

An application layer gateway for air traffic management communication by satellite

E. Kristiansen

European Space Agency, Keplerlaan 1, NL-2200 AG Noordwijk, The Netherlands

Simone Patella, Massimo Mazzocanti

Vitrociset, Via Tiburtina 1020, I-00156 Roma, Italy

Abstract

Aeronautical safety communication via satellite presents some interesting challenges, one of which is the problem of passing inelastic traffic (generated by real time events) over a limited bandwidth communication channel. Today's aviation standards prescribe the use of a reliable transport protocol, a proposition that is inherently incompatible with inelastic traffic. Though network capacity should be sized for the normal peak load, one must assume that unusually high peak loads will occasionally occur, leading to congestion. This paper describes the concept of an application layer gateway, capable of mitigating congestion by re-ordering and, if necessary, selectively discarding messages in accordance with set rules. A demonstration test bed is described. Test results, confirming the capability of the gateway to relieve congestion, are also given.

Acronyms

<i>AGW</i>	=	Application Layer Gateway
<i>AOC</i>	=	Airline Operational Communication
<i>ATM</i>	=	Air Traffic Management
<i>ATS</i>	=	Air Traffic Services
<i>CLNP</i>	=	Connectionless Network Protocol
<i>CLTP</i>	=	Connectionless Transport Protocol
<i>IP</i>	=	Internet Protocol
<i>OSI</i>	=	Open Systems Interconnection (ISO standardized protocol stack)
<i>PEP</i>	=	Performance Enhancing Proxy
<i>QoS</i>	=	Quality of Service
<i>RTT</i>	=	Round Trip Time
<i>SDLS</i>	=	Satellite Data Link System
<i>TCP</i>	=	Transmission Control Protocol
<i>TP4</i>	=	OSI Transport Protocol level 4
<i>UDP</i>	=	User Datagram Protocol

I. Introduction

Aeronautical safety communication encompasses two types of services:

- Air Traffic Services (ATS), including communication for air traffic control between a controller and the pilot.
- Airline Operational Communication (AOC) between the airline and the pilot.

ATS and AOC are collectively referred to as Air Traffic Management (ATM). ATM communication was traditionally by voice, using a network of ground based VHF stations, but an evolution towards replacing voice by data communication is ongoing, with the aim to phase out voice services for all but emergencies within the 2020 to 2030 time frame.

Satellite communication has had its place in ATM communication for more than a decade with the Inmarsat AMSS, the Japanese MTSAT and, more recently Iridium. But all of these are used exclusively in oceanic and

remote areas where no ground based infrastructure is present (with the exception of rather unreliable HF voice service).

In recent years, however, interest in using satellite for ATM also in continental airspace is mounting, driven mainly by two considerations:

- A shortage of VHF spectrum in certain parts of the world as a consequence of growing air traffic.
- The “dual link” concept of increasing availability and reliability of communication by having two completely separate communication systems available at all times and everywhere.

The European Space Agency has studied a satellite communication system specifically designed for ATM in the SDLS (Satellite Data Link System) project in past years, and is currently starting the Iris (Iris is not an acronym) project, aiming at the full design and development of such a system. The work presented in this paper was one element of the SDLS project.

ATM data communication via satellite exhibits a number of unusual characteristics that pose interesting challenges to the system designer. Infrequent, short messages; limited bandwidth in the satellite channel; channel shared by many users; stringent QoS requirements; the need for efficient use of resources, are just a few of the key issues that such a system has to cope with. Though voice services will be part of a future system, they are not addressed in this paper.

Chapter II discusses these problems in more detail. Chapter III discusses our proposed solution, the Application Layer Gateway (AGW), while chapter IV addresses other approaches to the problem. A test bed of the AGW is presented in chapter V, and chapter VI summarizes the main results from the test bed. Finally, some conclusions are drawn in chapter VII.

II. Problem description

A. ATM traffic profile

ATM data traffic consists predominantly of very short messages (message body between ~20 and a few hundred bytes), spaced by intervals of several seconds to several minutes. A few longer messages (up to a few kilobytes) are also present. Messages are triggered by events related to the progress of the flight or, for a few message classes, by the passing of time. In other words, the traffic is *inelastic*, consisting of mainly short, infrequent messages occurring at unpredictable, irregular intervals.

