A CROSS-LAYER MECHANISM FOR THE EFFICIENT MANAGEMENT OF TCP-BASED TRAFFIC

Giovanni Giambene[1]
Michele Luglio, Cesare Roseti [2]

Abstract

This paper deals with an Interactive Satellite Network (ISN), based on the DVB-RCS standard, where terminals communicate with the Network Control Center (NCC) through a bent-pipe satellite and a return link based on an MF-TDMA scheme. In our study we consider that TCP-based application running on remote terminals generate traffic towards the NCC. A bandwidth on demand scheme is operated at the NCC that centrally allocates resources on the basis of the requests made by the terminals. A cross-layer design approach is proposed in this work to allocate resources at the MAC layer according to the needs of the TCP congestion window behaviour. Resource requests are made by terminals according to a forecast on their congestion window trend. The NCC allocates resources to terminals and can limit the increase in their congestion window in order to avoid that they congest the network. The obtained results are that the proposed dynamic bandwidth allocation permits to improve the throughput at the TCP level (time outs are avoided) and allows reducing the transfer time.

1. Introduction

Telecommunication and broadcasting services today have an increasing interest in supporting interactivity. Customers want to select, order, store and manage what they receive on their terminals and, ideally, they also need to interact from the same terminal with the network. Hence, the distribution network is becoming an asymmetric interactive network. Satellite communications combine high bandwidth, wide area coverage, reconfigurability, and multicast capabilities. Satellite communication systems represent an interesting solution to provide Internet connectivity to users located either in remote areas or in locations where fiber cabling does not represent a viable choice.
DVB-RCS is a standard that supports interactivity in broadband satellite networks. In such systems, a certain number of terminals communicate with the *Network Control Center* (NCC) through the return channel by means a GEOstationary (GEO) orbital configuration that is characterized by large propagation delays, thus yielding a large *Bandwidth Delay Product* (BDP) value in case of high-bit-rate.
The *Transmission Control Protocol* (TCP) is the standard transport layer protocol, on which most of the Internet applications are based. TCP is based on an ACK mechanism that regulates the flow control scheme and the congestion control technique. Such mechanisms are based on dynamically updated windows: the receiver advertised window and the congestion window. On the basis of the received ACKs during a first phase the congestion window is increased according to an exponential law (slow start) and during a second phase according to a linear growth on a round trip time basis (congestion avoidance). Over a long latency path, such as satellite links, TCP performance is seriously compromised.
Data are injected in the network considering a sliding window equal to the minimum between congestion window and receiver window. The rationale is that the sender doesn't know in

[1]CNIT - Research Unit of Siena, Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Siena, Via Roma, 56 - 53100 Siena, Italy, email: giambene@unisi.it
[2]Università degli Studi di Roma "Tor Vergata", Via del Politecnico, 1 - 00133 Roma, Italy, email: {luglio , cesare.roseti}@uniroma2.it

advance the congestion state of the network. On the other hand, in an ISN the NCC has full control of the status of resources and their allocation. Hence, in such a case, it would be important that the NCC dynamically allocates resources to terminals for return link transmissions according to the evolution of the TCP windows and, in particular, of the congestion window. Moreover, satellite communications are typically affected by packet losses that reduce the goodput at TCP level and cause a highly dynamic behavior of the TCP traffic. This behavior can cause inefficiency on lower layers (mainly layer 2 that contains the radio resource allocation protocol).
Using the CF-DAMA protocol, the free capacity is uniformly distributed among the terminals active in a certain instant in the network, without considering the state of each active connection. Whereas, a capacity allocation strategy, which takes into account the TCP window trend, can optimize and fairly share the available resources.
This paper presents an innovative resource allocation algorithm based on a cross-layer interaction between TCP and MAC layer. Such an algorithm aims to synchronize the requests of resources with the TCP transmission window trend in order to assign/remove capacity dynamically on the basis of the actual transmission state of each data source.

