4

Transport Protocol for Future Aeronautics

Muhammad Muhammad and Matteo Berioli
German Aerospace Center (DLR), Oberpfaffenhofen
Germany

1. Introduction

Transmission Control Protocol (TCP) (Postel, 1981) is the vastly researched, used and implemented transport protocol from the bouquet of standardized protocols we have. Since its first introduction, TCP has experienced tremendous revolutions specifically regarding the way it should control congestion (Allman et al., 2009). Many TCP flavors exist, one can control its sending rate according to link bandwidth estimation like in the Westwood variant (Mascolo et al., 2001), while another tunes its congestion window (cwnd) based on the round-trip time (RTT) evaluation such as TCP Vegas (Brakmo & Peterson, 1995), just to name few examples. However, all of these TCP versions assume a bulk transfer of data and thus they are optimized upon this type of traffic.

On the other hand, Air Traffic Management (ATM) data traffic has other characteristics than file transfer. Messages generated from aeronautical services are triggered by events in the aeronautical environment. Further, these messages have bursty inter-arrival times, which can go up to several minutes, relatively small size mostly in the order of few hundreds of bytes and a maximum delay (a maximum of few seconds) that they have to be delivered within. As can be realized, a TCP source may experience some inactive periods due to the burstiness of the traffic. However, in (Jacobson, 1988), it is recommended that to control and avoid congestion, TCP should use slow start mechanism when restarting a transmission after a ratherly long idle period.

The NEWSKY project (NEWSKY, 2009) recommends to design a transport protocol for avionics which is based on the User Datagram Protocol (UDP) but with reliability measures. Taking these suggestions into account, the Reliable User Datagram Protocol (RUDP) (Bova & Krivoruchka, 1999) is an interesting option. However, it has features from TCP, which are not desirable for the ATM context, like connection initiation and shutdown, congestion control and byte streaming mechanism, which according to the authors of (NEWSKY, 2009) and (Muhammad & Berioli, 2010) reduce the performance of a transport protocol when transferring small files. Therefore, RUDP could be a highly qualified candidate to transport aeronautical traffic in case it is carefully re-designed and adjusted to consider the properties of an ATM traffic over highly asymmetrical links.

Furthermore, in the context of checking the validity of TCP to transfer messages of ATM services, questions related to the performance of the protocol will be addressed. For example, the relation between the *cwnd* of TCP, aeronautical messages and congestion control.

The Seamless Aeronautical Networking through integration of Data links, Radios and Antennas (SANDRA) (SANDRA, 2011) concept consists of the integration of complex and disparate communication media into a lean and coherent architecture. SANDRA as a project is divided into several sub projects (SPs). This work belongs to the one dealing with seamless networking specifically for the transport layer protocols and performance task. Transport layer protocols should be assessed with respect to their suitability for aeronautical communication and their impact on the system availability and reliability. Finally, an optimization for these protocols in order to be used in avionics context should be provided.

The work in this chapter is organized as follows. In the next Section, an overview about the aeronautical services is presented. In Section 3, the system architecture design in which the protocol will be operating is discussed. Section 4 explains some properties and illustrates few drawbacks of using TCP in avionics from theoretical and technical perspectives. The RUDP is further investigated in Section 5 where adjustment recommendations are given. In Section 6, detailed analysis of the ATM traffic pattern is shown and suggestions on system design are provided. Finally, a conclusion is drawn in Section 7.

2. Aeronautical services

The currently operating Aeronautical Telecommunication Network (ATN) protocol is based on the ISO/OSI stack and uses the TP4 protocol (ITU-T, 1995) via Dialogue Service (DS). However, the future envisaged protocol is expected to be based on the IPv4/v6 protocol stack, as identified in (NEWSKY, 2009).

The study of potential future communications technologies to meet safety and regularity of flight communications requirements, i.e. those supporting Air Traffic Services (ATS) and Airline Operational Control (AOC), have been initiated by the European Organization for the Safety of Air Navigation (EUROCONTROL) and the Federal Aviation Administration (FAA). The second version of the Communications Operating Concept and Requirements (COCR) document (EUROCONTROL & FAA, 2007) identifies the requirements placed on the communications that take place between the aircraft and ground radios, i.e. the air-to-ground (A/G) and ground-to-air (G/A) links. Further, the COCR divides the airspace into five domains. Thus, ATM services vary according to the domain in which the position of the airplane will be. Figure 1 illustrates the airspace domains, which are:

- Airport (APT) consists of the airport and the close area surrounding the airport.
- Terminal Maneuvering Area (TMA) also surrounding the airport but on a larger scale.
- En Route (ENR) is the continental or domestic airspace used by Air Traffic Control (ATC).
- Oceanic, Remote, Polar (ORP) is the same as ENR but it covers geographical areas generally outside the domestic airspace.
- Autonomous Operations Area (AOA), in this block of airspace, the airplane is self-separate, i.e. ATC is not used.

The corner stone of this work is to introduce a reliable communication environment for the ATS and AOC services. The ATS applications are the most critical for the safety of the flight. They involve the messages between the cockpit and the control center, i.e. the ATC center in the airport. Paired to these services are the AOC, which are primarily concerned with the safety and regularity of the flight. AOC applications are responsible for the communication between the aircraft and the AOC center, company or operational staff at an airport.

The messages generated by these applications have inter-arrival times in the order of seconds up to several minutes. Moreover, each message has a maximum latency time that it has to be delivered within. Typically, this time is lower than a message inter-arrival time. It is also worth to mention that the ATM traffic is inelastic; this means that these services cannot adjust their

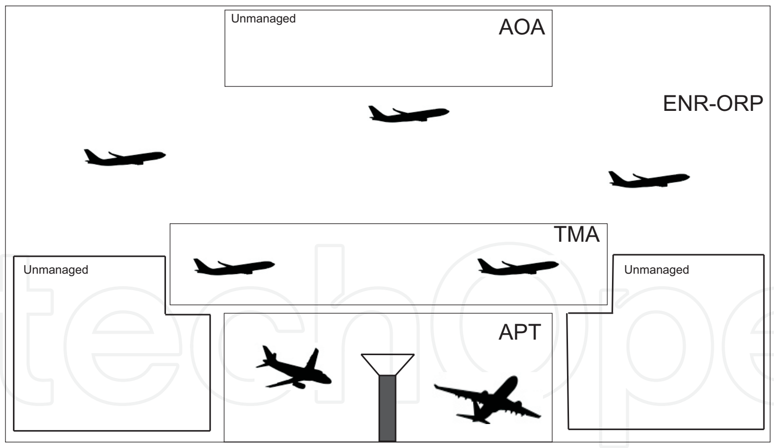


Fig. 1. Airspace domains as envisaged by the COCR document, namely, APT, TMA, ENR, ORP and AOA

message generation to the network conditions but they require a minimum level of resources, see (EUROCONTROL & FAA, 2007).
Within the COCR document, there are 32 ATS services, 21 belong to AOC and 2 NET services. The messages generated by these services have relatively small size with respect to a common file on the Internet. For example, the *FREETEXT* message is only 377 *bytes*. This service represents a textual message between the cockpit and the AOC center. Very few messages are of kilobytes size. The shortest message size is 82 *bytes* while the largest is around 21 *kilobytes*, which deals with the weather reports and is called *WXGRAPH*. Section 6 will present detailed analysis about the aeronautical traffic pattern.

