TCP driven CAC scheme for HAPS and satellite integrated scenario

Georgios Theodoridis [(1)], Cesare Roseti [(2)], Niovi Pavlidou [(1)], Michele Luglio [(2)],

(1) Aristotle University of Thessaloniki, Department of Electrical & Computing Engineering,

P.O. Box 54124, Thessaloniki, Greece

(2) Dipartimento di Ingegneria Elettonica, University of Rome "Tor Vergata"

Via del Politecnico 1, 00133, Rome, Italy.

***Abstract* – Unfortunately, TCP-based applications suffer both from long propagation delay and link errors introduced by the satellite and the HAP segment accordingly, leading to poor end-to-end performance. Hence, based on the fact that many of the applications that are envisioned to be provided through HAPs run TCP as transport protocol, this article presents an innovating CAC scheme, which shall not take into account only the QoS requirements and the system capacity, but also the end-to-end TCP performance as well. For an LMDS network architecture, where the services are forwarded to the end-users through HAP located base stations, we study the effectiveness of the TCP-based CAC algorithm in upgrading the efficiency of the bandwidth utilization, for the downlink channel.**

Index Terms – CAC, HAP, satellite, TCP

I. Introduction

To provide telecommunication services, when terrestrial infrastructures deployment might fail or not exist, a scenario with HAPS and satellite integrated can represent a very profitable solution. In fact, the use of unmanned vehicles allows relatively short deployment time, while a geostationary satellite guarantees a very large coverage. Furthermore, such a communication platform can provide a broadband and cost-efficient Internet access in a vast gamut of cases:

1) Provision of LMDS
2) Exceptional events (i.e. Olympic games, political meetings, etc,)
3) Emergency situations (i.e. earthquakes, terrorist attack, chemical disasters, wars, etc.)
4) Temporally overload of the terrestrial networks (i.e. support capacity to high traffic areas).

In particular, this paper focus on a communication scenario where many users are connected to the HAPS station that, in turn, is connected with a remote satellite Hub station. In this context, a very important issue is to guarantee the QoS requested from each user. To this purpose, some Call Admission Control (CAC) schemes can be implemented. On the other hand, since the achieved throughput of a data source running TCP depends on both the sender-perceived RTT and the transmission error rate [1][2][3], a CAC schemes, taking into account only physical measurements, might lead to an inefficient and unfair utilization of the channel and poor end-to-end performance.

We propose an innovative TCP measurement-based CAC scheme running on top of the HAPs. The basic idea is that TCP measures periodically the average goodput for each active source and gives such a value as an input for the CAC algorithm (cross-layer interaction). In this way, the decision to accept a new connection shall be taken according to the TCP feedback.

II. TCP Basics

TCP is a transport protocol supporting most of the Internet applications (i.e., email, file transfers, web browsing, remote access, etc.). It provides to upper layers a connection-oriented, reliable and byte stream service [4]. The basic TCP task is to accept a data flow from an application and subdivide it into properly-sized chunks, called "segments" or "packets". Then, TCP associates a "sequence number" to each segment and sends it to the lower layers. On the receiving side, TCP reacts to a successfully reception of a given segment by sending a cumulative acknowledgement (ACK) to the sender and by delivering data to the application. Based on this scheme, three TCP function are implemented:

1) *Flow Control*: monitoring of the "in flight" data by a mechanism called "sliding window" [5];
2) *Congestion control*: adaptation of the transmission rate according to the network congestion state [6][7];
3) *Error control*: detection and recovery of the lost packets [1][8].

III. CAC Basics

CAC (Call Admission Control) is a process for managing the arriving traffic (at call, burst, or packet level) based on some predefined criteria. Namely, it is an algorithm that admits/rejects arriving users to optimize some objective function. In brief, using the traffic characteristics received from the user (traffic descriptors), the QoS objectives of both the candidate and the already active users and the network conditions, CAC judges whether a new connection can be accepted or not. As a result, due to the fact that CAC runs in order to prevent the occurrence of congestion, it is described as a preventive congestion control scheme (vs. reactive), aiming generally at achieving fair resource sharing, maximizing revenue, guaranteeing transmission rate and QoS.

