulation process is performed at the destination SSS to deliver the user data unit to the destination user.

If the non-encapsulation method is used, the SSS forwards the received data unit without modifications in the destination address field if addressed to a single user. Therefore, in the case of point-to-point transmission the non-encapsulation method performs better than the encapsulation method because no extra overhead in processing and transmission is introduced.

A very different behaviour, however, is observed in the case of multicast service.

If encapsulation is used, the GAA creates a new data unit, containing the multicast frame in its payload, for each user in the Group (end user or SubAgent). At the destination, the data unit is delivered unaltered, with the Group Address preserved.

The encapsulation method allows for additional information in the header of the encapsulating data unit. This may be used to improve network efficiency. For instance, a data unit containing multicast information may be given higher priority so as to reduce frame loss probability. This feature is particularly significant if the multicast transmission requires an acknowledgement from all destinations or the SubAgent technique is used. In the latter case, the loss of the data unit addressed to a SubAgent is very costly because it prevents the reception by its subset of users.

In the encapsulation scheme, the (encapsulated) data unit still carries the QOS information chosen by the source (to be delivered intact to the final destination), while the (encapsulation) header carries QOS and priority chosen by intermediate CLSs.

A further advantage of encapsulation is to maintain the data unit Group Address information. The knowledge of Group Address is relevant for further screening of the multicast frame at the destination. For example, in certain conditions the destination may be willing to accept frames from source $a$, say, under one Group Address but not under another Group Address. Beside the above mentioned extra features, en-

![User data unit Payload Length](image)

**Fig. 3. Protocol efficiency.**
capsulation allows more flexibility to implement new options, which may be defined in future SMDS versions to meet the needs of new generations of private connectionless networks.

The non-encapsulation scheme does not allow to provide all the above features, as no header manipulation is possible without loosing its original information.

The drawback of the encapsulation method is represented by the increased protocol overhead. Its effect depends on the traffic characteristics, i.e. on the statistical distribution of the payload length at the CLS-layer. When the encapsulation is performed, an overhead equivalent to two ATM cells is added to the data unit at the ATM layer. The BOM segment of the original data unit becomes the first COM of the encapsulating data unit while the BOM payload is filled with the header of the encapsulating data unit. In the same way, the original EOM segment becomes the last cell but one of the encapsulating data unit and a new EOM is produced containing trailer information (4 bytes) and a 40 bytes fixed pad. In this way we obtain a full compatibility between the operations performed in the ATM and in the DQDB environment.

To evaluate and compare the overall efficiency of SMDS over ATM with encapsulation and non-encapsulation we define the efficiency \( E \), as the ratio between the length of the payload of the SMDS user data unit and the length of the relative ATM cells in bytes [10].

The efficiency of the system, as function of the user data length at level 3, is shown in Fig. 3. The jumps in the curve are due to the discontinuities of the PAD length corresponding to changes of the payload length. When the payload increases, the efficiency reaches the asymptotic value \( E = 0.83 \), due to the SAR sub-layer and ATM layer overhead.

The loss in efficiency due to encapsulation is significant up to a data unit length of a few hundreds bytes. For longer payloads the difference becomes negligible.

The results plotted in Fig. 3 may be used to compare the overall performance obtained when dealing with traffic mixes as encountered in an Ethernet (IEEE 802.3) environment.

### Table 1 Overall protocol efficiency

<table>
<thead>
<tr>
<th>Model</th>
<th>Encapsulation</th>
<th>Non-encapsulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stuttgart University [21]</td>
<td>0.62</td>
<td>0.65</td>
</tr>
<tr>
<td>Emden-VolksWagen [21]</td>
<td>0.41</td>
<td>0.5</td>
</tr>
<tr>
<td>University environment [7]</td>
<td>0.49</td>
<td>0.5</td>
</tr>
<tr>
<td>Neg-expon distribution</td>
<td>0.69</td>
<td>0.77</td>
</tr>
</tbody>
</table>

As an example, we have considered four data unit length statistical distributions. The first was measured at Stuttgart University in an environment where both interactive and file transfer services were supported [21]. The second was measured at Emden VolksWagen offices where only very short data units were transmitted [21]. The third, presented in Ref. [7], was measured in an other university environment. The fourth considers data units with negative-exponential length distribution with an average of 1300 bytes.

