ISRN SICS-R--96/06--SE

An ATM Adaptation Layer
for Reliable Transfers

by

Gunnar Karlsson

An ATM Adaption Layer for Reliable Transfers

by

Gunnar Karlsson

July 1996

gk@sics.se

Swedish Institute of Computer Science
Box 1263, S-164 28 KISTA, SWEDEN

An ATM Adaptation Layer for Reliable Transfers

Gunnar Karlsson

Swedish Institute of Computer Science

Box 1263, S-164 28 Kista, Sweden, E-mail gk@sics.se

There has been much attention given to ATM Adaptation Layers (AAL) over the ten-year history of ATM standardization. Yet none of the existing standards include a retransmission protocol to ensure reliability of the transfers. In this paper we propose an AAL for doing selective retransmissions of individual segments that have been lost or corrupted in the transfer. Since the amount of retransmitted data equals the amount lost, there is little risk of aggravating congestion and the reliability is provided as economically as possible, which is a virtue for low-capacity radio links. The protocol relies on the orderly delivery of ATM virtual circuits so that data octets can be identified uniquely by their location in the data stream.

1. Introduction

The development of adaptation layers for ATM has been less than straight forward. From ITU's original set of four, only one version has been retained in practice – AAL 1 – and a fifth version has been added by the ATM Forum. AAL 1 serves for circuit emulation but little else, and AAL 5 is promoted by the Forum for all types of communication services. AAL 2, designated for variable-rate video, has not been defined. The fate that befell AAL 3 and 4 was that they were merged into version 3/4 and superseded by version 5, mainly due to high overhead and prickly implementation. Although AAL 3/4 is retained in ITU-T Recommendation I.363 it appears that few intend to use it.

Fig. 1 The proposed ATM protocol stack.

Despite the many proposals, none of AAL 1, 3/4 and 5 include the functionality to make the adaptation layer a full-fledged transport protocol. This means that a transport protocol such as TCP is needed above the AAL to ensure the reliability of the transfers. The architecture is consequently disfigured since protocols from different standardization communities are merged into a less than integral whole. The complications of fitting protocols together that have not

been co-designed is manifested in unpredictably low throughput and long response times [5].

The protocol stack that we consider for ATM is depicted in Fig. 1. (The layering should be considered conceptually and does not indicate partitions of an implementation.) The role of ATM adaptation is to ameliorate the behavior of the network for the communication applications. This paper suggests a limited set of functionality that would enable the adaptation layer to assume the role of a transport layer. We consider three levels of data integrity that the protocol should support:

- Maintain the framing structure of the application.
- Ensure that the amount of data delivered equals the amount sent.
- Guarantee the correctness of delivered data.

Of these three levels, only a) is offered by AAL 5. Level a) is needed when the application frames the information (as opposed to stream oriented protocols). For many applications, integrity at level b) may be sufficient: video and audio do not require absolute delivery of every octet. Undetected loss of data octets would, however, place encoders and decoders out of synchronization which for compression, scrambling or FEC is an irrecoverable state.

The handling of the loss and delay variations are for these information types best done at the application layer where both the signal's syntax and the application context are known. Synchronous service should not be expected from an asynchronous network; there are well designed and tested networking systems if that is desired. (Such TDM systems may however share resources with the ATM network at the SDH layer.)

Applications that are not designed for asynchronous transfers could be supported by a suitable intermediate protocol which performs clock synchronization and delay equalization (like RTP [7]). As ATM gets more widely used, more applications will be refitted to accept the asynchronous service at hand.

Many data applications do require all three levels of integrity to be ensured by the network protocols. Reliable transfer—meaning absolute delivery of every octet of data—can only be provided by means of retransmission. Forward-error correction is a useful method of raising the reliability but more errors can be introduced in the channel than what the code can correct. (The two techniques are good compliments, however [1].) The main contribution of this paper is a method for doing retransmission of individual segments at the adaptation layer in order to ensure reliable delivery of data.

The proposed adaptation protocol provides all levels of data integrity. We rely on the following assumptions for our design:

- The probability of bit errors will be much smaller than the probabilities of cell loss caused by burst errors and by multiplexing overload.
- Virtual circuit identifiers will not be a scarce resource and the tariffing will not penalize the use of multiple virtual circuits. This would mean that the fixed part of a connection charge does not depend on the number of parallel circuits that are established simultaneously.
- Flow control is not needed in the AAL: it is either avoided by the use of rate control for deterministic and statistical bit-rate service, or it is handled by the exchange of resource management cells for the available bit-rate service [6]. (We do not believe that an ABR service will eliminate cell loss, however.)