2. Reference Scenario

The reference scenario is an ISN, based on the DVB-S/DVB-RCS standard [1],[2]. DVB-S is used for the forward link (from NCC to RCSTs) and DVB-RCS is employed for the return link (from RCSTs to NCC). Below the transport layer and the IP layer the *Multi Protocol Encapsulation* (MPE) provides segmentation & reassembly functions. MPEG2-TS (Transport Stream) packets of fixed length (188 bytes) are transmitted according to time division multiplexing. To the block of data coming from the application layer, a TCP header of 20 bytes an IP header of 20 bytes and an MPE header+CRC trailer of 12+4 bytes are added; such bytes are fragmented in the payloads of MPEG2-TS packets. The DVB-RCS multiple access discipline on the return link is of the MF-TDMA. According to this scheme resources are time slots on different available carrier frequencies with different possible available bandwidths. DVB-RCS resources are divided in super-frames that are characterized by suitable portions of time and frequency bands; each super-frame is divided in frames that are composed of a certain number of time lots. The frames can have different duration, bandwidth and number of timeslots.
In our scenario we envisage a GEO bent-pipe satellite, user terminals (i.e., *Return Channel Satellite Terminals*, RCSTs) and an NCC, according to a classical star topology (see Fig. 1). RCSTs are fixed with *Return Channel via Satellite* (RCS) that allows transmitting data or control signaling. The NCC is the core of the network: it provides control and monitoring functions and it manages network resources (i.e., time slots on different available carrier frequencies) allocation according to a *Bandwidth on Demand* (BoD) approach.
FTP like traffic is considered to be conveyed from RCTSs to the NCC through the TCP protocol. RCSTs send their resource requests to the NCC. The NCC looks at the available uplink resources and sends a broadcast response to the RCSTs (forward channel) by means of *Service Information* (SI) tables (i.e., the *Burst Time Plan*, BTP, sent every superframe). The NCC assigns to each active RCST a set of bursts, each of them is defined by a frequency, a bandwidth, a start time and a duration.

3. Cross-Layer Basics Concept

The communication protocol stacks are based on a layered architecture paradigm. In fact, the protocol suites provide distinct functional modules by the definition of different protocol

layers. The main benefit of such an architectural approach is that it facilitates the complex design of the network architecture. Nevertheless, the changes on the communication features, leading to full coverage services, ubiquitous mobile access, diverse user devices, autonomous networks and software dependence, require new design methodologies. In such a context, the main issue is that the lack of information to share among the protocol layers can cause a sub-optimal utilisation of the available radio resource. Therefore, a cross-layer approach has been introduced to replace the strict layered protocol stacks [3]. It aims to guarantee a sharing of information also among not-adjacent layers in that network where the assumptions, suitable for the wired stacks, result inadequate. For example, the well-known assumption in TCP protocol, that packet losses are just due to network congestion, decades in wireless systems where the time-varying channel conditions can cause the corruption of TCP packets. In these cases, the reception of information on the channel state from the lower layers, may allow TCP to take more suitable decisions in setting its congestion window and then the transmission rate. Furthermore, the diffusion of wireless, satellite, ad-hoc, heterogeneous (in terms of both traffic and communication environment) and personal area networks, requires a co-ordinated adaptation from multiple layer and an optimisation of the network performance in the case the transmission power and the radio resources are rare and must be used with the maximum efficiency. As a consequence, the cross-layer model, shown in fig. 2, is becoming the simpler and the most flexible approach to design the next-generation communication systems.
The "vertical" communication between not-adjacent layers can be obtained by different methods [4]. A brief survey of the main methods described in the literature is as follows:

- *Packet headers*. In IPv6, additional headers can be used to carry information by in-band message carriers (Interlayer Signalling Pipe).
- *ICMP Messages*. Dedicated internal message can be exchanged by creating appropriate holes in the protocol stack [5],[6].
- *Management by "third parts"*. Internal or external "entities" store, update and distribute information coming from not-adjacent layers.