IV. System Description

The reference scenario considers the study of the downlink channel of an integrated HAP-Satellite architecture. In specific, our scenario refers to the implementation of an LMDS network through the exploitation of the HAP infrastructure as the *last-mile* solution [9][10]. For the analysis of the downlink channel capacity, we take into account only the data streams generated by TCP-based applications originating in the core network. These data streams, in order to be forwarded to the end-users, follow the path through the Satellite Gateway to the GEO Satellite and from there they are broadcasted to all the HAPs situated under the Satellite coverage, in order to be distributed locally (Fig. 1).

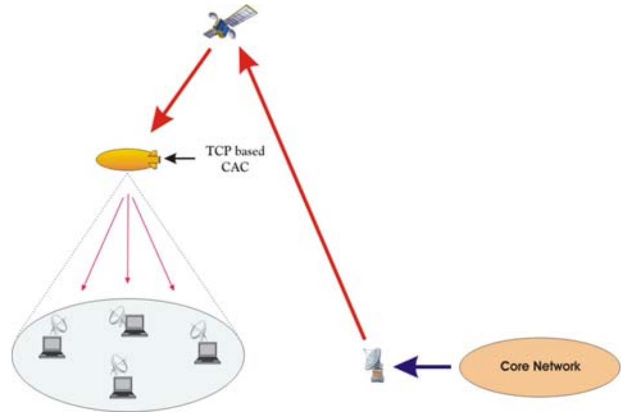

Figure 1: Network Architecture

As far as the system's technical parameters are concerned, the HAP is situated at an altitude of 18km, providing single-cell coverage to an 60km radius area, which corresponds to a minimum elevation angle of 20º. For this elevation angle, considering a system operating at the frequency band of 47/48 GHz, the available data rate for the downlink channel can be estimated ≥60Mbps, with 99% availability [9]. Nevertheless, the aforementioned data rate is shared among a great variety of applications, originating either locally by the HAP users or somewhere in the core network. Thus, for the performing the evaluation of our CAC algorithm, we consider that only 10Mbps (approximately equal to the capacity of the DVB-S standard) are granted for the provision of TCP-based services that are forwarded to the HAP through the Satellite backhaul link. Moreover, the nominal data rate of each connection is specified to 512Kbps, which however, as it has been described above, is further degraded due to the reception errors in conjunction with the implementation of TCP as the fourth layer protocol. Furthermore, multiple access is succeeded by the use of FD-TDMA techniques; according to the proposed scheme, the resultant time-slots/channels are no more dedicated to each user for the whole duration of one's connection, but they are dynamically assigned to all the active users, based on any of the existing scheduling algorithms (the description of these algorithms is beyond the scope of the present paper).

V. CAC-TCP interaction design

TCP implements a reactive congestion control based on a gradually probing of the network congestion state. In particular, the increasing of the transmission window is triggered by the ACKs reception. Furthermore, TCP interprets the missing reception of an ACK as a signal of network congestion, and then reduces its transmission window.

On the basis of this, two main considerations when evaluating the TCP performance on the considered scenario can be done:

1) The sender perceives a long RTT (~ 560 ms) due to satellite link. Then, TCP sender takes a long time to increase its transmission window;
2) Both wireless and satellite links can be affected by transmission errors. In the absence of feedback about the nature of the losses (corruption or congestion), TCP reduces its transmission window even though there is not congestion in the network.

Hence, TCP control loop may represent a strict constraint for the achievable data rate. Therefore, in several cases TCP connections are unable to fully exploit the resource assigned according to QoS requirements. Thus, in order to efficiently use the whole system capacity, without to further affect the QoS, TCP feedbacks may drive CAC decisions. Our main idea is to use samples of the TCP transmission rate per connection, taken at regular interval, as input for the CAC algorithm, in order to compute an estimate of the average achieved rate per connection, and then of the overall unused resource. Finally, a new connection will be accepted if its nominal data rate requirement is less than the unused resources.

On the contrary, according to the basic CAC algorithm against which our proposed scheme is compared, there is no monitoring of the real data rate of the connections and the used data rate is considered to be equal to the sum of the nominal data rates of the admitted connections. However, such a CAC scheme leads to a suboptimal utilization of the network resources. Being more specific, as it is stated above, due to the TCP functionality, the actual rate of the active data flows is usually lower in comparison with its nominal value. Thus, the CAC algorithm is misled to a rather "pessimistic" computation of the extent of the available bandwidth and rejects as overload flows that can actually be serviced properly.