The overall protocol efficiency has been evaluated for the four data unit length distributions. The results for encapsulation and non-encapsulation are compared in Table 1. The overhead needed to implement encapsulation reduces the overall protocol efficiency of about 10%. Note that the amount of this degradation depends on the traffic characteristics.

### 8.2. Reassembly vs on-the-fly

Two alternative forwarding schemes can be implemented in a CLS, namely: reassembly and on-the-fly. In this section, we briefly describe the alternatives and comment on their efficiency in handling multicast traffic.

Before the variable length connectionless data units enter the ATM network, they get segmented into ATM cells. Thus, the data units arrive at the CLSs interleaved with each other. Note, however, that sequencing of ATM cells belonging to the same data unit is maintained, as they are transmitted on the same VP.

The two alternative schemes to handle the data unit at the CLS are:

(a) Reassembly: the first and all subsequent cells belonging to the same data unit are stored
until the last cell comes in. Then the reassembled data unit is processed, routed and, if there is room in the CLS output buffer, delivered for segmentation and transmission on the proper outgoing VP.

(b) On-the-fly: the first cell is processed and routed to output if there is room in the output buffer of the proper outgoing VP. At the same time, management tables are updated so that the CLS is able to route all subsequent cells belonging to the same data unit to the proper VP. If during the transit of a data unit a cell is dropped because of congestion, all the subsequent cells of the same data unit are also dropped.

An extensive evaluation and comparison of the two schemes for single destination traffic is reported in Ref. [2]. In the following we summarise these results referring to the model shown in Fig. 4. Input data units have a mean length of 50 cells and arrive from 3 input VPs, each with a peak rate of 50 Mbps. On the output side, 5 VPs, each with peak rate 30 Mbps, exit from the CLS. The buffer pool is statistically shared among all output queues.

Fig. 5 shows frame loss versus output buffer size $B$ (cells) for different mean input rate, over each VP. The dashed and solid curves show the reassembly and the on-the-fly CLS performance, respectively.

The on-the-fly alternative presents a remarkable reduction in data unit loss for low input loads. This is due to the fact that this strategy makes more effective use of buffers since it does not require storage of the whole data unit. The relative advantage, however, decreases (and may even disappear) as the load increases. In fact, in this condition, see the 40 Mbps case, the buffer is fairly saturated and the on-the-fly CLS discards more data units as it operates practically only tail dropping (i.e., the front of the data unit is accepted, but the tail is rejected), while the reassembly always either accepts or discards the whole data unit.

Note that tail dropping invalidates data units already partially forwarded, thus causing a

---

Fig. 4. On-the-fly CLS model.

Fig. 5. CLS data unit loss versus buffer size for different mean VP input rates.
wastage of transmission capacity in all subsequent CLSs up to destination.

To overcome the tail dropping inefficiency, which has been observed in our experiments to be very heavy at high loads, a Discard algorithm is proposed, that reduces the probability of tail dropping by the on-the-fly scheme.

The basic operation of the Discard algorithm, which applies only on BOM cells, is as follows.

When a BOM is received, the actual number of cells, \( n \), in the shared output buffer is measured. If \( n \) is below a given threshold \( S \), the BOM is accepted. Otherwise, the buffer is approaching a congestion situation, and a check is performed to verify whether enough buffer space is available for the incoming data unit. If buffer space is available, the BOM is accepted, otherwise, it is dropped.

Note that no reservation is made for the buffer, but only a check is made. The information needed for the check is available in the BOM payload. This algorithm has no relevant impact on performance in optimal conditions, but it guarantees that the overall throughput does not decrease, when congestion and overload occur. In fact, without the algorithm, a large number of data units would experience tail dropping, severely degrading network performance. The purpose of the algorithm is to concentrate loss to a relatively small number of data units. This is achieved by forcing the loss of full data unit rather than the loss of several tails.

The comparison of loss rate in the on-the-fly CLS with and without the data unit Discard algorithm is presented in Fig. 6; traffic characteristics are as in the previous experiment and a heavy input load of 40 Mbps is considered while using threshold \( S = 0.85 \cdot B \). The results show that the Discard algorithm is very effective in reducing the fraction of tail dropped data units, which, otherwise, would approach one. Note that the Discard algorithm not only noticeably reduces the tail dropping but also makes the on-the-fly performance consistently better than the reassembly performance even for high load conditions.