The QoS requirements are extremely strict: Permitted delays are in the order of a few seconds for the most urgent messages, up to maybe a minute for less urgent ones. It is noted that late delivery may actually be hazardous. Delivering outdated reports or instructions that are no longer consistent with the current flight situation may be dangerous, and must be avoided.

B. The ATN Protocol

The currently used communication protocol stack for ATM is ATN (Aeronautical Telecommunications Network), an OSI-based protocol standardized by ICAO in the 1990s. See Figure 1.

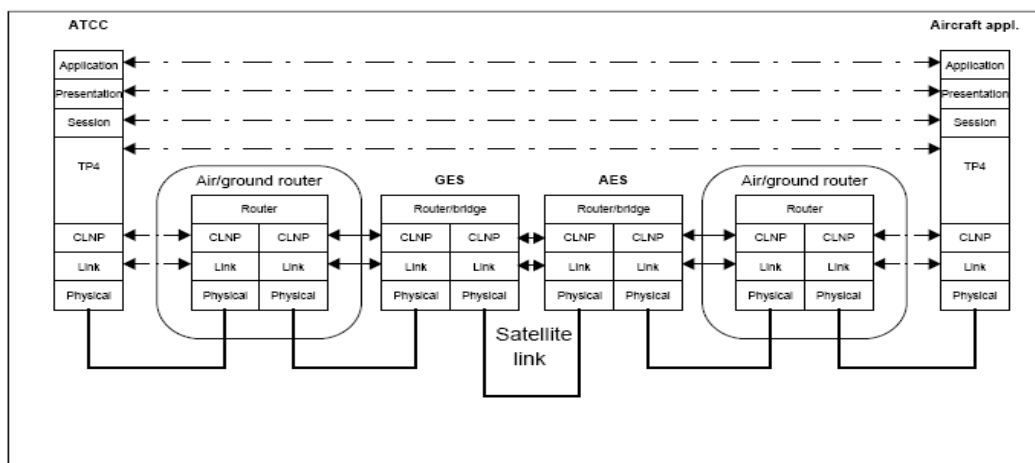


Figure 1: The ATN protocol stack

The network layer protocol is the connectionless OSI protocol CLNP (equivalent to IP in the TCP/IP model). Like IP, CLNP may lose or duplicate packets, and may deliver packets out of order.

The transport layer protocol is the reliable, connection oriented OSI TP4 protocol (equivalent to TCP). Like TCP, TP4 provides error free, in-sequence delivery of data, and TP4 provides flow and congestion control. The mechanisms used by TP4 to achieve this are quite similar to those of TCP. TP4 and TCP are end-to-end protocols, resident only in end systems.

The ATN protocol stack is designed for *elastic* applications, i.e. applications like transfer of large files that can adapt their rate of sending to the capabilities and loading of the network.

C. Congestion

A well dimensioned network for inelastic traffic will be sized for normally occurring peak traffic plus some margin to cope with the burstiness of the traffic. However, any excess capacity is costly, both in economical terms and in terms of radio spectrum occupancy. It is not feasible to size for the theoretically highest possible peak load of all aircraft wanting to communicate at exactly the same time.

The rate of incidence of congestion can be reduced by provisioning additional communication resources, but it cannot affordably be reduced to zero. Said differently, the traffic distribution has a very long, thin distribution tail, and it is not affordable to size for covering the full tail.

Therefore, one has to assume that congestion will occasionally occur. And it will occur when it is least wanted: When some abnormal air traffic situation creates a higher-than-normal communication traffic load.

Imagine, for example, a major airport being closed due to weather or an accident. All air traffic headed for that airport needs to be re-routed to other destinations where, in turn, the extra traffic calls for the creation of holding patterns. All this is pretty standard, but to accomplish it, a lot of additional communication traffic is generated.

D. Mismatch between application and protocol

As pointed out above, the ATM applications are *inelastic* while the network assumes *elastic* traffic. This is a fundamental mismatch that gives rise to serious problems when the network load approaches, or exceeds, the available capacity, even for short time periods.

There are several distinct problems:

- TP4 insists on delivering all data, and delivering it in sequence, meaning that queues will start building up, and all traffic will be delayed.
- If the congestion persists, delays will grow unbounded. When the delay exceeds a certain value, timeouts will expire, triggering unnecessary retransmissions, thus further loading the already congested links.
- TP4 congestion control works by marking packets with the “congestion encountered” bit, causing the receiver to reduce its receive window. This mechanism is not effective for the bursty traffic profile: Knowing that there was, or there wasn’t congestion one minute ago says nothing about the current conditions. And what would you do with that information, anyway?