3. System architecture

The trend in the aeronautical sector, driven by an increasing number of flights, is the introduction of new communication services and the transition of the ATM from analogue techniques towards digital communication services results in the development and deployment of new and heterogeneous link technologies, see (Kissling, 2009). One example of a link technology under development for direct A/G radio communication is the L-band Digital Aeronautical Communication System (L-DACS)-1 communication standard (Brandes et al., 2009) and (Graupl et al., 2009). Another example is the European Space Agency (ESA) Iris programme (ESA, 2009) using satellite communications, which is currently under development. The new link technologies will complement the existing ones (like VHF Digital Model (VDL)-x) and will also coexist with future, short range links such as WiMAX as part of the communication infrastructure in the APT area. The system architecture under consideration in this work requires the aircraft to have several link technologies available for communication with the ground, which all have heterogeneous properties. One of the most important properties changing with the link is the propagation delay with RTTs in the order of 500 *ms* for GEO satellites down to few milliseconds for direct A/G links. Besides the delay, the communication bandwidth and datarates change as well with the links, starting with few bits per second and reaching higher datarates with kilobits per second. Finally, also from the channel characteristics, packet error and loss rates may change with the link technology. The integration of these different links into one network results in a system architecture as illustrated (simplified) in Figure 2. As shown there, only the ATS and AOC services are considered. Each of these services may have one or

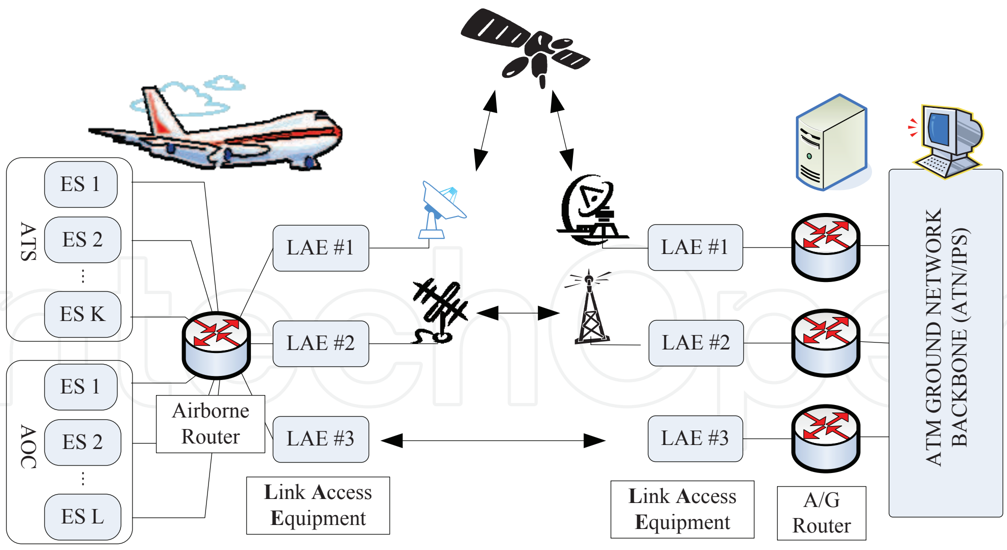


Fig. 2. System architecture

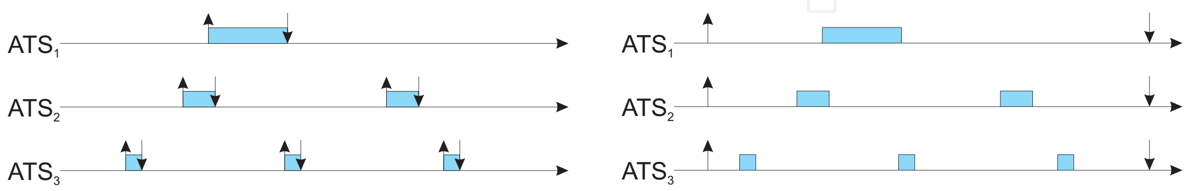
several End Systems (ESs), which generate and receive data messages. A data message is transmitted via the airborne router as one or several IP packets over a link with a specific link layer and physical layer protocols belonging to the selected Link Access Equipment (LAE). On the ground side, the network layer IP datagrams are then routed via the ATM ground network (assumed as a backbone ATN/IPS network) towards the correspondent ES in the ground network. For safety related operational services ATS and AOC, reliable message transmission is a key demand since loss or corruption of messages can have severe consequences. Transport layer protocols, which shall guarantee the reliable transmission of data should work efficiently over all different link technologies present. For provision of transmission reliability in the future ATN/IPS network, up to now the TCP protocol has been envisaged (ICAO, 2010). The different conceptual options to use the TCP protocol and the requirements to a transport protocol have been reported in (Kissling & Graupl, 2008) and (Ehammer, Graupl, Rokitansky & Kissling, 2009). In (Kissling & Baudoin, 2008), different possibilities for protocol stack architectures in the ATM scenario have been investigated. The major key issues for using transport layer protocols in the ATM environment found in this work were the provision of a reliable communication service which makes best possible use of the scarce communication bandwidth and does not require coexistence of different transport layer protocols in parallel. Deployment of several link-specialized transport protocols in parallel would cause higher cost for maintenance, management and especially for the standardization procedure which every transport layer protocol has to go through.

4. Transmission control protocol

TCP is the commonly used protocol by many of today's Internet most popular applications like Hypertext Transfer Protocol (HTTP) and File Transfer Protocol (FTP). It provides reliability and ordered delivery of a stream of bytes being transmitted from a program on one computer to another over an unreliable network.
TCP is a connection oriented protocol. That is, a connection should be established before sending data. Depending on the RTT between the two communicating nodes, a waiting time of:

$$T_w = 2 \cdot RTT \tag{1}$$

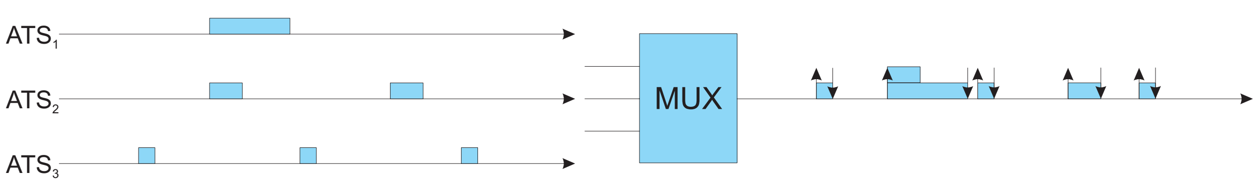
is measured before receiving the first byte of any actual information.
The work done by (Ehammer, Graupl, Rokitansky & Kissling, 2009) describes several possibilities on the way that TCP could be operated over aeronautical networks and showed the advantages and drawbacks of each method. These methods deal with the multiplexing or not of the ATM services and the network layer interaction. These techniques are: (a) re-establishing of connections for each transmission and for every service (Figure 3(a)), (b) establishing a connection once for each service and keep it open (Figure 3(b)), (c) re-establishing of the connection for each transmission of a multiplexed set of services (Figure 4(a)) and (d) establishing a connection for a multiplexed set of services and keep it open (Figure 4(b)).