Our proposed scheme can be realized by the implementation of a TCP proxy in the HAP, in parallel to the CAC, so as both to minimize the signaling overhead and to make the cross-layer interoperability viable.

VI. Simulation Description

In order to simulate the differences between the PER (Packet Error Rate) among the users scattered inside the coverage area, the coverage area has been divided in three concentric circular zones with different mean PER values respectively (Fig. 2). The chosen values correspond to the variations in the channel/reception conditions due to different propagation paths.

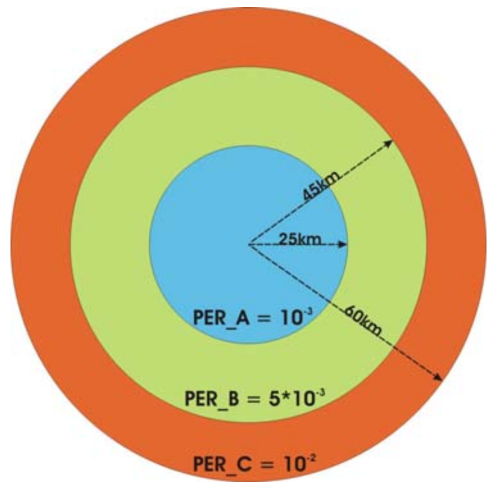

Figure 2: PER zones

Moreover, the arrivals of new admittance requests follow Poisson distribution, with mean interarrival time, *aver_int_time*, while the same stands for the duration of each connection. As a matter of fact, the mean duration of the connections, denoted as *aver_con_dur*, is specified equal to 500secs, which (for 512Kbps nominal rate) corresponds to 32MB average size of files to be transferred. The parameters: *aver_int_time*, *aver_con_dur* as well as the number of the network users, *num_of_users*, determine the traffic load of the network (referring always to TCP-based data forwarded through the Satellite). In detail, *num_of_users***aver_con_dur*/*aver_int_time* gives the average number of active users in the network for infinite network capacity, while *num_of_users**512**aver_con_dur*/*aver_int_time* gives the average nominal aggregate traffic load forwarded to the network. The simulated values of the above traffic parameters as well as their meaning in terms of traffic load are summarized in Table I.

num_of_users	*aver_con_dur*	*aver_int_time*	**Average number of users**	**Average, nominal traffic load (Kbps)**
26	500	722	18	9219
26	500	703	18.5	9468
26	500	684	19	9731
26	500	667	19.5	9979
26	500	650	20	10240
26	500	634	20.5	10498
26	500	319	21	10753
26	500	605	21.5	11002
26	500	591	22	11262
26	500	578	22.5	11516
26	500	565	23	11781
26	500	553	23.5	12036
26	500	542	24	12280

Regarding the evaluation of our algorithm, the scheme has involved the implementation of two different simulation tools sequentially. In specific, firstly we set up the considered scenario by means of the Network Simulator ns-2 (release 2.27) in order to get TCP statistics (i.e., RTT measurements, throughput, etc.) for different values of PER. In particular, we performed simulations by configuring sender and receiver nodes, running TCP NewReno as transport protocol and FTP as application protocol.

Additionally, we developed a C++ simulation tool capable of emulating the general network environment and implementing the CAC algorithm manipulating the TCP output info. In brief, the C++ tool makes use of the aforementioned info in order to specify at regular intervals the bitrate per active connection. At last, the CAC algorithm is implemented so as to make the admission/rejection decision based on whether the aggregate bitrate of the already active users is low enough (in relevance to the available downlink bandwidth) to allow the satisfaction of the QoS requirements of the new user. A new connection is admitted to the network if at that instant:

$$Net\_Cap - \sum_{i=1}^{N} TCP\_rate(i) - Margin \leq Con\_Nom\_Rate \quad (1)$$

where:

1) Net_Cap : The capacity of the network (10Mbps)
2) TCP_rate(i) : The TCP transmission rate of i_{th} connection
3) Con_Nom_Rate : The nominal rate of the candidate connection (512Kbps)
4) Margin : This parameter has been added to the algorithm as a safety margin, to prevent the occurrence of congestion due to the fluctuation of the TCP transmission rates of the admitted connections. Its value, after several simulation runs has been chosen equal to 100Kbps.