A key point in this context is that congestion control resides in the transport layer of the end systems:

- The only entity that could relieve congestion is the sending application, by dropping low priority traffic in favor of higher priority.
- But data is queued in the transport or network layer, and, due to the layered architecture, there is no way the protocol stack can inform the application about congestion.
- The router in front of the bottleneck link only knows about the network layer, and there is nothing it can do to relieve congestion. If it drops packets (which it will have to do sooner or later, if congestion persists), they will be retransmitted by TP4.
- Result: Congestion collapse, with no obvious recovery (except breaking all TP4 connections).

III. The Application Layer Gateway

A. Rationale

There are basically two ways to deal with congestion in the presence of inelastic traffic:

- Delay less urgent traffic in favor of more urgent traffic. If congestion is mild and of short duration, it may still be possible to meet the QoS requirements of all traffic.
- Discard traffic. In case of heavy and/or long-lasting congestion this is the only solution, however unpalatable it may seem.

The network elements that know the current loading situation of the bottleneck link are the satellite terminals (ground Earth stations and airborne terminals), where data will be accumulating in the buffers until they overflow. Therefore, a network element introduced for the purpose of mitigating congestion should logically be placed at the satellite terminals.

As explained above, such a congestion mitigation element must operate above the reliable transport layer because

- The transport layer, being end-to-end reliable, cannot discard data
- Data discarded at lower layers will be retransmitted by the transport layer

This leads to the inevitable conclusion that an intelligent gateway at application layer is needed.

The Application Layer Gateway (AGW) is such a gateway.

B. The AGW concept

The AGW protocol model is shown in Figure 2.

Each application message has associated with it certain QoS (Quality of Service) requirements, either explicitly or implicitly by the type and possibly the context of the message. The most important QoS parameter is the remaining time-to-live, after which the message becomes obsolete. It is noted that, in the ATM context, the time-to-live is an inherent message property. Any message exceeding it may be hazardous: Imagine receiving a delayed clearance or a delayed position report that is no longer consistent with the current flight context.

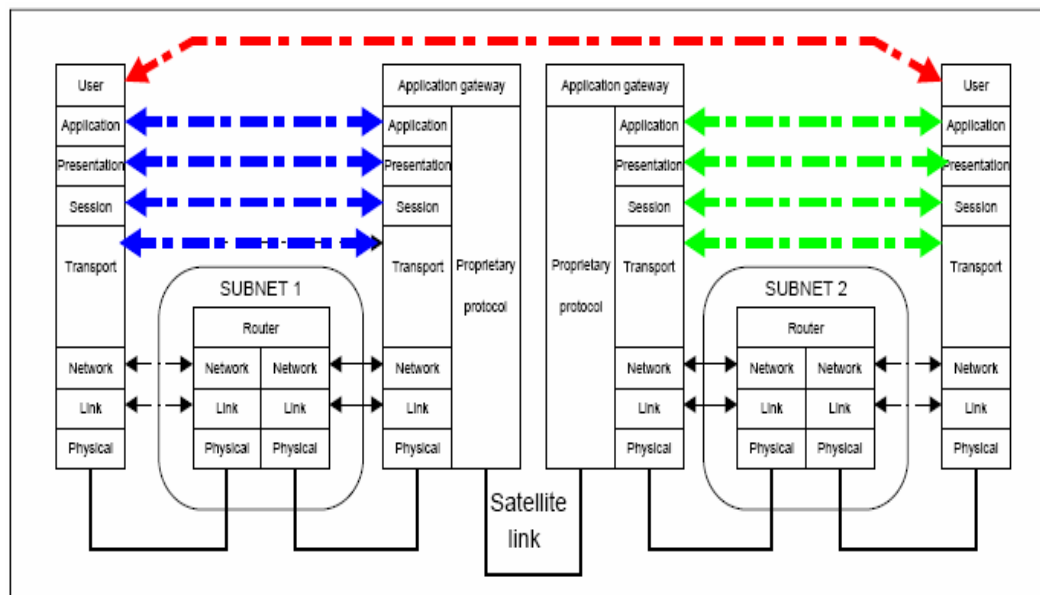


Figure 2: The Application Gateway protocol model

The communication is intercepted at the satellite terminals by the AGW that operates at application layer. This means that ATM messages are decoded by the AGW that then performs the following actions:

- Analyze and categorize the message in order to extract relevant parameters such as its length, type and priority.