We now want to verify whether the on-the-fly outperforms reassembly also in the multicast case. In our model we assume that in the CLS the multicast data units are dispatched one to each of \( N \) separate output VPs (i.e. the multicast tree at this CLS node has \( N \) branches).

In the reassembly scheme, a multicasting data unit can be handled in several different ways, depending on the number of copies kept in memory.

1. A single copy of the data unit is assembled in memory, and is then scheduled for transmission sequentially on each one of the \( N \) output VPs (i.e. the transmission on the \( i + 1 \)st out-

![Fig. 6. CLS data unit loss versus buffer size with and without Discard algorithm.](image-url)
put VP is scheduled after the \(i\)th transmission is completed).

(2) Multiple copies of the same data unit are buffered (one for each output VP), and are dispatched in parallel.

(3) Multiple BOMs (one per output VP) are created, but only one copy of the main body of the data unit is kept in memory. When the BOM reaches the head of the queue, the output processor reads the data from the memory.

Solution (1) is not very attractive because, although it memorises only one copy of the data unit in the buffer, it holds this buffer for \(N\) consecutive transmissions. Solution (2) is the easiest to implement. However, it uses as much buffer space as solution (1). In fact, it holds \(N\) copies of the data unit for one transmission time. Solution (3) is the most economical, in terms of buffer occupancy, practically one copy of the data unit for one transmission time. It is, of course, the most complex to implement, since it requires that the output channels be enabled to read concurrently from the same buffer. However, if there is only one physical trunk (at 150 Mbps, say) connecting the CLS to the ATM switch, then, there is only one output processor serving all VPs. In this case, the concurrent read does not pose problems. One should also note that in solution (3) the buffer residence time in memory is dictated by the slowest VP in the multicast set. Apart from these considerations, one may conclude that when solution (3) is adopted, the CLS performance in multicasting (in terms of buffer occupancy, data loss and delay) using the reassembly option is comparable to that observed with point to point traffic.

Next, we want to evaluate the on-the-fly option. Here again, there are several possible solutions.

(i) Replication of the incoming cells for all the queues.

(ii) Keeping a single copy of each cell in memory.

In solution (i), the cells belonging to the multicast data unit are replicated \(N\) times as they come into the CLS. More precisely, each cell (starting with the BOM cell) is copied into each VP output queue, where it is then processed on-the-fly. This scheme is simple to implement. It is the obvious extension of the single cast on-the-fly scheme. The main drawback is the potential waste of memory in the case in which several VP output queues are backlogged. In this case, the CLS would be forced to store up to \(N\) separate copies of the same data unit! To overcome this problem, solution (ii) can be adopted. In this solution, only one copy of each cell is kept in memory, and is shared by all outputs. More precisely, multiple BOMs are created and are immediately dispatched to the respective output VPs. The output channels progressively read the shared cells as they are ready to transmit. A shared cell is released once it has been read by all outputs. This cell by cell release reduces memory occupancy, but requires more processing overhead than the reassembly solution (3). Note that with solution (ii) on-the-fly multicasting requires about the same amount of buffering as on-the-fly single casting (in effect, it may require slightly more since memory residence is dictated by the slowest output queue).

In comparing the above solutions, we conclude that even in a multicasting environment the on-the-fly solution outperforms the reassembly solution (in terms of buffer occupancy and, therefore, of cell loss and delay). On-the-fly, however, is potentially more processor consuming than reassembly. Thus, CLS processing considerations may ultimately determine which option to adopt.

8.3. Packet dropping

The CLS network is a pure connectionless network. Thus, there is no provision for hop-by-hop flow control, such as windowing or back pressure. One form of congestion prevention, which is available, is the judicious (dynamic) allocation of bandwidth to the overlay network VPs. However, this is not completely effective since the connectionless traffic is by its nature unpredictable and the bandwidth resources are not always available when needed. Another form of congestion prevention is represented by the SMDS Credit Manager [16]. Again, however, the Credit Manager control does not fully serve our purpose because it is applied to the aggregate...
traffic entering the CLS network from an SMDS port, irrespective of its destination. So, without knowing the traffic pattern a priori, there is no guarantee that a particular set of SMDS credit allocations will prevent congestion.