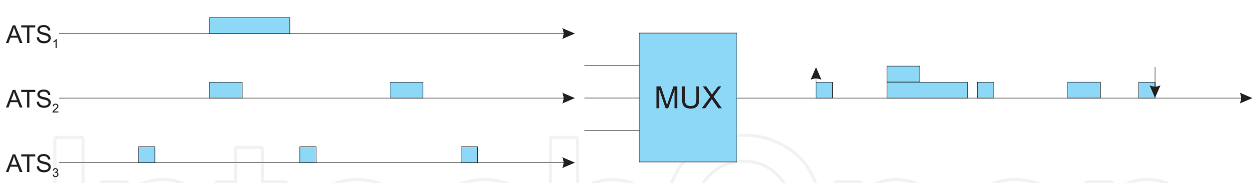
(a) Connection established/shutdown for every transmission

(b) Connection established/shutdown once for every service

Fig. 3. Transport layer session connection establishment/shutdown with no services multiplexing



(a) Connection established/shutdown for every multiplexed transmission



(b) Connection established/shutdown once for all multiplexed services

Fig. 4. Transport layer session connection establishment/shutdown with services multiplexing

Further, in (Ehammer, Graupl & Rokitansky, 2009), the authors discussed several well known TCP optimization techniques and found out that for TCP to operate over aeronautical communication network, a minimum of a certain link capacity should be available per user, else TCP will run into performance problems.
Moreover, within the aeronautical environment, where the messages to be transmitted in a flight are scarce, maintaining a single TCP connection is cumbersome. This is due to the fact that within a flight the aircraft will change various links with different capacities and delays, giving rise to links handover management overhead. In (Daniel & Kojo, 2008) and (Daniel et al., 2008) the authors had developed cross-layer assisted TCP sender algorithms

to reduce the unnecessary packet retransmissions and congestion control actions due to vertical handovers. In other words, they developed mechanisms to help TCP to reduce retransmission overhead when operated in wireless environment, in which handovers may occur on events such as the links properties are interchanged between high/low delays, high/low bandwidth or upon retransmission timeouts (RTOs) during disconnection.
On the other hand, allowing TCP to establish a session for every transmission brings three disadvantages. Figure 5 illustrates this approach in comparison with the single TCP connection (discussed previously). First, for every message to be received, T_W should be waited before the first bit of the actual data delivery, which affects the delivery time of the message. Secondly, the overhead introduced by TCP, because of the connection initiation and shutdown is on average large compared to the sizes of the messages produced by the ATM services. Finally, the *cwnd* of a TCP sender will not be able to capture the congestion status when operating below aeronautical services because of the traffic pattern they adhere.

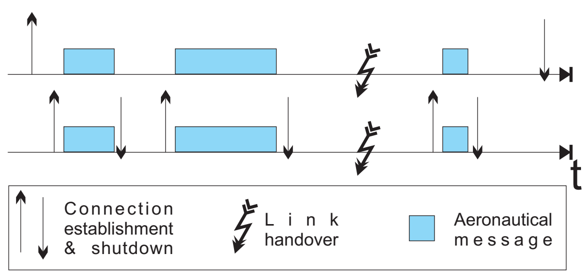


Fig. 5. Transport layer (TCP) connections as envisaged in (c) and (d)

In (Muhammad & Berioli, 2010), we developed a model to theoretically estimate the overhead produced by different transport protocols when transferring a file. The investigations showed that for small file sizes, reliable, connectionless transport protocols with no congestion and flow control have the lowest overhead cost. In this work, we extend the model to measure the overhead produced by the connection establishment and shutdown mechanisms of TCP.
According to the specification of TCP in (Postel, 1981), to start a session, a client should send a SYN packet to the server that will respond with a SYN − ACK and finally the client will confirm with an ACK (3-way handshake). On the other hand, when the server finishes transferring the file, it will send a FIN packet, and the client will reply with an ACK. Further, when the client makes sure that it has the complete contents of the file, it will send another FIN to the sender and wait for the ACK packet. These two mechanisms are bi-directional and thus we split our model based on the forward (FW) and return (RT) channels, respectively.

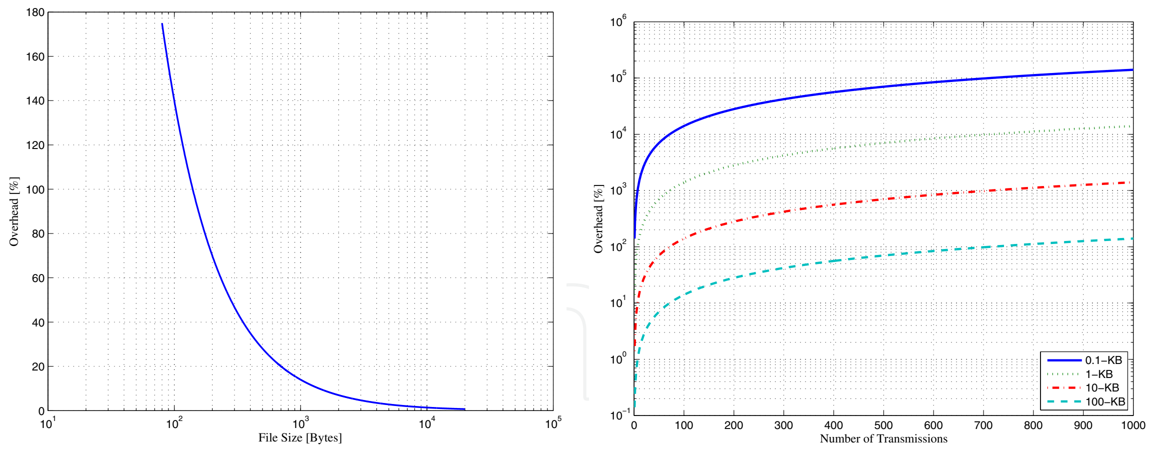
$$\begin{aligned} S_{\mathrm{HC}}^{\mathrm{FW}} &= H_{\mathrm{SYN-ACK}} + H_{\mathrm{FIN}} + H_{\mathrm{ACK}}, \\ S_{\mathrm{HC}}^{\mathrm{RT}} &= H_{\mathrm{SYN}} + H_{\mathrm{FIN}} + 2 \cdot H_{\mathrm{ACK}} \end{aligned} \tag{2}$$

where $S_{\mathrm{HC}}^{\mathrm{FW}}$ and $S_{\mathrm{HC}}^{\mathrm{RT}}$ are the sizes of the overhead on the FW and RT channels, respectively. Moreover, H_{X} is the size of the header of the packet X, bearing in mind that TCP connection establishment and shutdown packets are only TCP headers.
Now, the overall overhead related to a connection start and end in TCP amounts for:

$$H_{\mathrm{C}} = S_{\mathrm{HC}}^{\mathrm{FW}} + S_{\mathrm{HC}}^{\mathrm{RT}} \tag{3}$$

Therefore, the TCP session overhead produced for a file of size F is:

$$OH_{\mathrm{C}} = \frac{H_{\mathrm{C}}}{\mathrm{F}} \tag{4}$$



(a) One message per transmission (b) Multiple connections

Fig. 6. Overhead estimation of a TCP connection establishment and shutdown

The two plots in Figure 6 show the overhead generated for different message sizes and the overhead cost for transmitting the same file multiple times with different TCP connections. For this simulation, we used the minimum TCP header size (20 *bytes*) assuming no options are used. In Figure 6(a), we show the connection overhead percentage (on the *y-axis*) generated for different sizes of messages in the range of the ATM services messages size (*x-axis*). Here, we assume a single message is transferred in a connection. As can be clearly seen that the overhead of sending a message of small size is considerably large.
Moreover, Figure 6(b) represents the overhead generated for multiple file sizes with several TCP connections. For instance, when simulating a file of size 10 *kilobytes*, the *x-axis* represents the number of transmissions of that file each time with a new connection initiated and shutdown. As it can be realized, the higher the number of transmissions, the higher the overhead. Conversely, the larger the file, the lower the overhead.
Taking the complete traffic profile characteristics into account and not a single message as above, Figure 7 shows the *cwnd* (displayed with a *line*), in bytes, of TCP versus the actual transmitted bytes (represented by *stars*). It can be clearly realized that the *cwnd* of TCP cannot cope with the real transmitted information. We can say that the *cwnd* of TCP was deceived by the incoming application data. Thus, it shows a very oscillating behavior and this is related to the long idle periods between two consecutive transmissions, which is associated with the inter-arrival times of the ATM messages. Further, the average size of this *cwnd* (reflected by the *dotted line*) is very close to the value of the average *cwnd* resulted in the simulations done by (Ehammer, Graupl, Rokitansky & Kissling, 2009).

