

Flow Control and Active Queue Management for Integrated Services in an Aeronautical Satellite Network

Yadong Shang^{*}, Michael Hadjitheodosiou[†] and John Baras[‡]
*HyNet Center, Institute for Systems Research,
University of Maryland, College Park, MD 20742, USA*
E-mail: {shangyd, michalis, baras}@umd.edu

The IP-based broadband aeronautical satellite network will provide numerous new applications and services for both airspace system operations and passenger communications. Application of commercial off the shelf technologies and techniques has the potential to make network operations economically and technically realizable. However the performance of data communication protocols and applications over aeronautical satellite networks is degraded dramatically. For the satellite channel, the window based flow control algorithm may cause transient congestion. The flow control mechanisms with or without explicit feedback from the congested resource are limited in performance and not effective. We propose a hierarchical scheduler for satellite gateways. Different service disciplines are used for integrated services. A Random Early Detection Flow Control (REDFC) algorithm is proposed for best-effort traffic at the satellite gateways. In REDFC, each packet is transmitted with a probability that is a function of the queue length. If the packet is not transmitted successfully, it will be retransmitted after certain time interval. We also propose a virtual parallel queue structure as a new active queue management scheme for short life and long life web traffic. The new flow control and active queue management schemes will provide better performance for integrated services in aeronautical satellite networks.

I. Introduction

THE world's aviation industry is soaring into the 21st Century with projected increases in business, recreation, and personal travel. The current airspace systems are quickly becoming overburdened by increases in air traffic coupled with the use of old technologies and legacy systems [1]. New Internet infrastructure and technologies capable of providing high-speed and high-quality services are needed to accommodate multimedia applications with diverse quality of service (QoS). A Satellite communication system, distinguished by its global coverage, inherent broadcast capability, bandwidth-on-demand flexibility, suitability to free flight concepts, and the ability to support mobility, is an excellent candidate to provide broadband integrated services for aeronautical communications [2].

Future aeronautical satellite systems will offer Internet connections at up to broadband (tens of Mbps) data rates via networks of GEO or LEO satellites. Figure 1 illustrates the IP-based network topology of aeronautical satellite networks. This system will be composed of three major segments: cabin segment with on-board networks, space segment for interconnection of the cabin with the terrestrial networks, ground segment which provides the interconnection to the terrestrial personal and data networks as well as the Internet backbone. In this paper, we focus on the GEO satellites because of their stationary, relative to earth, large coverage, and significant reduction in system complexity comparing to LEO satellite systems. The GEO satellites in the present work are bent pipe satellites, which are simply signal repeaters in the sky. They are physical layer devices and no switching is performed on board.

The aeronautical satellite networks will provide numerous new applications for both airspace system user operations and secure air traffic management. Application of commercial off the shelf technologies and techniques

^{*} Research Assistant, HyNet, University of Maryland, College Park, MD 20770.

[†] Assistant Research Scientist, HyNet, Institute for Systems Research, University of Maryland, AIAA Member.

[‡] Professor, HyNet, Institute for Systems Research and ECE Department, University of Maryland, AIAA Member.

has the potential to make network operations economically and technically realizable. However, the performance of data communications protocols and applications over aeronautical satellite systems is the subject of heated debate in the research community.