Inevitably, traffic fluctuations will lead to overload on some internal CLS links and thus to buffer overflow. As is common practice in datagram networks, the CLS network reacts by dropping data units. Care is taken to drop entire data units rather than randomly picked cells (as discussed in Section 8.2). Generally, data unit dropping has a beneficial effect on congestion, in that it forces the reduction of the end-to-end protocol (e.g. TCP) window, and thus the temporary reduction of input traffic.

Several packet dropping policies have been proposed for datagram networks [11,12]. The general objectives of such policies are fairness, efficient buffer utilisation and simplicity of implementation. A popular policy which achieves the above objectives and is suitable for the CLS overlay network is the following:

when the free buffer pool drops below a given threshold, drop first the data units queued to the most congested links.

This scheme prevents hogging of the buffer pool by a single queue.

Before discussing the effect of such dropping policy on multicasting, we review some relevant characteristics of the multicast traffic.

Typically, multicasting uses a connectionless transport layer (e.g. UDP instead of TCP) in order to avoid the complexities of establishing connections to multiple destinations and maintaining multiple windows and credits. Thus, acknowledgements, if required, are handled by the application layer. Multicast traffic also generally consists of short bursts (e.g. ARP messages, control information updates, database updates, etc.). Thus, it is less likely to cause permanent congestion than a file transfer. One exception would be the multicasting in multimedia connections (e.g. videoconferencing, imaging, etc.). However, multimedia users would be better served by the conventional ATM service rather than by SMDS.

Upon entering the CLS designated either as root of the multicast tree or SubAgent, the multicast data unit is replicated and dispatched to $N$ output queues. If the node is congested, some of the copies get through, some are dropped. If the application does not require reliable multicasting (i.e. retransmission of lost packets is not required), this solution is acceptable. Examples of non reliable multicasting are the periodic transmission of sensor data, or the periodic propagation of network information.

If, on the other hand, reliable multicasting is required (e.g. distributed data base updating), it is easy to see that the proposed drop policy does not work very well. In fact, failure to receive an ACK even from one single destination will force the application to retransmit the multicast packet again to all destinations. Thus, it would have been better to drop all copies of the data unit at the congested CLS node, in order to avoid unnecessary overhead. The application could of course retransmit individual (as opposed to multicast) packets to the destinations which did not return an ACK. This, however, will require higher processing overhead at the source (to schedule individual retransmissions). It may also involve higher overhead in the network, especially if a large fraction of the destinations was missed.

In view of the potential overhead problems created by the retransmission of multicast data, it is advisable to give high priority to the reliable multicast traffic. This high priority allocation to multicast traffic should not have a major impact on the rest of the SMDS traffic, especially if the multicast traffic is characterised by short bursts and represents only a small percentage of the total traffic.

8.4 Virtual topology selection

The multicast tree algorithm was designed to handle a very general mesh topology. However, in special situations it is possible to choose a CLS topology which can greatly simplify the multicast operations. For example, a star topology would render multicast routing trivial. Unfortunately, it would also lead to unacceptable overload of the star CLS.

A possible compromise between uniform CLS load distribution and simplification of controls
(including multicasting) is the virtual ring topology. The CLS overlay ring would operate in a way similar to a buffer insertion ring, with destination removal [5]. Multicast traffic handling would become straightforward. Congestion protection along the ring is guaranteed by the fact that once a packet is accepted on the ring, it is delivered to destination. Intermediate CLS nodes need to buffer at most a full size packet in transit on the ring. Packets may be dropped only at the ring entry points. Fairness can be maintained using a credit circulation technique of the type proposed for the Metaring [5].

The virtual ring topology is attractive in small ATM networks. In an arbitrary ATM network, however, the ring may not be the most efficient topology to carry all connectionless traffic. Indeed, a mesh topology generally offers more direct paths (fewer hops) between node pairs than a ring. A possible alternative is a mixed solution consisting of a mesh topology to carry generic connectionless traffic, and an embedded ring, for multicast traffic.

We want to point out that the support of the multicast service may influence the topology design, as well as other traffic control considerations (such as, for instance, flow control and fairness).