- Attempt to build a schedule that satisfies the QoS requirements of all queued messages. It is assumed that, in particular, the remaining time-to-live can be deduced from time stamps, message type, priority, etc. The schedule may imply that some low priority messages that are getting close to their deadline are sent before higher priority messages that have a longer time-to-live.
- If such a schedule can be built, messages are transmitted over the satellite link according to the schedule, to the peer AGW that, in turn, delivers the message to the destination.
- If a schedule that meets the requirements of all messages cannot be built, messages are discarded according to set rules until a schedule for the remaining messages can be constructed.

The rules for building the schedule and discarding messages may be quite simple, such as

- Low priority is discarded in favor of high priority
- Certain message types are considered less important than others

but may also include criteria related to message context, e.g.

- If an old and a newer report of the same parameter are in the queue, the new one goes before the old one, or the older one is discarded in favor of the newer one
- Many exchanges consist of a dialogue of a few messages going back and forth. If the first message of a dialogue was successful, the subsequent messages have a higher value, since the dialogue will be restarted by the application if not completed
- Certain classes of messages must be delivered in sequence, others may tolerate re-ordering

The AGW may either be implemented as a proxy, i.e. its existence is transparent to the end systems, or it may be a network element with its own address, explicitly accessed by the end systems.

C. Problem areas

The AGW needs to know the detailed format of at least the headers of all messages that traverse it. ATN message formats are standardized already, so this is not a big problem, at least in the short term. But the ATM concept is likely to evolve, leading to the introduction of new messages in the future, which will require an update of all AGWs.

AGW message ordering/discard criteria may also evolve as the environment changes or new operational concepts emerge.

A second problem area is security.

Today, ATN defines only end-to-end application layer message authentication. Since the AGW does not modify messages, it will be transparent to this feature. But a migration from OSI-based ATN protocols towards IP-based protocols is likely to take place at some point in the future. This might include IPsec security. In this case, if the IPsec payload is encrypted, the message headers and contents are not available to the AGW. One (probably the only) solution to this is to terminate the IPsec associations at the AGW, that can then decrypt the message and transport it to its peer AGW either using IPsec or some other security measure local to the satellite link. Whether this is permissible is not clear. The AGWs would reside within the satellite terminals, i.e. within the security perimeter of ATM, so they might be considered trusted entities. But this does break the end-to-end security.

D. Future migration to TCP/IP

It is envisaged that ATM will eventually migrate from OSI protocols to a TCP/IP based network. It is expected that the ATN message formats will remain essentially the same (possibly evolving slowly themselves, but independently of any evolution of the underlying transport network).

TP4 and TCP are quite similar in concept, as are CLNP and IP. So little changes in the AGW concept if/when such a migration takes place. The core of the AGW remains basically the same, but the protocol stack towards the terrestrial and on-board networks change.

Actually, the AGW can assist a gradual migration: Since the AGW deals with messages, it is feasible to have, for example, an on-board OSI-based network and an IP-based terrestrial network. The AGW essentially decouples the evolution of the ground network, the on-board network and the satellite network. It is worth noting that a migration study, completely independently from and without knowledge of the AGW idea, came up with a very similar gateway for the purpose of facilitating a phased OSI to IP migration.

IV. Other potential solutions

A. Transport layer proxy (“PEP”)

Many satellite networks make use of a so-called Performance Enhancing Proxy (PEP) that splits the end-to-end path at transport layer into 3 consecutive transport connections: From sender to satellite terminal; from satellite terminal to satellite terminal; from satellite terminal to recipient.

It has been proposed to include PEPs in the ATM network. However,

- PEPs are aimed at a different problem: Mitigating the effects of propagation delay and propagation errors in high bandwidth x delay networks (“long, fat pipes”)
- PEPs may help mitigating the effects of the TP4/TCP congestion control not being well adapted to the ATM traffic profile
- *But PEPs cannot do anything to mitigate the real problem: Inelastic traffic over an elastic transport protocol.*

So a PEP is *not* a solution, though some seem to think so.

B. Unreliable transport protocol

Both the OSI and the TCP/IP protocol stacks contain an unreliable datagram protocol: CLTP in OSI, UDP in TCP/IP. It is not currently foreseen to introduce CLTP in ATN. But the use of UDP in a future migration to IP-based protocols is being discussed.

In case of congestion, a datagram transport protocol will discard data packets. Congestion will not allow ever-increasing queues to build up, so congestion collapse is no longer a threat.