5. Candidate protocols

In the summer of 1984, the standardization of the Reliable Data Protocol (RDP) was held by (Velten et al., 1984) with the idea of running applications such as remote loading and debugging. Six years later, their colleagues at BBN Communications Corp. updated the standard with a newer version (Partridge & Hinden, 1990). In short, RDP is connection oriented and offers reliable transport to efficiently support the bulk transfer of data.
Moreover, based on these two standards with the same protocol properties, RUDP defined in (Bova & Krivoruchka, 1999) (IETF Internet-draft) extended the primitive[1] UDP (Postel,

[1] Because, according to the authors, TCP is too complex.

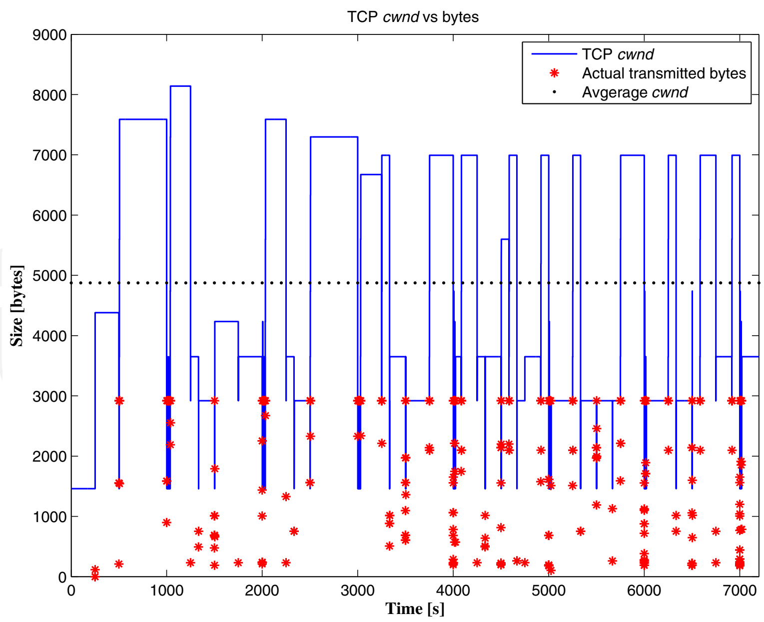


Fig. 7. *cwnd* of TCP vs the actual transmitted bytes from the aeronautical services

1980) with window and flow control, acknowledgement mechanism, retransmission of packets lost on the way and avoidance of buffer overflow in order to transport telephony signaling.

As mentioned in Section 2, each aeronautical service sends a different message. The generation of these messages is triggered by events. The services at the two ESs, i.e. the airplane and the control center, send their messages independently. Further, the generated messages differ significantly among each other in terms of message arrival time, size and latency requirements. However, almost all of them require reliable delivery service by the transport protocol. Some services do not require full reliability like surveillance ones because they are generated more frequently, see (Ehammer, Graupl & Rokitansky, 2009). Thus, a reliable transport protocol to be designed should be aware of the properties of these messages.

Figure 8 illustrates the protocol stack using the provisioned transport protocol above the Internet Protocol (IP) layer that takes care of the addressing and below the aeronautical services labeled $A_1, A_2, ...$ An aeronautical transport protocol should provide reliability, ordered delivery and honors message boundaries like UDP. Further, it should be IP based and connectionless transport protocol in order to reduce the session initiation and shutdown overhead. Finally, it should be able to cope with highly asymmetrical communication channels such as satellite links.

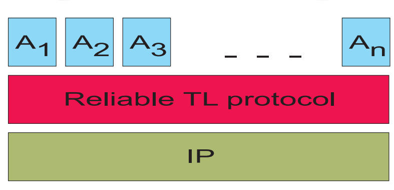


Fig. 8. The protocol stack to be used by a reliable transport UDP-based protocol such as enhanced version of RUDP

Aeronautical communications involve layer 2 (Media Access Control (MAC) layer) congestion control mechanisms such as L-DACS. Therefore, the motivation behind removing the

congestion control methods from the transport layer (layer 4) is to prevent any further decrease of the performance in case the two layers mechanisms clash. Further, in this case, it makes more sense to move the congestion control to the MAC layer since they have the knowledge about the links properties and thus there is no need for cross-layer communication. Therefore, reducing system complexity.
Furthermore, in (Ehammer, Graupl & Rokitansky, 2009), it was shown that the *WXGRAPH* message on the FW link (sized 21 *kilobytes*) is the one that is congesting the network due to its large size compared to the other services. One proposal could be, the implementation of a dual stack. Where TCP takes care of transmitting the messages from this service and keeps the others of small sizes for the new protocol.

6. System dimensioning

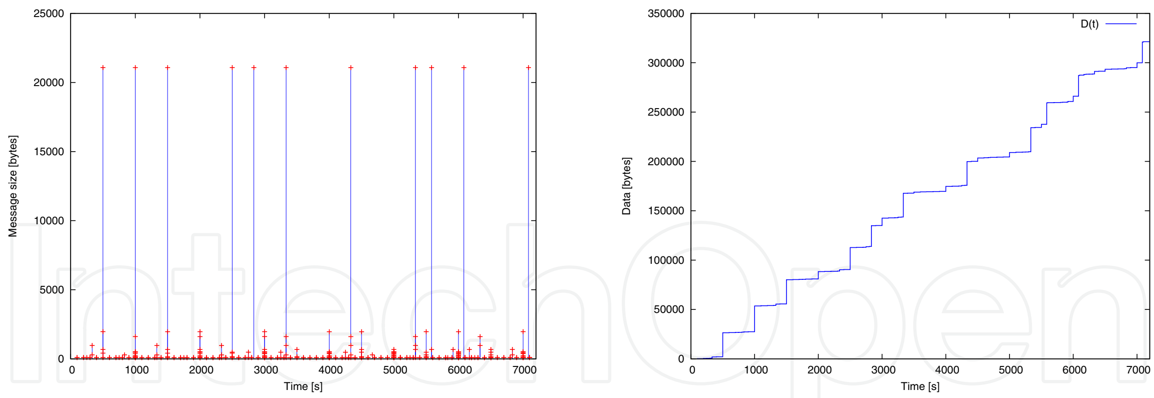
The requirements of the aeronautical traffic mandate a certain behavior from the lower layers. This behavior is governed by the properties offered by the system. This section will elaborate more on the aeronautical traffic properties. Additionally, some recommendations on the system properties will be derived.