9. Conclusions

Multicasting is an important feature of SMDS. It is necessary, for example, for extending typical LAN services to an enterprise network consisting of several LANs interconnected by SMDS. Multicast implementation is straightforward if SMDS is based on a connectionless and broadcast technology (e.g., DQDB). However, it becomes a challenge when SMDS is based on ATM. In such cases, the multicast implementation is closely coupled with the problems of connectionless traffic support and connectionless routing.

In this paper, we have focused on SMDS over an ATM network. We have shown that an overlay network of ConnectionLess Servers can efficiently handle the connectionless traffic. Then we have built the SMDS multicast support on top of the CLS overlay network, exploiting the routing procedure already existing there. The proposed solution is based on a multicast tree with CLSs at the root and at the various levels of the tree acting as GAA network.

We have then examined several options that exist in the CLS overlay implementation in order to determine the impact that multicast may have on such implementations. Our findings are summarised below.

In defining the multicast implementation, we have found that encapsulation offers distinctive advantages over non-encapsulation, it lends itself to the support of new features and widely justifies the small loss in efficiency, which has been observed.

The two CLS alternatives have been compared: on-the-fly and reassembly. We have shown that, at a cost of an increase in cell processing overhead, the on-the-fly strategy leads to lower data unit loss than the reassembly strategy.

As for buffer overflow handling (i.e., packet dropping) we have shown that reliable multicast packets require special care, in order to avoid heavy overhead caused by retransmissions. We also have advocated the use of priorities.

Finally, we have shown that special topologies (e.g., virtual ring) can actually greatly simplify the multicasting operations and are in fact quite attractive for small geography. Mixed solution (which includes a virtual ring embedded in the ATM topology), may also prove to be overall quite cost effective.

Future research directions in this area will include the development of models to study different packet dropping strategies and the investigation of special classes of topologies which are suitable for multicast traffic.

References

P. Crocetti received the degree in Electronics Engineering from the Università di Favia, Italy, in 1990. His thesis was developed during a stage in Italtel Central Research Laboratories, on policing algorithms in an ATM environment. Since 1990 he has been with Italtel as a researcher on performance evaluation and broadband services. His main interests concern traffic control, traffic modeling and internetworking issues in broadband networks. He is a delegate in the European SMDS Interest Group.

L. Fratta received the doctorate in electrical engineering from the Politecnico di Milano, Milano, Italy, in 1966. From 1967 to 1970 he worked at the Laboratory of Electrical Communications, Politecnico di Milano. As a Research Assistant at the Department of Computer Science, University of California, Los Angeles, he participated in data network design under the ARP An project from 1970 to 1971. From November 1975 to September 1976 he was at the Computer Science Department of the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, working on modeling analysis and optimization techniques for teleprocessing systems. In 1979 he was a Visiting Associate Professor in the Department of Computer Science at the University of Hawaii. In the summer of 1981 he was at the Computer Science Department, IBM Research Center, San Jose, CA, working on local area networks. During the summers of 1983, 1989 and 1992 he was with the Research in Distributed Processing Group, Department of Computer Science, U.C.L.A., working on fiber optic local area networks. During the summer of 1984 he was with Bell Communication Research working on metropolitan area networks. At present he is a Full Professor at the Dipartimento di Elettronica of the Politecnico di Milano. His current research interests include computer communication networks, packet switching networks, multiple access systems, modeling and performance evaluation of communication systems, and local area networks.

Dr. Fratta is a member of the Association for Computing Machinery and the Italian Electrotechnical and Electronic Association.
M. Gerla was born in Milan, Italy. He received a graduate degree in engineering from the Politecnico di Milano, in 1966, and the M.S. and Ph.D. degrees in engineering from UCLA in 1970 and 1973, respectively. He joined the Faculty of the UCLA Computer Science Department in 1977. His research interests cover the performance evaluation, design and control of distributed computer communication systems and high speed computer networks (ATM and Optical Networks).

M.A. Marsiglia received a graduate degree in Computer Science from the Università degli Studi of Milan in 1990. In the same year, she joined the Italtel Central Research & Development Laboratories as a researcher on broadband communication protocols and services. Her main interests concern network architecture and protocols definition for the support of broadband connectionless services on DQDB and ATM technologies. She is a delegate in the European Telecommunications Standards Institute (ETSE) NAS group and in the European SMDS Interest Group (ESIG).