The proposal herein is akin to the service-specific connection-oriented protocol (SSCOP) designed for the signaling AAL [2]. Our suggested protocol, called AAL 6, is intended for selective repeat of lost octets of data while the SSCOP does retransmissions on a frame level. The segmentation and reassembly sub-layer of the protocol has been suggested for real-time transfers of audio and video [4].

The outline of the paper is as follows. Section 2 describes the segmentation and reassembly sub-layer of the AAL 6, Section 3 the convergence sub-layer, and Section 4 outlines the retransmission protocol.

2. Segmentation and reassembly

The responsibility of the SAR sub-layer is to segment convergence-sublayer frames into multiplexing units (cell pay-loads) at the sending side, and to reassemble them at the receiving side. ATM networks will lose cells in the transfer, either due to multiplexing overloads, bit-error in the cell header, or misrouting. The latter is caused by detected bit-errors in the VPI/VCI fields, or by incorrect entries in the switches' routing tables.

In order to reassemble a frame properly, the receiver needs to identify lost cells. These may be detected by aid of a sequence number on each segment, and by a length-field or a CRC on the full frame. A sequence number per cell does not suffice to determine the amount of lost data when cells are partially filled. Neither is a length-field per frame enough since it could be received in error. This means that level b) integrity cannot be offered when retransmission is not possible or wanted.

Our proposal is therefore to identify lost octets of data. Each cell or frame therefore contains a count of how many octets of data has been previously sent. The receiver compares that value with its own count of received octets. Any discrepancy indicates a loss of data with length equal to the sent minus received octet counts. Three fields of protocol information are needed for this.

(i). Length-field for partially filled cell.

(ii). Count of previously transferred octets.

(iii). CRC for detection of errors in these fields.

The segment-length field (SL) in (i) requires 6 bits. The octet-count in (ii) needs $\lceil \log_2 (N + 1) \times O_{pl} \rceil$ bits if N contiguous cell-losses should be detected, each with maximal pay-load O_{pl} octets. The field is structured as follows.

SC : a segment-count, giving the number of segments O_{pl} octets long.

OC : an octet-count, giving the number of octets in addition to the full segments.

The fields in cells i and $(i+1)$ are related by

$$SC_{i+1} = SC_i + \lfloor (OC_i + SL_i) / O_{pl} \rfloor \text{ and}$$

$$OC_{i+1} = OC_i + SL_i - \lfloor (OC_i + SL_i) / O_{pl} \rfloor \times O_{pl}$$

($\lfloor x \rfloor$ denotes the integer-part of x , and $\lceil x \rceil = \lfloor x \rfloor + 1$ when x is not integer). For example, if 723 octets have been sent and $O_{pl} = 47$, then $SC_i = 15$ and $OC_i = 18$. If $SL_i = 31$, then $SC_{i+1} = 16$ and $OC_{i+1} = 2$.

The *OC* needs 6 bits. The modulus of *SC* has to be chosen according to the bit-rate of the service compared to the link-rate and to the cell-loss probability of the transfer.

Fig. 2 Ambiguity caused by undetected bit-error.

All AAL-header fields have to be protected against bit and burst errors. Fig. 2 illustrates the ambiguity that may result from an undetected bit-error in the segment-count. Undetected errors in the fields may lead to deletions and insertions in the data-stream. A deletion is an undetected loss of data (the signal is shortened); insertion means that garbage octets are added to the stream and may happen when the segment-length has a higher value than the number of valid octets in the payload. It may be detected by the aggregate count given by *SC* and *OC* in the subsequent header.

To avoid replication of functions at higher layers, the probability should be very low of having deletions and insertions present which cannot be detected or changed into erasures. This happens when the number contiguous losses exceeds the modulus of the segment count and when the CRC fails to detect bit errors in the segment and octet counts, and in the segment length. There should thus be a balance in the lengths of the *SC* and the CRC so that the two probabilities of failure are on par.

The length-field and the octet-count are not always needed. Two header-formats are therefore used in the adaptation layer. The most significant bit of the SAR control information is used to indicate the format used for each cell. When the length-field is O_{pl} and the octet-count is zero, then the bit is reset and only the segment-count is sent, protected by a CRC. Otherwise it is set and all three fields are sent, protected by a longer CRC. The two AAL-headers are shown in Fig. 3.

Fig. 3 The AAL-header formats with recommended bit-allocations (two alternatives given).

MSB=0		SC	CRC
Rec.	3	4 [bits]	
Alt.	4	3	
	5	2	

MSB=1		SC	OC	SL	CRC
Rec.	3	6		6	8 [bits]
Alt.	4	6		6	7
	5	6		6	6

The maximum pay-load is consequently 47 or 45 octets, depending on the header needed in the cell. For the transport-stream of MPEG-2 which has a fixed framing structure of 188-octets long packets this means that they fit into four cells [3]. Both header types will typically appear in streams for variable bit-rate services which do not have fixed frames like this MPEG-stream. This is illustrated in Fig. 4.