6.1 Aeronautical traffic properties

As described in Section 2, messages generated by aeronautical services vary in size, inter-arrival times and latency requirements. These variations do not only occur between two different airspace domains or links (A/G and G/A links) but also within the same transport layer connection in case all services are multiplexed on the same transport session, as illustrated in Figure 4(b), which is an option from the transport layer session management mechanisms highlighted in Section 4. Further, Figures 9 and 10 show an aeronautical traffic pattern on the FW and the RT links, respectively. These plots show a sample test for two hours flight simulation in the TMA-ENR domain based on statistical expected traffic reports from (Rokitansky et al., 2008).
Figures 9(a) and 10(a) show the size of the messages and their inter-arrival times from the application layer perspective. It is clear to see that the size of the messages fluctuates heavily especially on the FW link; on the RT link the largest measured message size is much smaller than the one on the FW channel; the inter-arrival times of these messages vary not only among the different services but also within every service as well. Nevertheless, few services are generated periodically.
Figures 9(b) and 10(b) represent the data flows of these aeronautical messages. These data flows can be described by means of the cumulative function $D(t)$, defined as the number of bits seen on the flow in the time interval $[0, t]$:

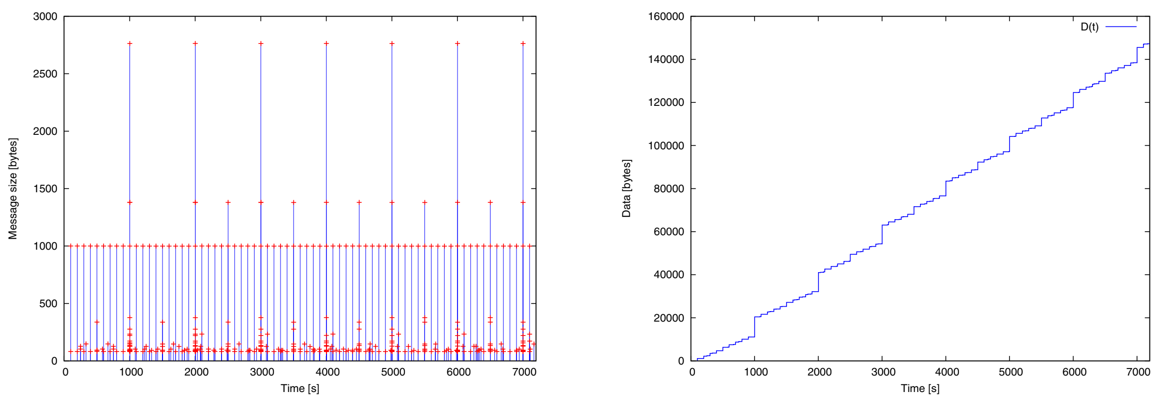
$$D(t) = \sum_i l_i \tag{5}$$

where l_i is the length in bytes of the message i.
The plots shown in Figure 11 represent the amount of bytes related to a specific service - identified by its size on the *x-axis*. For instance, the upper plot tells us that 80 % of the traffic data on the FW channel is generated by the *WXGRAPH* service (in which a single message amounts to 21 *kilobytes*). Additionally, all other services generate the remaining 20 % of the data traffic. In other words, at the network layer this can be seen as a flow of packets with more than 80 % of large sized datagrams and ca. 12 % of packets less than the maximum transmission unit (MTU) size of 1500 *bytes*.



(a) Messages generated on the FW link (b) The cumulative function $D(t)$ on the FW link

Fig. 9. Aeronautical traffic flow on the FW link in the TMA-ENR domain



(a) Messages generated on the RT link (b) The cumulative function $D(t)$ on the RT link

Fig. 10. Aeronautical traffic flow on the RT link in the TMA-ENR domain

On the other hand, only a single aeronautical service on the RT channel generates a message of size greater than the MTU size. However, this message can be fragmented only in two IP datagrams, if we also consider an MTU size similar to the one on the FW channel. Further, one can realize from the chart that almost 60 % of the traffic on the RT channel is generated by a service message of size 1000 *bytes*.

6.2 System properties

One of the major differences between an aeronautical application and a file transfer application, like FTP, is that avionics services are *inelastic*. This means that a service requires a certain minimum level of bandwidth and a certain maximum latency. In other words, an aeronautical application cannot adjust its rate, for example, to changeable network conditions. By contrast, an elastic application can adapt to network conditions. A file transfer, for instance, can easily adjust to different level of available bandwidth and latency as it has no severe latency requirements.

Based on the above mentioned facts and in order for the system to function properly, it is fundamental to define some minimum system requirements such as serving rate and queues length. Figure 12 illustrates a simplified network architecture in which the ATS/AOC client messages are buffered in a queue on the mobile router. This set-up is implemented on every

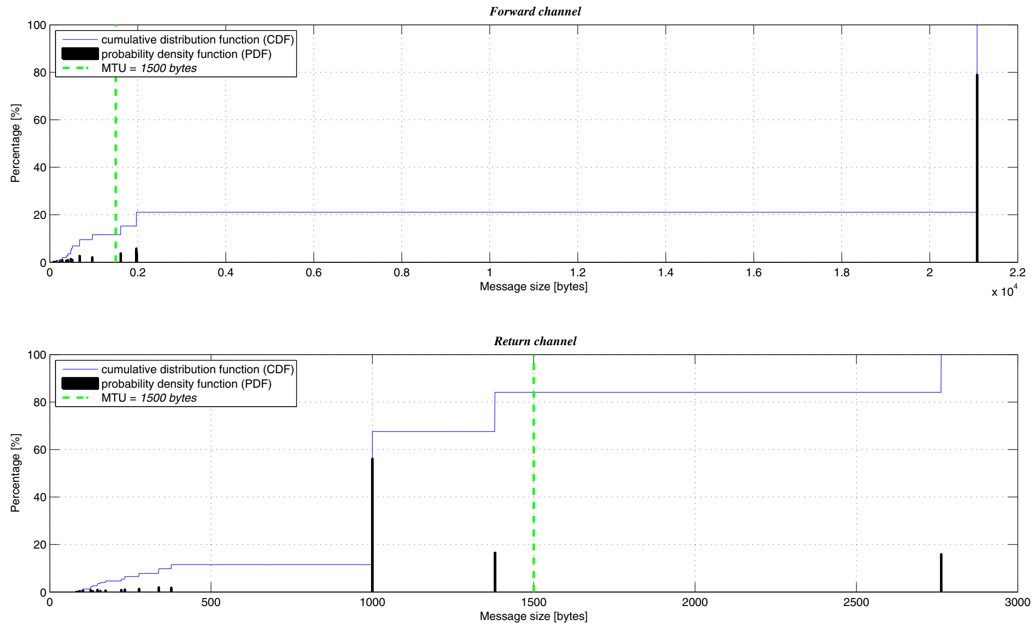


Fig. 11. Amount of data traffic generated by different ATM messages on the FW and RT channels, respectively

aircraft. Access to the wireless link to connect to the ground station is granted through layer 2, in the protocol stack (MAC layer). At the other end, i.e. the ground segment, the corresponding servers use a single queue, at the ground station, to buffer the messages directed to all airplanes.

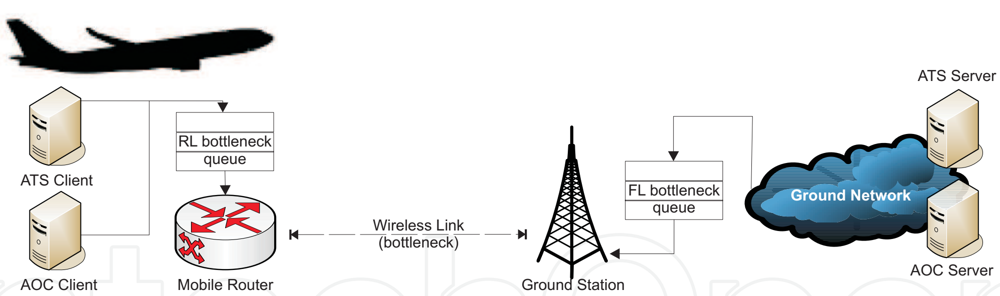


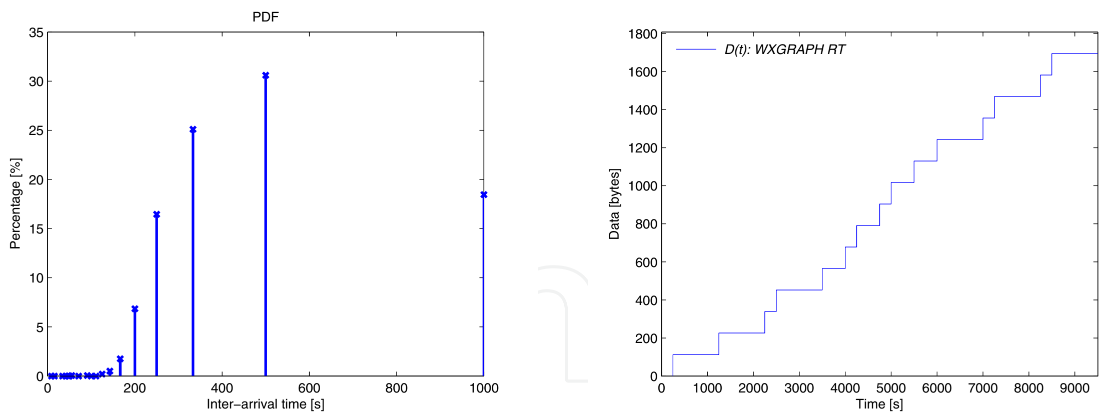
Fig. 12. Simplified network architecture