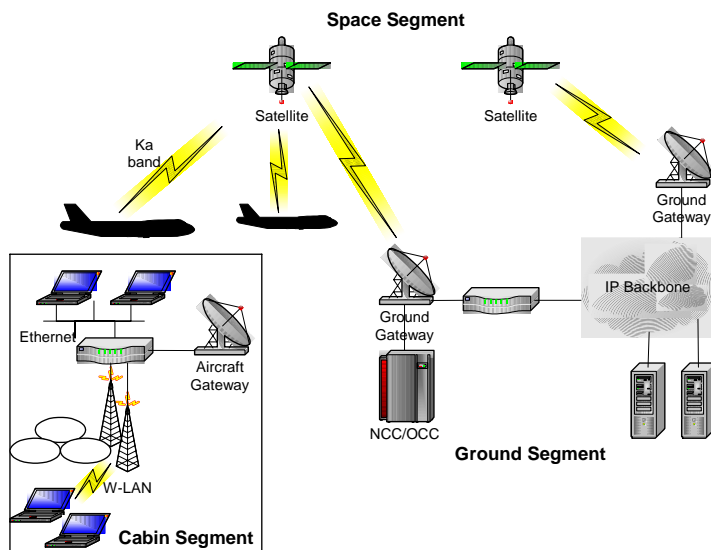


Figure 1: Aeronautical Satellite Networks

In aeronautical satellite networks, we are expected to provide diverse QoS guarantees (e.g., on delay, delay jitter, packet loss rate, throughput) for a wide range of traffic classes. There are a large number of architectural decisions that affect the design of a traffic management scheme. An effective traffic management and congestion control system requires several specific methods, each acting on a different time scale. A combination of congestion control mechanisms working at the data link, networking, and transport layers are required, along with proper capacity planning to overcome congestions lasting from short durations to very long durations. In this work, we propose a hierarchical scheduler at the satellite gateways for traffic management, components of which are designed for different traffic classes. A random early detection flow control (REDFC) scheme for web traffic is proposed and will be implemented and evaluated.

II. Problems of Traffic Management over Satellite Channels

The types of applications, which must be supported in aeronautical communications, can be divided into two categories: *safety communications* and *non-safety communications*. The aeronautical satellite communication system is normally used for communications related to the safety and efficiency of flight, but non-safety communications could be permitted on a non-interference basis, when priority and preemption can guarantee the precedence of safety communications [3]. These two application categories, safety and non-safety communications, include a range of particular communication services. **Table 1** assigns to each application category respective key services. Some services fit into more than one category. Moreover, not all services will be permanently required. In case of an emergency, for instance, the shutdown of passenger services for the benefit of flight security applications is acceptable. From a system design viewpoint, this immediately relaxes the worst-case data rate demand of aircraft communication system.

Table 1. Categorized Services

Category	Services
Safety Communication	ATC, Weather services, pilot reports, Cabin and cockpit surveillance video, flight recorder data, aircraft logistics and maintenance data
Non-safety communication	WWW, email, live TV, phone, fax, video-conferencing, file transfer, intelligent travel information, gambling

The next step is to derive individual traffic statistics for the identified service categories. The traffic generated and received by a single aircraft is a function of the distribution of passengers among first, business and economy class, the duration of the flight, the physiological flight time, and the set of available services. The traffic should be described as superposition of the traffic generated by each passenger according to the characteristics of the desired services in terms of data rate and QoS parameters. The system dimensioning of the satellite system is beyond the scope of this work. We will focus on the traffic management for one aircraft. However, the same scheme and solution can be extended to more complicated aeronautical satellite networks.

In previous work, we have designed a splitting based transport protocol, *AeroTCP*, for aeronautical satellite networks. The main idea is to use fixed window for flow control, static or adaptive congestion control, and super fast error control for satellite TCP connections. This algorithm can maintain high utilization of satellite channels and fairness for large file transfers. However, it still has some problems with respect to supporting other applications with bursty traffic, especially for integrated services with different QoS requirements.

The transport protocol in our scheme uses window flow control. The receiver window, $rwnd$, is used to make sure that the sender will not overflow the receiver. It is set to 2-4 times *peak rate delay product* (PRDP) to ensure high utilization of the satellite channel. The congestion window, $cwnd$, is used to make sure that the traffic load will not overflow the network. It is set to peak rate delay product (PRDP) or adaptively adjusted based on the number of active connections. There is the possibility that the aggregate rate of the input traffic temporarily exceeds the capacity of the network, in which cases packets may experience long delays or get dropped by the network. It is even worse for integrated services, while the bandwidth for best-effort services has lower priority than guaranteed services (**Figure 2**). In high-speed networks with connections with large *delay-bandwidth products*, gateways are likely to be designed with correspondingly large maximum queues to accommodate transient congestion. The large buffers are full most of the time at congestion. This would significantly increase the average delay in the network. The long delay may cause the upper layer TCP to timeout and retransmit the packet unnecessarily.