Fig. 4 The use of the AAL-headers. (*RC* denotes the count of received octets and $RC = 47 \times SC + OC$ if no loss has appeared).

3. Convergence

The reassembly with the loss detection outlined above ensures that the application can assume that the amount of data received is the same as sent (losses may be replaced by dummy values). The role of the convergence sub-layer is then to ensure the integrity of the application frames that it transfers.

Fig. 5 Format for the convergence sub-layer frame encapsulation.

The structure used for the frame encapsulation is shown in Fig. 5. A six-octet trailer will be appended to the frame, no padding is needed, as for the AAL 5 framing, since the SAR sub-layer does not require fixed-length segments. The end of frame is indicated by the ATM-Layer-User to ATM-Layer-User (*AUU*) bit in the cell header (as for AAL 5). The UU and CPI fields are assumed on basis of their justification for inclusion in AAL 5. Note that the total CS frame is less than 65536 octets long in order to be specified by a 16-bit length field.

It would be possible to add a sequence number to each frame trailer in order to enable retransmission of a corrupted frame. Given that the maximum frame length is large, a single cell loss or bit error would cause a huge increase in the amount of data to transfer. The cost is more delay, if the bit rate is restricted (for wireless systems, for instance), or more load if

the retransmission is added to the regular flow. The latter may even lead to network instability since the increase in load may aggravate the congestion which caused the loss.

4. Retransmission

To get around the problems with frame-level retransmissions we have designed a retransmission scheme that can repeat the transfer of individual segments (*ie*, cell pay loads) at the level of the SAR sublayer. Segment-level retransmission is often regarded as too costly in terms of overhead since each segment is supposed to be identified by a unique and thereby long sequence number. Recognizing that ATM networks give an orderly delivery simplifies the protocol considerably: cell loss is identified by the octet-based sequence number at the SAR sub-layer, and unique identification is simply done by counting received and lost octets and requesting retransmissions of the latter. The first and the subsequent transfers of a particular segment cannot be told apart if mixed on to a single virtual circuit, however. The protocol requires therefore two virtual circuits in the forward direction and one in the backward direction. The SAR sub-layer above is used on all three circuits to detect loss.

Suppose a full frame has been received, the end of frame being indicated by the *AUU* bit in the cell header, and the frame is verified by the CRC in the CS trailer. When it fails, the only resort is to request a new transfer of it. We assume however that the probability of bit-error is much lower than the cell loss probability and thus the full-frame retransmissions would be rare enough not to risk network stability, as alluded to above. For wireless channels with higher bit-error probability it may be worthwhile to include a CRC per segment in order to limit the minimum amount of data that must be retransmitted because of a single bit error (the length of the CRC should be commensurate with the bit-error rate of the link).

The retransmission works as follows. Received segments are appended to the existing content of the reassembly buffer. Every octet—received as well as lost ones—is counted which means that each octet is uniquely defined by its position in the stream of octets. This requires only regular integer word-resolutions: for instance, a 32-bit counter wraps around only every 14 seconds for a 2.5 Gb/s connection which is longer than any terrestrial round-trip delay.

Detected loss leads to two tasks: the corresponding amount of octets is reserved in the reassembly buffer for the data's final delivery, and the block of missing

octets is marked in a retransmission table. The table thus contains a complete and ordered list of the data segments that are expected on the second forward-going circuit. The format of the list is shown in Fig. 6. Each table entry for a missing block contains the number of the first octet and the length in octets of the missing segment.

Fig. 6 Format for the retransmission table.

First octet [32b]	Length [16b]
137	4
36453	47
456329	2231
36	12

The list is sent to the sender's AAL when end-of-frame is received or when the list is as long as fits in a cell (seven entries of 6 octets each). The latter means that there can be multiple lists sent when the frame is long and the loss rate high. The requested octets are re-sent on the second virtual circuit. If a segment is the last of a frame, then the *AUU* bit is set in the cell carrying the segment (in the last cell, in case of multiple cells needed to carry the segment).

Fig. 7 An example of the retransmission protocol.

The receiver uses the retransmission table in order to determine where segments delivered on that circuit belong. An entry in the table is erased when the corresponding data octets have been delivered. The operation of the protocol is illustrated in Fig. 7: A segment sent on the first circuit (FVC1) is lost. A retransmission request, giving the start and length of the lost segment, is sent on the backward circuit (BVC). The segment is retransmitted on the second circuit (FVC2). This is the first retransmission of the connection, as can be seen from the number of previously transferred octets on BVC and FVC2.