To illustrate this, let's assume that we have on the RT channel the service *WXGRAPH*, which is transmitting messages of a fixed size equal to 93 *bytes* with inter-arrival times illustrated in Figure 13(a). Figure 13(b), on the other hand, represents a possible realization of $D(t)$, as defined in (5).

Messages or packets (of fragmented messages) arrive (with a cumulative amount of data $D(t)$) at a buffer of size B and they are served in a First In First Out (FIFO) order with a rate r. The serving rate r exhibits the time needed for a message of size l to be transmitted on the link, in general:

$$t = \frac{l}{r} \tag{6}$$

This scenario can be easily modeled using a leaky bucket as shown in Figure 14. A leaky bucket controller, according to (Boudec & Thiran, 2001), is a device that handles the data flow



(a) PDF of the inter-arrival times for the *WXGRAPH* service on the RT link

(b) Cumulative function $D(t)$ of the *WXGRAPH* service on the RT link

Fig. 13. Properties of the *WXGRAPH* service on the RT link as reported in (Rokitansky et al., 2008)

$D(t)$ as follows. There is bucket of fluid of size B. The bucket is initially empty ($t = 0$). The bucket has a hole and leaks at rate of r units of fluid per second when it is not empty. The data flow $D(t)$ pours into the bucket in the time interval $[t_0, t]$ an amount of fluid equal to the amount of data $D(t) - D(t_0)$. Data that would cause the bucket to overflow is discarded.

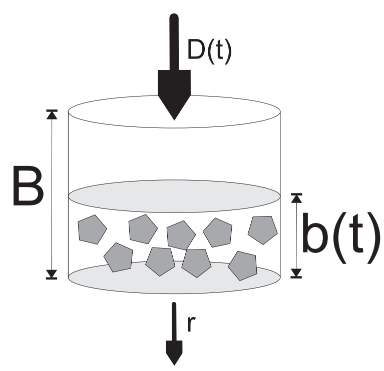


Fig. 14. Illustration of a leaky bucket model, pentagons represent the conformant data

6.2.1 Serving rate: r

Before deriving the minimum value of the required serving rate r, we need to make some assumptions such that the following work is clearer. First, let G_ζ describe a traffic generation stochastic process with emission rate λ_ζ, which is a stochastic variable with $\bar{\lambda}_\zeta = \mathrm{E}\left[\lambda_\zeta\right] = \frac{1}{\bar{t}_\zeta}$, where $\bar{t}_\zeta$ is the average inter-arrival time. Further, assume that the emitted traffic is reaching the queue (the object to be studied) with the same rate, λ_ζ. So, λ_ζ is also an arrival rate process. Then, we denote by:

$$G = \sum_\zeta G_\zeta \tag{7}$$

the total processes with an aggregate generation rate of:

$$\lambda = \sum_\zeta \lambda_\zeta \tag{8}$$

where $\zeta \in \Phi$, which is the number of processes. This follows:

$$\bar{\lambda} = \sum_{\zeta} \bar{\lambda}_{\zeta} \tag{9}$$

where $\bar{\lambda}$ is the average aggregate arrival rate of the process G, and the average inter-arrival time is:

$$\begin{aligned} \bar{t} &= \frac{1}{\bar{\lambda}} \\ &= \frac{1}{\sum_{\zeta} \bar{\lambda}_{\zeta}} \\ &= \frac{1}{\sum_{\zeta} \frac{1}{\bar{t}_{\zeta}}} \end{aligned} \tag{10}$$

At the first glance, we are interested in the behavior of a single process/service G_{ζ}, as if it is being served by a FIFO queue with serving rate r_{ζ}.
For a particular service G_{ζ}, if the number of arrivals in an interval $[0, t]$ is $\alpha_{\zeta}(t)$, then we can define:

$$\lambda_{\zeta}(t) \triangleq \frac{\alpha_{\zeta}(t)}{t} \tag{11}$$

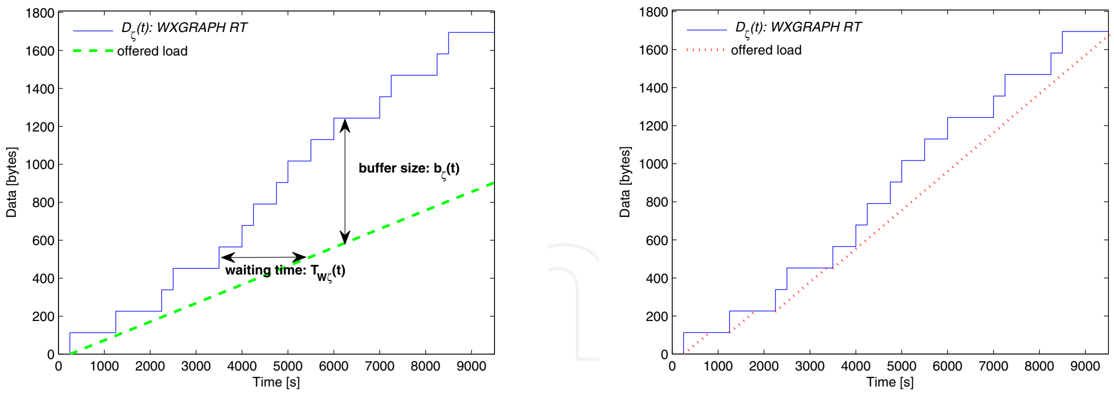
as the arrival rate of the service ζ. Consequently, the average arrival rate and inter-arrival time of the service can be deduced from the following; by assuming that this limit exists:

$$\begin{aligned} \bar{\lambda}_{\zeta} &= \lim_{t \to +\infty} \lambda_{\zeta}(t) \\ &= \frac{1}{\bar{t}_{\zeta}} \end{aligned} \tag{12}$$

The service rate r_{ζ} allows us to determine the time needed for a service message of size l_{ζ} to be transmitted on the link, which is specified in (6). As illustrated in Figure 15(a), if the serving rate is low, then the upcoming messages will be buffered, until the current ones are processed, and therefore all message are delayed.
Now, given the fixed message length l_{ζ} of a service ζ and the probability density function (PDF) of the inter-arrival times, we can determine the minimum serving rate required for this service. The minimum required serving rate r_{ζ} for service ζ can be evaluated by:

$$\begin{aligned} r_{\zeta} &= \frac{l_{\zeta}}{\bar{t}_{\zeta}} \\ &= l_{\zeta} \cdot \bar{\lambda}_{\zeta} \end{aligned} \tag{13}$$

From Figure 15(b), this can be understood considering that r_{ζ} is the *average slope* of the function $D_{\zeta}(t)$; it is clear that if the flow $D_{\zeta}(t)$ is served at a rate lower than r_{ζ}, the distance between $D_{\zeta}(t)$ and the dotted line (offered load) will increase indefinitely for $t \to +\infty$ (see "waiting time" in Figure 15(a)). Thus, this distance remains bounded only if the serving rate is higher or equal to r_{ζ}. In this case every message is being served the moment it enters the bucket; sometimes a message has to wait for some time in case the server is processing a previous job. This small waiting time is due to the fact that r_{ζ} is derived based on the average inter-arrival time, therefore, it represents a lower bound on the serving rate for a single service ζ.