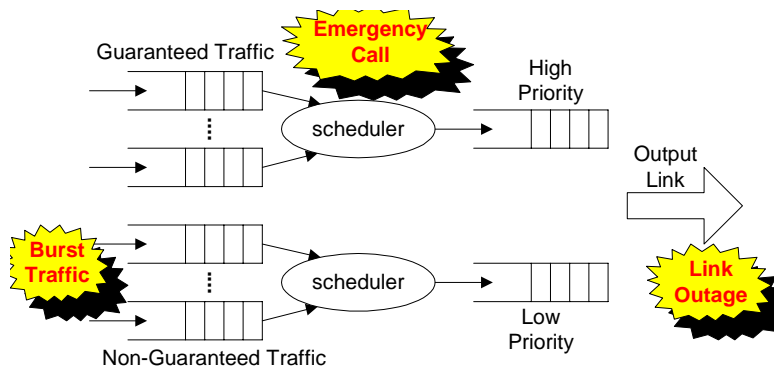


Figure 2: Serving both guaranteed and non-guaranteed traffic

The FIFO queueing scheme in the current Internet is also not efficient for supporting integrated services. This FIFO policy could cause three problems: 1) a bulk data transfer such as FTP download could have a lot of packets buffered in the FIFO queue at the satellite gateway, which increases the packet queuing delay of short files. 2) The FIFO queue cannot provide fairness in the presence of ill-behaved sources. A source, sending packets to a gateway at a sufficiently high speed, can capture an arbitrarily high fraction of the bandwidth of the outgoing bandwidth. 3) The situation is even worse for two-way transfers. When the users are sending data (say email with a large attachment or upload file using FTP) and browsing the web at the same time, a lot of data packets could be queued in front of ACKs in a FIFO queue, which increases the ACKs delay dramatically.

III. Related Work on Transient Congestion Control

Despite the fact that a number of schemes have been proposed for congestion control, the search for new schemes continues. There are two reasons for this. First, there are requirements for congestion control schemes that make it difficult to get a satisfactory solution. Second, there are several network policies that affect the design of congestion schemes. In this section, we will review several congestion control mechanisms at the transport layer and network layer for packet switch networks. The end-to-end adaptive window flow control schemes at transport layer

without explicit feedback from the gateway are limited by the timescales of the connection, difficulty of distinguishing the cause of packet loss, and thus are not considered here.

The most effective detection of congestion can occur in the gateway itself. The goal of monitoring the average queue size at the gateway, and of notifying connections of incipient congestion is to provide congestion avoidance by controlling the buffer space or average queue size, to avoid global synchronization and a bias against bursty traffic. Networks contain connections with a range of burstiness, and gateways such as drop tail and random drop (when a packet arrives at the gateway and the queue is full -- the gateway randomly chooses a packet from the gateway queue to drop) gateways have a bias against bursty traffic. With drop tail gateways, the more bursty the traffic from a particular connection, the more likely it is that the gateway queue will overflow when packets from that connection arrive at the gateway. Drop Tail gateways also introduce global synchronization. When the queue overflows, packets are often dropped from several connections and these connections decrease their windows at the same time. This results in a loss of throughput at the gateway.

Hashem [4] discusses some of the shortcomings of Random Drop and Drop Tail gateways, and briefly investigates Early Random Drop gateways. In the implementation of Early Random Drop gateways, if the queue length exceeds a certain drop level, then the gateway drops each packet arriving at the gateway with a fixed drop probability. Hashem points out that Early Random Drop gateways have a broader view of traffic distribution and reduce global synchronization. However, in future implementations the drop level and the drop probability should be adjusted dynamically, depending on network traffic.