VII. Results

The main performance factors evaluated for the efficiency of the bandwidth utilization is the *throughput* of the downlink channel, computed as the

$$throughput = \left(\sum_{i=1}^{N} TCP\_rate(i) \right) / Net\_Cap \quad (2)$$

while a second factor indicative of the network's performance is the Blocking Probability. Both of them studied for a wide range of traffic load in order to reach as safe conclusions as possible.
Being more specific, Fig. 3 shows the *throughput* of the network in the case of implementing either the basic or the proposed CAC algorithm. Fig 4 depicts the increase in *throughput* achieved by the use of the proposed scheme.

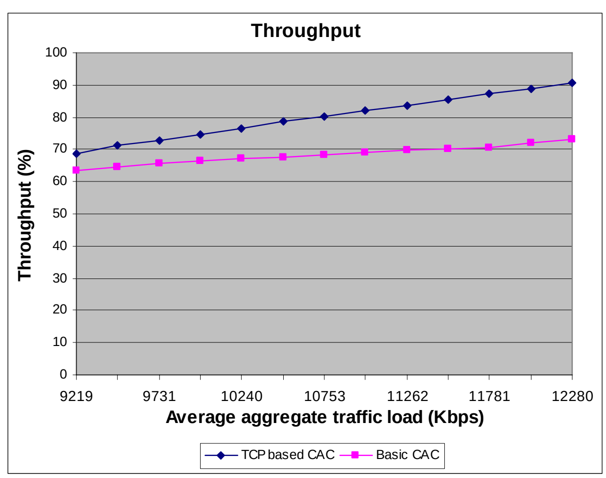

Figure 3: Throughput

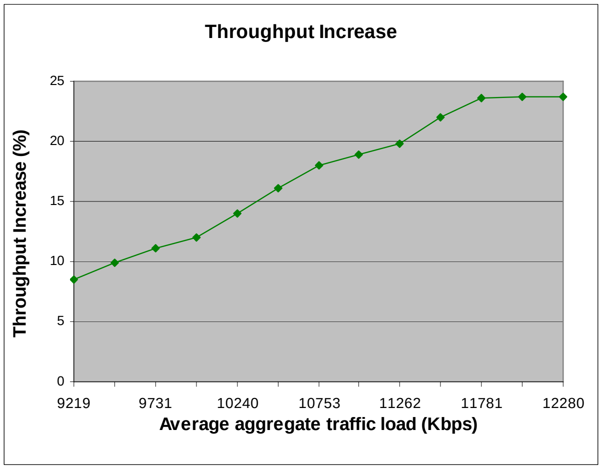

Figure 4: Throughput Increase

As it becomes apparent from Figures 3 and 4, for low traffic loads the *throughput* for both schemes is also low, while the corresponding difference between them is minor. This is due to the fact that for such low traffic loads the main reason of the *throughput* degradation is not the inefficiency of the CAC algorithm, but the fact that such a low traffic load cannot cover the bandwidth availability anyway. Thus for both algorithms the *throughput* of the network follows the increase of the traffic load. Nevertheless, in the case of the proposed scheme, the *throughput* reaches higher levels verifying the fact that there is a better utilization of the available resources.

Consequently, concerning the relative improvement introduced by our scheme, one can note that for wide range of traffic loads it is almost proportional to the increase of the traffic load, approaching the level of 25% *throughput* increase in relevance to the basic algorithm. However, from a certain point on the upper bound is set by the network capacity, causing the stabilization of the graph (it becomes parallel to the *x* axe).

Similar conclusions one can reach by studying the Figures 5 and 6 that present the Blocking Probability of the system for the same range of traffic load parameters. In this frame, a larger improvement can be detected. In fact, the proposed algorithm allows a decrease on the blocking probability ranging from 100 % (for the lower traffic load value) to 60 % (for the higher traffic load value).