Any loss on the second circuit is detected as on the first circuit and leads to a new retransmission request; the first entry in the table is erased, it has been answered (although not satisfactorily) and a new request for the segment is added at the end of the table. For instance, the third entry of the table in Fig. 6 could be for a full frame (length exceeds a single-cell segment) and the last entry for a failed retransmission (the first octet for the segment is lower than preceding entries).

There are complications that must be handled properly by the scheme: retransmission requests may be lost and the last segment of a transfer on either of the three circuits may be lost. The problems are solved by allowing either party to flush the circuits to avoid waiting for one another indefinitely. A flush is a segment with zero length ($SL=0$) and the AUU bit is set. It acknowledges the amount of octets previously sent on the circuit but does not add to the count of transferred or received octets and loss of a flush can therefore be safely disregarded.

Since loss of cell i is identified by the sequence number in cell $i+1$ (or higher), there is a problem for the last cell transferred on a virtual circuit. The problem is solved by sending a flush after the last cell. A flush on the first forward circuit results in either a retransmission request or a flush on the backward circuit. If the sender receives a flush on the backward circuit then it flushes the second forward circuit. Such a flush should be done routinely by the sender to confirm a completed transfer and before closing the connection. Note that the last flush on the second forward-going circuit may get lost without causing trouble: there are no retransmissions left anyhow. (The sender can actually flush the first circuit continuously until a flush or retransmission request is received on the backward channel.)

The second complication is caused by lost retransmission requests. When a request has been lost it will be detected once the following request arrives on the backward circuit and the sender will respond by sending a flush on the second forward-going circuit. It will also stop serving subsequent requests. When the flush is received, it first triggers a flush on the backward circuit to the sender and, second, all unresolved entries of the retransmission table are re-sent. The flush signals the sender to resume service of retransmission requests. (Retransmission of lost retransmission requests is thus handled by Go-back-N.)

5. Performance notes

The protocol does not use send-windows and has consequently none of the performance limitations associated with sequence number starvation. The retransmissions cover only the lost or corrupted segments in order to minimize the additional network load. The throughput of the SAR sublayer is obstructed when loss has occurred since the reassembly is halted while awaiting the retransmission. This may lead to overflow of reassembly buffers if the sender continues to send at a high rate. (The obstructing is, however, avoided if orderly delivery of application frames is not mandated.) There is thus a coupling between the retransmissions and the flow control to notify the sender so that loss can be avoided at the receiver without overly throttling the sending rate. The choice of flow-control protocol is independent of the retransmission protocol presented in this paper. We defer the throughput analysis until details of the flow-control implementations for ABR are available.

6. Summary and Conclusions

We have described an ATM adaptation layer for reliable transfers. The protocol allows detection of lost octets in a data stream and provides a retransmission mechanism for their replacement. In addition to the virtual circuit used for the data transfer, the protocol uses one virtual circuit from the receiver to the sender for ordering retransmissions and a second virtual circuit from the sender to carry the retransmitted segments of data.

A retransmission request consists of the number of the first lost octet and the total number of contiguously lost octets. Since ATM guarantees orderly delivery, counting octets provide a unique index to every octet in the data stream. Exception handling of lost retransmission requests and retransmissions is done by means of flushing the circuit with an empty cell.

The protocol could be suitable for ATM over wireless links for which the parsimony of the retransmissions is valuable. The performance and implementation of the protocol is for future study.

7. References

- [1] E. Ayanoglu, *et alii*, "Performance Improvement in Broadband Networks Using Forward Error Correction for Lost Packet Recovery", *Journal of High-Speed Networks*, Vol. 2, No. 3, 1993, pp. 287–303.
- [2] T. R. Henderson, "Design Principles and Performance Analysis of SSCOP a New ATM Adaptation Layer Protocol," *ACM Computer Communication Review*, Vol. 25, No. 2, April 1995, pp. 47–59.
- [3] ISO/IEC JTC 1/SC 29/WG 11, *CD 13818 Generic coding of moving pictures and associated audio information*, Part 1: Systems, Part 2: Video, Part 3: Audio, December 1993.
- [4] G. Karlsson, "ATM Adaptation for Video," in *Proceedings of Sixth International Workshop on Packet Video*, Portland, OR, September 26–27, 1994, pp. E3.1–5.
- [5] K. Moldeklev and P. Gunningberg, "How a Large ATM MTU Causes Deadlocks in TCP Data Transfers," *IEEE/ACM Transactions on Networking*, Vol. 3, No. 4, August 1995, pp. 409–422.
- [6] H. Ohsaki, *et alii*, "Rate-Based Congestion Control for ATM Networks," *ACM Computer Communication Review*, Vol. 25, No. 2, April 1995, pp. 60–72.
- [7] H. Schulzrinne *et alii* "RTP: A Transport Protocol for Real-Time Applications," Internet Request for comment no. 1889, January 1996.