The Random Early Detection (RED) [5] congestion control mechanisms monitor the average queue size for each output queue, and using randomization, choose connections to notify of that congestion. Transient congestion is accommodated by a temporary increase in the queue. Longer-lived congestion is reflected by an increase in the computed average queue size, which results in randomized feedback to some of the connections to decrease their windows. In order to avoid problems such as biases against bursty traffic and global synchronization, the RED gateway uses randomization in choosing which arriving packets to mark. With this method, the probability of marking a packet from a particular connection is roughly proportional to that connection's share of the bandwidth through the gateway. However, the RED gateway marks and notifies the source to reduce the window for that connection by dropping a packet, setting a bit in a packet header, or some other method understood by the transport protocol. The current feedback mechanism in TCP/IP networks is for the gateway to drop packets. Packet drop or buffer overflow will dramatically affect the performance of large bandwidth delay product networks.

Flow random early detection (FRED) [6] was motivated by the unfair bandwidth sharing of RED gateways. This is because RED imposes the same loss rate on all the flows regardless of their bandwidth. FRED maintains per-active-flow accounting and the drop probability is proportional to the flow's buffer usage. FRED provides greater fairness, flow isolation and flow protection than RED.

The DECbit congestion avoidance scheme [7] is a binary feedback scheme for congestion avoidance. The DECbit gateways use a congestion-indication bit in the packet header to provide feedback about the congestion in the network. When a packet arrives at the gateway, the gateway calculates the average queue length for the last (busy+idle) period plus the current busy period. When the average queue length exceeds one, then the gateway sets the congestion-indication bit in the packet header of arriving packets. The source uses window flow control, and updates its window once every two round-trip times. If at least half of the packets in the last window had the congestion indication bit set, then the window is decreased exponentially. Otherwise, the window is increased linearly. However, the average queue size of the DECbit scheme can sometimes be averaged over a fairly short period of time, resulting in failure to control the average queue size in high-speed networks. DECbit networks can exhibit a bias against bursty traffic. The adaptive window schemes where the source nodes increase or decrease their windows according to feedback depending on the queue lengths at the gateways, work only if the congestion and the connection last for a few round-trip times. They are not suited for short connections (like web browsers) that finish transmission in one round-trip time.

Queueing algorithms do not affect congestion directly. They determine the way in which packets from different sources interact with each other, which, in turn, affects the collective behavior of flow control algorithms. This effect makes queueing schemes a crucial component in effective congestion control. If a separate queue is maintained for each source-destination pair, fairness (max-min fairness) can be obtained by serving the queues in a

round-robin order [8]. This will cause the queues with large packets to get a larger share of the bandwidth than those with small packets. Schemes to tackle this inequity have also been proposed in [9]. Fair Queueing in [9] provides several advantages of fair bandwidth allocation, lower delay for sources using less than their full share of bandwidth, and protection from ill-behaved sources. However, all those queueing algorithms assume that the output line of the scheduler has constant bandwidth. Thus they are not suited for integrated networks where bandwidth available for best-effort traffic is affected by guaranteed traffic. In [10], a round-robin scheduler with the backpressure mechanism has been proposed, where the scheduler stops sending packets if the buffer at the output link is full. This scheme acts as a Drop Tail gateway and leads to global synchronization and bias against bursty sources.

In [11], a hop-by-hop mechanism for controlling congestion and flow control has been proposed in a packet switched network. Simulation results show that the scheme displays better dynamical behavior than that of congestion control schemes based on sliding windows, such as that of TCP. The scheme requires that each packet switch generate information about buffer occupancy and current service rates per traffic stream. However, the cost of the additional information required is high. It is also difficult to implement in a heterogeneous environment.

IV. Analysis and Proposed Solution

In aeronautical satellite networks, there are multiple classes of traffic on the gateway with many widely different QoS objectives. A *hybrid traffic management architecture*, components of which optimize different parameters like maximum delay, delay variation and fairness of distribution of bandwidth is thus needed. A higher layer arbitration policy to assign resources to these modules is also required. For best-effort traffic, the flow control and congestion control mechanisms of AeroTCP still have some problems when supporting integrated services with a wide range of burstiness. An additional flow control mechanism is needed to accommodate transient congestion at the gateway. In this section, we first define the traffic classes and their requirements for aeronautical satellite networks. A hierarchical priority-based scheduling scheme is proposed, along with a Random Early Detection Flow Control mechanism to provide congestion avoidance for web traffic.