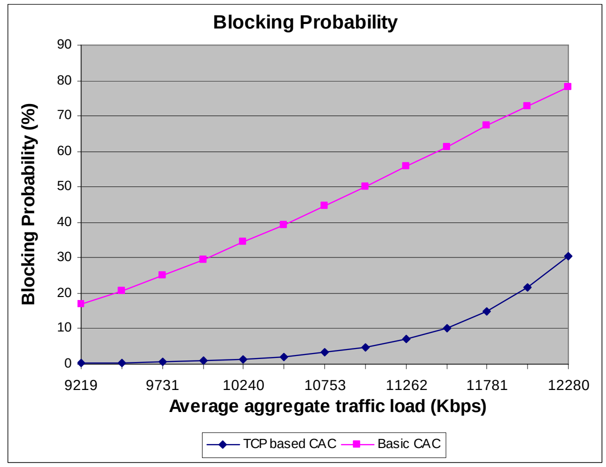

Figure 5: Blocking Probability

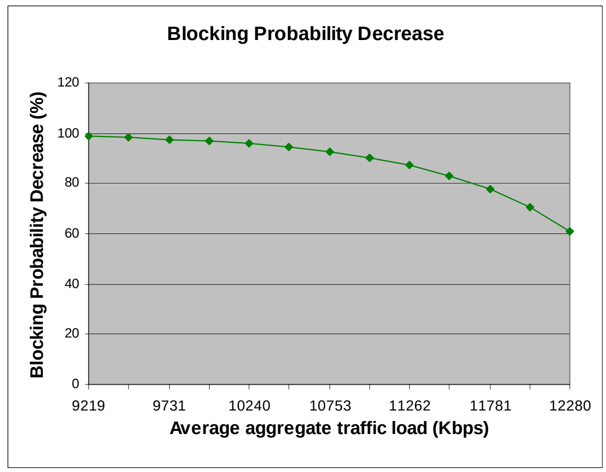

Figure 6: Blocking Probability Decrease

VIII. CONCLUSIONS

The combined use of HAP and satellite represents an innovative and challenging architecture to guarantee telecommunication broadband services even if terrestrial infrastructures are unavailable. Unfortunately, applications running TCP as transport protocol, present degraded performance due to large bandwidth-delay product and to the presence of transmission errors. In this paper, we proposed a new CAC algorithm that aims to optimize the resource utilization by using TCP inputs. By simulations, we demonstrated a considerable improvement of the performance (in terms of throughput and blocking probability), with respect to a basic CAC algorithm taking into account only QoS requirements.

ABBREVIATIONS

1) CAC : Call Admission Control
2) HAP : High Altitude Platform
3) LMDS : Local Multipoint Distribution Services
4) PER : Packet Error Rate
5) QoS : Quality of Service
6) RTT : Round Trip Time
7) TCP : Transport Control Protocol
8) DVB-S : Digital Video Broadcasting over Satellite
9) FTP : File Transfer Protocol

REFERENCES

[1] C. Partridge, and T. Shepard, TCP Performance over Satellite Links. *IEEE Network*, vol. 11, n. 5, 1997, pp. 44-49.

[2] T. Lakshman, and U. Madhow, The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss. *IEEE/ACM Transactions on Networking*, vol. 5, n. 3, 1997, pp. 336-350.

[3] G. Xylomenos, G. C. Polyzos, P. Mahonen, and M. Saaranen, TCP Performance Issues over Wireless Links, *IEEE Communication Magazine,* vol. 39, n. 4, 2001 pp. 52–58.

[4] W. Stevens, *TCP/IP illustrated, Volume 1*. Addison Wesley, 1994.

[5] M. Luglio, C. Roseti, and M. Gerla, The Impact of Efficient Flow Control and OS Features on TCP Performance over Satellite Links, *ASSI Satellite Communication Letter (Sat-Comm Letter), 9th edition, special issue on Multimedia Satellite Communication*, vol. III, n. 1, 2004, pp. 1-9.

[6] W. Stevens, TCP Slow Start, Congestion Avoidance, Fast retransmit, and Fast recovery Algorithms, *Internet RFC 2001,* 1997.

[7] V. Jacobson, Congestion Avoidance and Control, *Proceedings of ACM SIGCOMM '88 Conference*, 1998, pp. 314-329.

[8] Y. Zhang, Interworking and Computing over Satellite Networks. Kluwer Academic Publishers, 2003.

[9] T. C. Tozer and D. Grace, "High-altitude platforms for wireless communications", Electronics & Communication Engineering Journal, vol.13, no.3, pp.127-137, Jun 2001.

[10] S. Karapantazis, N. F. Pavlidou, "Broadband communications via high-altitude platforms: a survey", IEEE Communications Surveys & Tutorials, Vol. 7, no. 1, pp. 2-31, 2005.