(a) $D_\zeta(t)$ with low serving rate r_ζ; G_ζ is *WXGRAPH*-RT

(b) $D_\zeta(t)$ with the minimum required serving rate r_ζ; G_ζ is *WXGRAPH*-RT

Fig. 15. Different serving rates for the $D_\zeta(t)$ of *WXGRAPH*-RT

The estimation of the minimum required rate for an aggregation G of traffic sources can be similarly analyzed. If the total number of arrivals of the G service in the interval $[0, t]$ is denoted by $\alpha(t) \triangleq \sum_\zeta \alpha_\zeta(t)$, then from (11), the total arrival rate is:

$$\begin{aligned} \lambda(t) &\triangleq \frac{\alpha(t)}{t} \\ &= \frac{\sum_\zeta \alpha_\zeta(t)}{t} \\ &= \sum_\zeta \lambda_\zeta(t) \end{aligned} \tag{14}$$

in which, if we assume that for every $\zeta \in \Phi$ all limits as in (12) exist, as t approaches $+\infty$, we obtain the average of the aggregate arrival rate:

$$\begin{aligned} \bar{\lambda} &= \lim_{t\to+\infty} \lambda(t) \\ &= \lim_{t\to+\infty} \sum_\zeta \lambda_\zeta(t) \\ &= \sum_\zeta \lim_{t\to+\infty} \lambda_\zeta(t) \\ &= \sum_\zeta \bar{\lambda}_\zeta(t) \end{aligned} \tag{15}$$

Now, the ratio of the number of arrivals of service ζ to the total arrivals can be deduced from:

$$\frac{\alpha_\zeta(t)}{\alpha(t)} = \frac{\lambda_\zeta(t)}{\lambda(t)}$$

which as $t \to +\infty$ converges to

$$= \frac{\bar{\lambda}_\zeta}{\bar{\lambda}} \tag{16}$$

It follows that we can denote the average message size $\bar{l}$ for the aggregate process G as:

$$\bar{l} = \sum_{\zeta} \frac{\bar{\lambda}_{\zeta}}{\bar{\lambda}} \cdot l_{\zeta} \tag{17}$$

which is basically dependent on the average arrival rate of every service.
Finally, the overall minimum required serving rate r for the aggregation of all the services $\zeta \in \Phi$ is:

$$\begin{aligned} r &= \bar{\lambda} \cdot \bar{l} \\ &= \bar{\lambda} \cdot \sum_{\zeta} \frac{\bar{\lambda}_{\zeta}}{\bar{\lambda}} \cdot l_{\zeta} \\ &= \sum_{\zeta} \bar{\lambda}_{\zeta} \cdot l_{\zeta} \\ &= \sum_{\zeta} r_{\zeta} \end{aligned} \tag{18}$$

which is simply the aggregate serving rate of all services $\zeta \in \Phi$.

6.2.2 Buffer size: B

The buffer size B plays a vital role in aeronautical communications. It determines, besides the serving rate, the maximum waiting time that can be introduced by a queue also the probability to drop packets because of an overflow, i.e. the dropping probability. If B is very large, then a message/packet will experience a long waiting time that will affect the latency requirements of the service but a low dropping probability. Conversely, a small B will result in packets/messages being dropped whenever the buffer is full; thus, a higher dropping probability, but lower waiting time.
Also, starting the evaluation for a single process served at rate r_{ζ}, the waiting time of a packet, arriving at time t, in the queue is:

$$T_{W_{\zeta}}(t) = \frac{b_{\zeta}(t)}{r_{\zeta}} \tag{19}$$

where $b_{\zeta}(t)$ is the queue size at time t, i.e. $D_{\zeta}(t) - r_{\zeta} \cdot t$, see Figure 15(a).
Assuming that the limit as t approaches $+\infty$ exists for every service $\zeta \in \Phi$, we can determine the expected waiting time per service message in the buffer as:

$$\bar{T}_{W_{\zeta}} = \lim_{t \to +\infty} \frac{1}{t} \int_0^t T_{W_{\zeta}}(x)\, \mathrm{d}x \tag{20}$$

which is independent of the used queueing model. Further, the average queue length for service ζ can be represented by:

$$\bar{b}_{\zeta} = \lim_{t \to +\infty} \frac{1}{t} \int_0^t b_{\zeta}(x)\, \mathrm{d}x \tag{21}$$

If the last two limits in (20) and (21) exist, then the Little's theorem from (Kleinrock, 1975) states that the average buffer length in a queueing system can be rewritten as a function of the average process arrival rate times the average waiting time:

$$\bar{b}_{\zeta} = \bar{\lambda}_{\zeta} \cdot \bar{T}_{W_{\zeta}} \tag{22}$$

Equation (22) describes an average queue size for the service ζ. For this single service, the queueing system could be modeled as $M/D/1$ where M refers to memoryless arrival rate and D stands for deterministic serving rate since the for a single service the message size l_ζ is constant, which implies a constant serving time per message.
On the other hand, the aggregation of multiple services into a single queue allows the usage of a model with general serving time distribution, i.e. $M/G/1$, for which the size of the messages vary accordingly.
The above mentioned models describe a queue with a length that can grow indefinitely. Therefore, to limit it with a certain buffer size B, the queueing model $M/G/1/B$ could be implemented. This assumes that the aggregate arrival rate of the services has Poisson distribution, as proposed in the (EUROCONTROL & FAA, 2007). Clearly, for a finite buffer size B, the queue cannot accommodate all the messages arriving from the services because of their stochastic property. This is illustrated in Figure 16. As it can be clearly seen that not all of the total arrivals (λ) are being buffered since the queue can reject arrivals when the buffer is full. Thus, arrivals noted by (λ_e) denotes the rate of entering the system when the buffer is not full whereas (λ_d) indicates the rate of not entering the system because of the unavailability of resources.

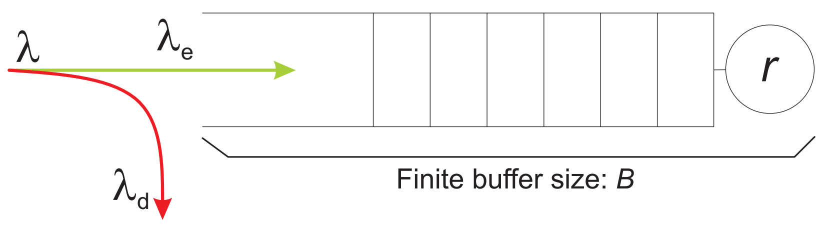


Fig. 16. Illustration of a queue with limited size B and dropping probability

The work in literature about queueing systems, such as in (Kleinrock, 1975) and (Smith, 2004), allows the derivation of the dropping probability as a function of arrival rate, buffer size and serving rate. This helps in designing a system that takes the latency requirements of the services into consideration and with minimum losses.