A. Hierarchical Scheduler

All applications and services are categorized into five classes of service: Constant Bit Rate (CBR) for applications like voice where the packet arrival rate is fairly constant over time and there are strict QoS requirement, Real-time Variable Bit Rate (rt-VBR) for applications like real time video, where the packet arrival rate is bursty due to the nature of the video codes and the QoS requirements are strict, Non Real-time Variable Bit Rate (nrt-VBR) for off-line video content (music, movie) where the traffic rate is bursty but the QoS requirements on the delay are loose. High Priority Unspecified Bit Rate (HPUBR) for some critical applications and interactive Internet applications where traffic load is light but they are delay sensitive. Unspecified Bit Rate (UBR) for all other Internet applications like web browser and FTP where traffic is bursty and throughput sensitive. We note that not all applications are presented in both directions from and to the aircraft. For example, surveillance video is only from aircraft to ground, while FTP and web browser traffic are mainly from ground to aircraft (bulk ACKs on return channel). Also note that the definition is not rigid. Other classifications and prioritizations could also be defined based on the applications and their QoS requirements.

The central part at the aircraft gateway and the ground gateway is a service integrator, which will provide interfaces for the wireless and wired service access points in the cabin, as well as the interface to the terrestrial networks at the ground. All services will be bundled and transported between a pair of service integrators, which provide a QoS monitoring mechanism for each packet stream. The major part of the service integrator is a hierarchical scheduler (**Figure 3**). For each output link into the satellite network, packets received from different access segments are segregated into different queues based on the type of services as indicated by the packet header. Traffic can be received from different networks such as UMTS, Bluetooth, and wired/wireless LAN connected to an IP backbone. Different service disciplines are used for three guaranteed performance services: delay-bounded scheduler for max delay sensitive traffic, jitter-bounded scheduler for jitter sensitive traffic, and rate-based scheduler for nrt-VBR traffic. A priority-based top-level scheduler serves lower level schedulers for the different traffic classes.