But the datagram transport has a number of drawbacks as compared to the AGW solution:

- It discards *packets*, not *messages*. The difference may seem subtle, but is significant: A packet may contain several short messages, and a long message may be split into several packets. Discarding one packet of a multi-packet message means loss of the full message; discarding a multi-message packet may discard more than was intended
- Packets are discarded at random, possibly taking into account priority, but not type of message (a router is not supposed to look inside the packet)
- CLTP and UDP have no notion of “time-to-live” of a packet (except in the sense of hop count, which is a different issue altogether)
- Packets are typically discarded on entering the congested queue, rather than on exiting the queue. This means that fresh data is discarded in favor of older, possibly expired data that has been sitting in the queue for a long time

One might argue that these problems can be solved by applying additional “intelligence” in the packet discarding mechanism. If this is done, the result is really more or less the AGW described above, just under a different name of “intelligent router”, “packet scheduler”, or something like that.

By the way: The AGW, though initially intended for reliable transport, could just as well be used in conjunction with unreliable transport protocols in the tail networks

V. The AGW test bed

As part of the ESA SDLS project, a prototype of the AGW was developed and integrated into a demonstration environment, see Figure 3.

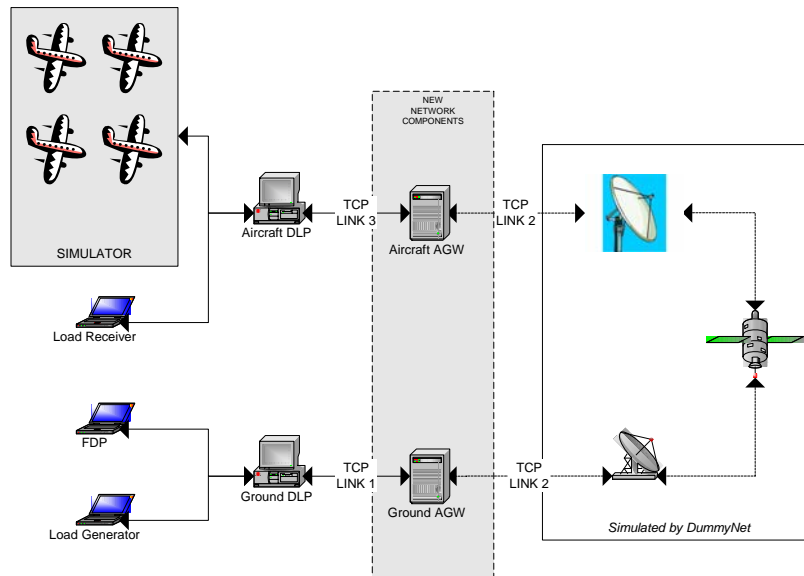


Figure 3: AGW prototype and demo setup

The test bed consists of

- Two prototype AGWs
- An emulated satellite link (using Dummynet)
- Ground and air Data Link Processors (DLP). They are the interface between the air/ground communication link and ground ATCC systems and aircraft avionics, respectively. These were real DLPs.
- A simulation of up to 50 aircraft, including track generation and display, and automatic communication message generation as a function of progress of the flights
- A Flight Data Processor, emulating the ATC centre controller work position, including message entry and reception and graphic display of messages as well as a simulated radar display
- Load generators and receivers in order to generate background load
- Data collection and statistics tools (not shown in the figure)

The test bed was based on TCP/IP, rather than ATN protocols. The mechanisms of TCP and IP were considered sufficiently similar to TP4 and CLNP that this was warranted, taking into account that the aim of the test bed was to validate and demonstrate the AGW concept, not to provide a first step in the development of real AGWs. Another reason for choosing TCP/IP was that the test bed setup relies heavily on already available network elements, in particular for creating a demonstration environment representative of a real air traffic control environment.

The test bed used a relatively simple decision algorithm based on priority and remaining time to live of each message. It did not implement any more sophisticated features involving e.g. memory of past traffic.

The test bed can be operated in two modes: With AGWs in place, and with end-to-end TCP, bypassing the AGWs. The purpose of the latter mode is to serve as reference for comparison of performance with and without AGW.

VI. Test bed results

Three types of test were carried out:

- Very light load. The objective is to verify that the AGW interferes only minimally with traffic when no congestion is present
- Very heavy load. The objective of this test is mainly technical: Verify that the AGW performs as designed under heavy congestion. This test is not representative of any foreseen operational situation
- Operational heavy load situation. The traffic load is somewhat below congestion most of the time, with short periods of congestion. The objective is to show that the AGW can improve overall performance significantly under light congestion.