7. Conclusions

The requirements of a simple but reliable transport layer protocol for the safety critical ATM services like ATS and AOC have been discussed. The basic assumption that draws the design phase of this study was based on a priori recommendations. The new protocol to be designed has to address two goals; first, take out some algorithms of TCP that complicate its operation and add overhead, specifically congestion and flow control plus session initiation and shutdown procedures. Secondly, cope with the pattern of the aeronautical traffic, which is different from the Internet traffic like FTP, for example. Within this study, some drawbacks of using TCP for aeronautical services were highlighted and some protocol enhancements for RUDP to be an alternative solution to overcome these weakness points of TCP have been recommended. Furthermore, the system properties in terms of service rate and buffer sizes on both ESs were discussed.

8. Acknowledgments

The research leading to these results has been partially funded by the European Community's Seventh Framework Programme (FP7/2007-2013) under Grant Agreement nř 233679. The SANDRA project is a Large Scale Integrating Project for the FP7 Topic AAT.2008.4.4.2 (Integrated approach to network centric aircraft communications for global aircraft operations). The project has 31 partners and started on 1st October 2009.

9. References

Allman, M., Paxson, V. & Blanton, E. (2009). TCP Congestion Control. IETF RFC 5681.

Boudec, J.-Y. L. & Thiran, P. (2001). *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*, Springer-Verlag, Berlin, Heidelberg.

Bova, T. & Krivoruchka, T. (1999). Reliable User Datagram Protocol. IETF Internet-draft.

Brakmo, L. & Peterson, L. (1995). TCP Vegas: End to End Congestion Avoidance on a Global Internet, *IEEE Journal on selected Areas in communications* 13(8): 1465–1480.

Brandes, S., Epple, U., Gligorevic, S., Schnell, M., Haindl, B. & Sajatovic, M. (2009). Physical Layer Specification of the L-band Digital Aeronautical Communications System (L-DACS1), *Integrated Communications, Navigation and Surveillance Conference, 2009. ICNS '09.*, Arlington, Virginia, United States, pp. 1 –12.

Daniel, L., Järvinen, I. & Kojo, M. (2008). Combating Packet Reordering in Vertical Handoff Using Cross-Layer Notifications to TCP, *Proceedings of the 2008 IEEE International Conference on Wireless & Mobile Computing, Networking & Communication*, IEEE Computer Society, Washington, DC, USA, pp. 297–303.

Daniel, L. & Kojo, M. (2008). Employing Cross-Layer Assisted TCP Algorithms to Improve TCP Performance with Vertical Handoffs, *International Journal of Communication Networks and Distributed Systems* 1(4/5/6): 433–465.

Ehammer, M., Graupl, T. & Rokitansky, C.-H. (2009). TCP/IP over Aeronautical Communication Systems - Effects on Bandwidth Consumption, *IEEE/AIAA 28th Digital Avionics Systems Conference, 2009. DASC '09.*, FL, USA, pp. 4.A.3–1 –4.A.3–9.

Ehammer, M., Graupl, T., Rokitansky, C.-H. & Kissling, C. (2009). The Operation of TCP over Aeronautical Networks, *Integrated Communications, Navigation and Surveillance Conference, 2009. ICNS '09.*, Arlington, Virginia, United States, pp. 1 –11.

ESA (2009). ARTES 10 - Iris Programme.
URL: *http://telecom.esa.int/iris*

EUROCONTROL & FAA (2007). Communications Operating Concept and Requirements for the Future Radio System, COCR v2.0.

Graupl, T., Ehammer, M. & Rokitansky, C.-H. (2009). L-DACS 1 Data Link Layer Design and Performance, *Integrated Communications, Navigation and Surveillance Conference, 2009. ICNS '09.*, Arlington, Virginia, United States, pp. 1 –12.

ICAO (2010). Manual for the ATN using IPS Standards and Protocols.

ITU-T (1995). Protocol for Providing the Connection-mode Transport Service, *Recommendation X.224*, International Telecommunication Union, Geneva.

Jacobson, V. (1988). Congestion Avoidance and Control, *SIGCOMM Comput. Commun. Rev.* 18(4): 314–329.

Kissling, C. (2009). Data Link Selection in Mobile Aeronautical Telecommunication Networks, *The 27th AIAA International Communication Satellite Systems Conference, ICSSC '09*, Heriot-Watt University, UK.

Kissling, C. & Baudoin, C. (2008). Protocol Stack Options in Heterogeneous Aeronautical Networks, *4th Advanced Satellite Mobile Systems, 2008. ASMS 2008*, Bologna, Italy, pp. 43 –48.

Kissling, C. & Graupl, T. (2008). Options for Using Transport Layer Protocols.

Kleinrock, L. (1975). *Queueing Systems Volume 1: Theory*, Wiley-Interscience.

Mascolo, S., Casetti, C., Gerla, M., Sanadidi, M. Y. & Wang, R. (2001). TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links, *Proceedings of the 7th annual international conference on Mobile computing and networking*, MobiCom '01, ACM, New York, NY, USA, pp. 287–297.

Muhammad, M. & Berioli, M. (2010). Overhead Estimation of Selected Protocols for File Transfer, *The 5th Advanced satellite multimedia systems conference (asms)*, Cagliari, Italy, pp. 449 –454.

NEWSKY (2009). D12 Efficient Resource Management Techniques, Design, Assessment and Selection.

Partridge, C. & Hinden, R. (1990). Version 2 of the Reliable Data Protocol RDP. IETF RFC 1151.

Postel, J. (1980). User Datagram Protocol. IETF RFC 768.

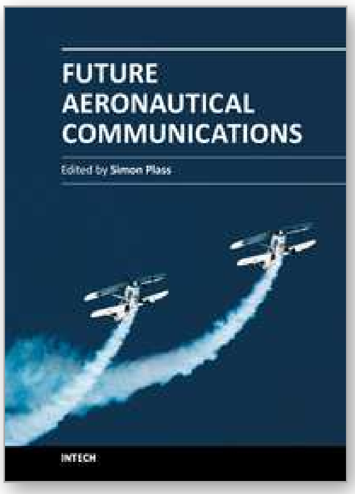
Postel, J. (1981). Transmission Control Protocol. IETF RFC 793.

Rokitansky, C.-H., Ehammer, M. & Graupl, T. (2008). Communication Capacity Assesment for the Iris Satellite System, *IEEE/AIAA 27th Digital Avionics Systems Conference, 2008. DASC 2008*, FL, USA, pp. 2.B.3–1 –2.B.3–13.

SANDRA (2011). Seamless Aeronautical Networking through integration of Data links Radios and Antennas.
URL: *http://www.sandra.aero*

Velten, D., Hinden, R. & Sax, J. (1984). Reliable Data Protocol. IETF RFC 908.

Smith, J. M. (2004). Optimal Design and Performance Modelling of M/G/1/K Queueing Systems, *International Journal of Mathematical and Computer Modelling* 39(9/10): 1049–1081.



Future Aeronautical Communications
Edited by Dr. Simon Plass

ISBN 978-953-307-625-6
Hard cover, 378 pages
Publisher InTech
Published online 26, September, 2011
Published in print edition September, 2011

There are well-founded concerns that current air transportation systems will not be able to cope with their expected growth. Current processes, procedures and technologies in aeronautical communications do not provide the flexibility needed to meet the growing demands. Aeronautical communications is seen as a major bottleneck stressing capacity limits in air transportation. Ongoing research projects are developing the fundamental methods, concepts and technologies for future aeronautical communications that are required to enable higher capacities in air transportation. The aim of this book is to edit the ensemble of newest contributions and research results in the field of future aeronautical communications. The book gives the readers the opportunity to deepen and broaden their knowledge of this field. Todayâ€™s and tomorrowâ€™s problems / methods in the field of aeronautical communications are treated: current trends are identified; IPv6 aeronautical network aspect are covered; challenges for the satellite component are illustrated; AeroMACS and LDACS as future data links are investigated and visions for aeronautical communications are formulated.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821