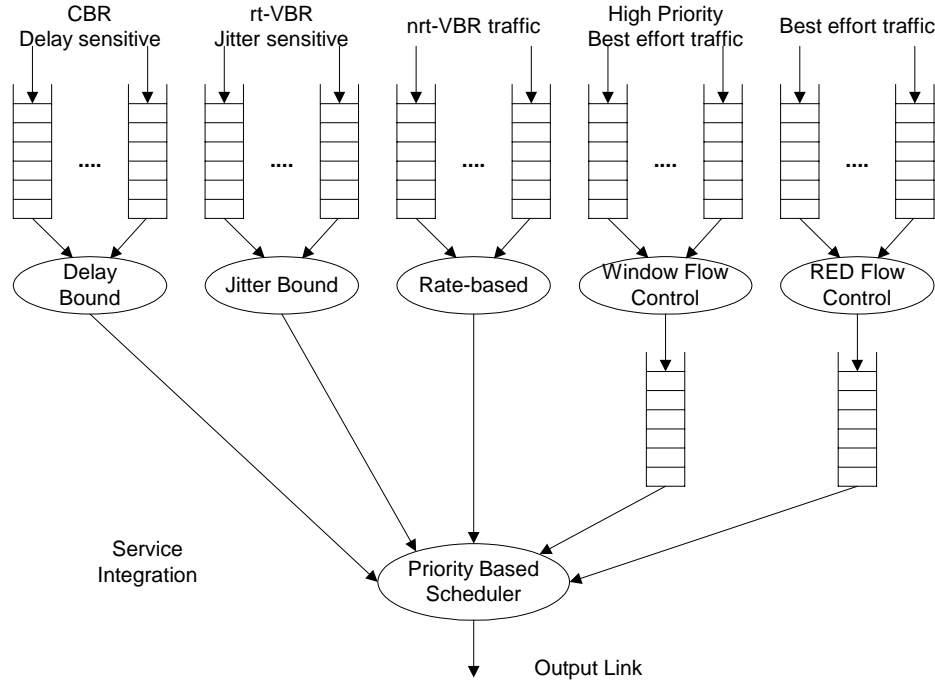


Figure 3: the hierarchical scheduler at the service integrator

The lower level schedulers send packets to the top-level scheduler to contend for service. Best-effort packets are transmitted only when no packets from the guaranteed service queue are available for transmission. The high priority best effort traffic queue is mainly for interactive applications (telnet, DNS lookup) and aircraft applications (flight data, pilot report) that are delay sensitive, while the low priority best traffic queue is mainly for Internet applications (FTP, HTTP) that are throughput sensitive. The preemptive priority algorithm is more tractable analytically, however the non-preemptive version is easier to implement. Fixed window flow control is used for high priority best-effort traffic since its total traffic load is light compared to web traffic and guaranteed services. The flow control algorithm for best-effort traffic is discussed in the next section.

B. Random Early Detection Flow Control (REDFC)

We propose a Random Early Detection Flow Control (REDFC) algorithm for best-effort traffic at the gateways. In the previous section, we designed a splitting based transport protocol with static or adaptive window flow control. This mechanism is not efficient for avoiding transient congestion at the gateway. The gateway can provide congestion avoidance by dropping the packets based on the average queue size. However, this may cause transport layer protocol timeouts or frozen window, resulting in loss of throughput. The window adjustment scheme without explicit feedback from the gateway cannot respond quickly and accurately to transient congestion. Since the congestion feedback is available at the gateway, the transport protocol could use this information to adjust its window size. It is more efficient if the transport protocol could use this information to control the packet transmission rate. This is the main idea of the Random Early Detection Flow Control (REDFC).

For each packet allowed for transmission (we still use window flow control to make sure the sender will not overflow the receiver and congestion control for fair sharing), the REDFC gateway calculates the average queue size, using a low-pass filter with an exponential weighted moving average (EWMA). The average queue size is compared to two thresholds, a minimum threshold and a maximum threshold. When the average size is less than the minimum threshold, the packet is transmitted without delay. When the average queue size is greater than the maximum threshold, no packets are allowed to transmit. When the average queue size is between the minimum and the maximum thresholds, each packet is transmitted with probability p_a , where p_a is a function of the average queue size. Each time a packet is transmitted, the probability that a packet is transmitted from a particular connection is roughly proportional to that connection's share of bandwidth at the gateway. When this packet is not successfully

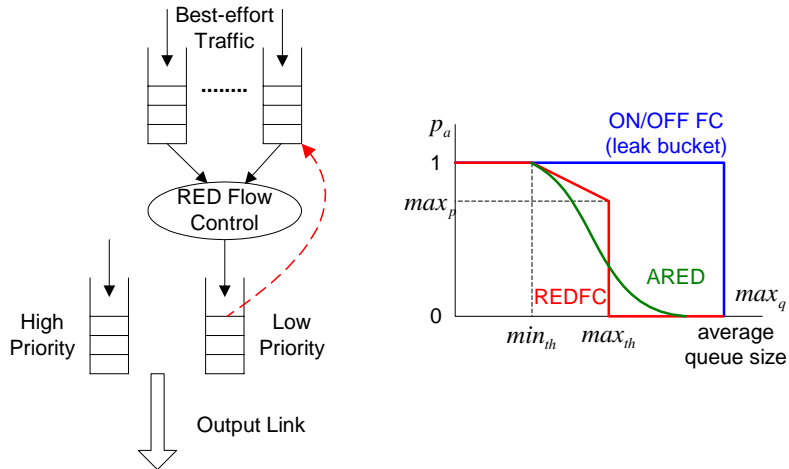


Figure 5: Random Early Detection Flow Control

We will study the setting of the various parameters for the REDFC algorithm. The queue weight w_q is determined by the size and duration of bursts in the queue size allowed at the gateway. The minimum and maximum thresholds min_{th} and max_{th} are determined by the desired average queue size. It is difficult to determine the optimum average queue size for maximizing throughput and minimizing delay for various network configurations. This is heavily dependent on the characterization of the network traffic as well as on the physical characteristics of the network.