The tests were carried out with a mix of 3 types of messages. This is a simplification compared to a full operational scenario that would likely support more message types:

- CPDLC (Controller-Pilot Data Link Communication). These are high-priority, urgent messages typically containing clearance requests, clearances and clearance responses (e.g. “Climb to flight level 290” and the response “Wilco”)
- FLIPCY (Flight Plan Consistency). These were considered of medium priority and urgency.
- ADS-C (Automatic Dependent Surveillance – Contract) reports. These are regular position reports. Because the reports are repeated at rather short, regular intervals, we considered these of low priority. We did not consider the possible additional intelligence in the AGW to keep track of discarded reports, and attempt not to discard consecutive reports from the same aircraft.

The message traffic was based on realistic flight profiles, and both the initiating and the receiving end systems produced graphical display of the traffic, including maps showing aircraft positions. Showing both sending and reception of data enabled us to show quite convincingly whether, in any given scenario, data was delivered promptly, delayed, or not at all.

C. Very light load – results

All messages were delivered within their time-to-live.

Most messages were delivered in their original sequence, but re-ordering occurred occasionally when messages of different types happened to be in the queue at the same time.

The TCP reference case showed similar behavior, but always delivered messages in the original sequence (inherent property of TCP).

D. Very high load – results

A summary of the results of a test run is as follows:

HIGH PRIORITY MSGS	With AGW	Without AGW
Transmitted Messages	2500	2500
Messages delivered in Time	2500	17
Average Delay	1748.55 ms	41587.91 ms
MEDIUM PRIORITY MSGS	With AGW	Without AGW
Transmitted Messages	5000	5000
Messages delivered in Time	1721	681
Average Delay	17221.53 ms	41599.73 ms

LOW PRIORITY MSGS	With AGW	Without AGW
Transmitted Messages	2500	2500
Messages delivered in Time	755	789
Average Delay	21127.49 ms	41613.75 ms

As already mentioned above, this test is not representative of any plausible operational situation: The offered traffic is in the order of twice the capacity of the network.

It is nevertheless interesting to see what happens in such an extreme case.

With the AGW, all high priority messages (with a short time-to-live) are delivered in time. About one third of the medium priority messages and one third of the low priority are also delivered in time.

Without the AGW, it can be seen that the average delay is much higher, and that only about 15% of all messages are delivered in time. Only a few of the high priority messages are delivered in time due to their short time-to-live.

It should also be noted that the results with AGW are stable irrespective of the duration of the test while in the test without the AGW, the delay will keep growing (congestion collapse) since TCP insists on delivering all data, while the AGW discards data that cannot be delivered in time.

E. Normal load – results

A summary of the results of a test run is as follows:

HIGH PRIORITY MSGS	With AGW	Without AGW
Transmitted Messages	2500	2500
Messages delivered in Time	2500	1200
Average Delay	1369.45 ms	5441.99 ms
MEDIUM PRIORITY MSGS	With AGW	Without AGW
Transmitted Messages	5000	5000
Messages delivered in Time	5000	5000
Average Delay	9879.62 ms	5465.45 ms

LOW PRIORITY	With AGW	Without AGW
Transmitted Messages	2500	2500
Messages delivered in Time	1657	2500
Average Delay	19863.17 ms	5480.58 ms

With AGW, all messages with high and medium priority are delivered in time, while about one third of the low priority messages are dropped.

In comparison, without AGW, all medium and high priority messages are delivered in time, while only half the high priority messages are delivered.

VII. Conclusions

The paper offers a solution to the problem of transferring inelastic traffic over an elastic transport protocol: The Application Layer Gateway (AGW). It is recalled that, in such a scenario, the network should be dimensioned for the foreseen peak load plus some margin, but that it has to be assumed that congestion will occur, be it rarely. It is essential that the occurrence of congestion does not lead to collapse of the network. In order to achieve this, there is only one solution: Re-order and, if necessary, discard traffic in order to relieve the congestion. The AGW approach is a way of selectively re-ordering and discarding traffic. An alternative, but less performant solution is also briefly discussed: The use of a datagram transport protocol.

A test bed has been built, and a brief summary of results is presented, confirming the feasibility of the AGW and its capability to achieve the set goal.