C. Active Queue Management for Best Effort Traffic

In REDFC, each packet is transmitted with a probability that is a function of the queue length. If the packet is not transmitted successfully, it will be retransmitted after certain time interval. However, a great portion of Internet traffic is short-life web and UDP traffic. Most web traffic has a small file size. Dropping short-life TCP and UDP packets is not very effective in alleviating the congestion level at the bottleneck router. From the viewpoint of TCP, one or several packet losses lead to extra delay for retransmission and even cause TCP timeout. This delay severely degrades the performance of delivering short messages such as web pages and web browsers; they experience a long waiting time even with a high-speed network. In addition, the performance of adaptive RED is severely degraded by these short but bursty connections.

We propose a virtual parallel queue structure as a new active queue management scheme (Figure 6). The idea is to separate the long-life and short-life traffic into two different virtual queues. The first queue is to run the drop-tail policy and work for the short-life TCP and UDP packets. In order to have a small mean delay, the service rate of this drop-tail is dynamically determined by its virtual queue length. The remaining long-life traffic is directed to a virtual queue and is transmitted by the REDFC algorithm. Both virtual queues share the same physical buffer memory [12].

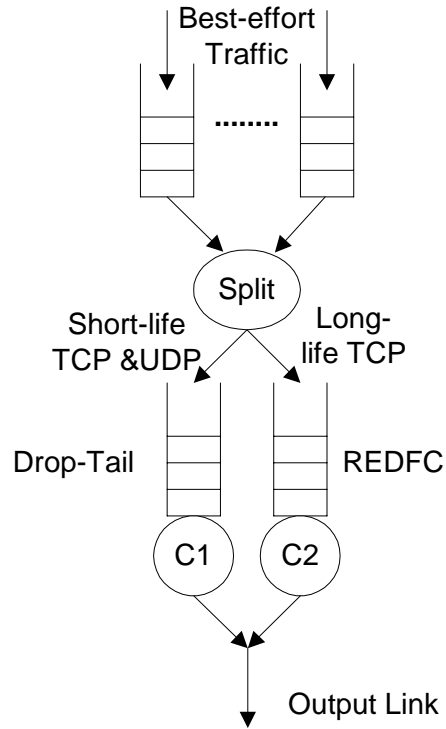


Figure 6: Parallel queue structure

The first problem is how to split the long-life traffic from other short-life traffic at the gateway. To this end, the router

has to know the age or elapsed time of each TCP connection. Since we use TCP splitting protocol, this information is known at the satellite gateway. The simple way to split the traffic is based on TCP header information. All web traffic and UDP are considered as short-life traffic, while FTP application traffic is considered as long-life traffic.

We can also estimate the elapsed time by using the following approach:

- When a packet arrives from a new source-destination pair, which is not seen by the router in the past T seconds, we treat it as a new TCP connection and identify this connection as a short-life connection.
- Send the new connection packets to the drop-tail queue.
- Set a counter for the number of packets from this connection.
- If the cumulative packets number is greater than a threshold N, then we believe that the file size is large enough. We redirect the subsequent packets of this connection to the REDFC queue.
- Remove the counter if there is no packet arrival in the last T seconds.

Figure 7: Traffic Split algorithm

The second problem is to determine the service rate for each queue. In order to have a small mean delay for short-life traffic, the service rate of the drop-tail queue is dynamically determined by its virtual queue length. We define the maximal threshold $maxth_i$ and minimal threshold $minth_i$ for $i=1, 2$. The service rates $C_1(t)$ and $C_2(t)$ are given by the following algorithm:

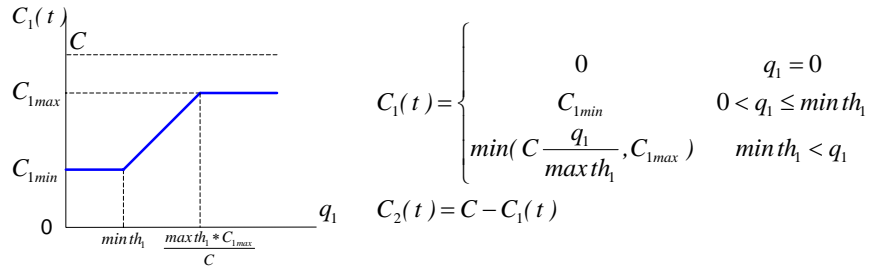


Figure 8: Service rate for virtual parallel queue

V. Conclusions

At the satellite gateway for aeronautical networks, we want to provide QoS guarantee for a wide range of applications and services. The new traffic management scheme will provide congestion control in all time scales for integrated services. For web services with bursty traffic, a new random early detection flow control algorithm is proposed and will be evaluated and analyzed. This scheme should maintain the average queue size, minimize the packet drop rate, and achieve fairness for the gateways. It also avoids the bias against bursty traffic and global synchronization.

The parallel queue structure also provides more degrees of freedom to control the router by considering different bandwidth allocation policies and dynamic thresholds for REDFC. Here, the bandwidth allocation policy is a simple function of the current virtual queue length. However, it is well known that web traffic is strongly correlated and has a long-range dependency property. Based on observations of the recent past traffic, the future bandwidth demand of web traffic was shown to be predictable. In future work, we will consider the optimal bandwidth allocation policy based on the prediction of congestion levels.

Acknowledgments

This work was supported by the Maryland Hybrid Networks Center (HyNet), under NASA cooperative agreements NCC8-235 and NAG3-2844. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the National Aeronautics and Space Administration.

References

- ¹Blueprint for NAS Modernization 2002 Update, <http://www.faa.gov/nasarchitecture/Blueprint2002.htm>
- ²Y. Hu, V. O.K. Li, "Satellite-Based Internet: A Tutorial", IEEE Communications Magazine, Mar. 2001, pp. 154-162.
- ³O. Ercetin, M. O. Ball, L. Tassiulas, "Next Generation Satellite Systems for Aeronautical Communications", Technical Research Report, National Center of Excellence in Aviation Operations Research, NEXTOR T.R. 2000-1, ISR T.R. 2000-20.
- ⁴E. Hashem, "Analysis of Random Drop for Gateway Congestion Control", Report LCS TR-465, Laboratory for Computer Science, MIT, Cambridge, MA, 1989, pp. 103.
- ⁵S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, Volume 1, Issue 4, Aug. 1993, pp. 397-413.
- ⁶D. Lin and R. Morris, "Dynamics of Random Early Detection", In ACM SIGCOMM'97, 1997, pp. 127-137.
- ⁷K. K. Ramakrishnan and R. Jain, "A Binary Feedback Scheme for Congestion Avoidance in Computer Networks", ACM Transactions on Computer Systems, V.8, N.2, 1990, pp. 158-181.
- ⁸L. Hahne and R. G. Gallager, "Round Robin Scheduling for Fair Flow Control in Data Communications Networks", Proc. IEEE International Conference on Communications ICC'86, Toronto, Canada, June 22-25, 1986, pp4.3.1-5.
- ⁹A. Demers, S. Keshav, S. Shenker, "Analysis and Simulation of a Fair Queueing Algorithm", In Journal of Internetworking Research and Experience, pages 3-26, October 1990. Also in Proceedings of ACM SIGCOMM'89, pp. 1-12.
- ¹⁰X. Zhou, "Congestion Control and Medium Access Control in Satellite Data Networks", Ph.D. Thesis, University of Maryland, ISR Technical Report TH2005-2.
- ¹¹P. Mishra and H. Kanakia, "A Hop by Hop Rate-based Congestion Control Scheme", Proceedings of SIGCOMM'92, August 1992, p. 112-123.
- ¹²J-S. Jou, X. Tan, J. S. Baras, "A Parallel Queue Structure for Active Queue Management", Proceedings of the 38th Annual Conference on Information Sciences and Systems CISS 2004, pp. 467-472, Princeton, NJ, March 17-19, 2004; also ISR Technical Report TR2003-9.