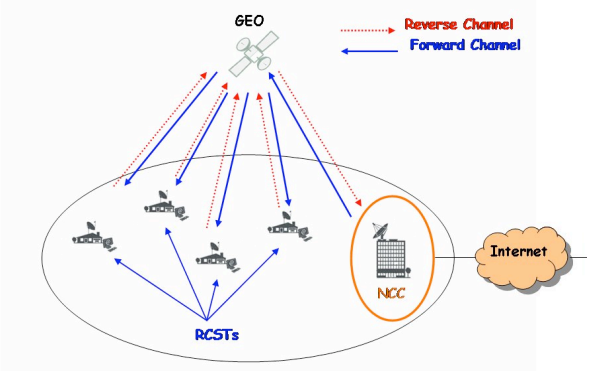

Fig. 1. System architecture.

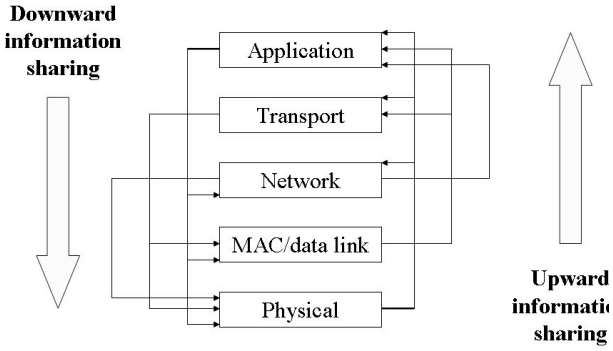

Fig. 2. Cross-Layer Design Model.

4. TCP Mechanisms

TCP is a transport protocol that provides a connection-oriented, reliable and byte stream service to the higher layers [7]. Basically, a transmitting TCP agent accepts a data flow from the application and divides it into sized chunks ("segment" or "packet"), identified in unambiguous manner by a sequence number. On the receiver side, TCP reacts to the reception of a packet by issuing an acknowledgement (ACK) to the sender and by delivering data to the application. ACKs are cumulative, so that the reception of an ACK, corresponding the i-th packet, notifies to the sender that also the previous packets have been successfully received. Three main TCP functions can be identified as flow control, congestion control, error control.

Flow Control. TCP implements a mechanism, called "sliding window", which allows the receiving TCP agent to control the amount of data "in –flight" in a given time instant. In particular, the receiver limits the maximum transmission window size by using a variable called "advertised window". The aim is to avoid that the transmission rate exceeds the receiver buffer capacity.

Congestion Control. To adaptively react to the congestion state of the network, TCP uses two algorithms: *Slow Start* (SS) and *Congestion Avoidance* (CA) [8]. In detail, TCP probes the state of the network by gradually increasing its "congestion window" (cwnd) until congestion occurs and packets are dropped. Initially, during the SS phase the cwnd increases exponentially on a round-trip time basis. When cwnd reaches a threshold value ("slow start threshold") TCP switches to congestion avoidance and the window increases linearly.

Error Control. The default TCP loss recovery mechanism is the "timeout". Basically, after the transmission of a packet, the sender expects to receive the corresponding ACK within a "timeout interval". If the ACK is not received in such an interval, TCP retransmits the packet, resets the timeout interval, halves the slow start threshold and sets cwnd to 1. Waiting for a timeout and re-starting in SS phase, every time a loss occurs, leads to waste a large amount of bandwidth. To mitigate such a problem, TCP Reno (1990) introduced the "Fast Retransmit" and "Fast Recovery" algorithms [8]. Since TCP generates duplicate ACKs (dupACKs) when an "hole" is detected in the packet sequence, the Fast Retransmit algorithm considers the arrival of 3 dupACKs as an indication that a packet has been lost and retransmits it immediately. In addition, Fast Recovery interprets the packet lost as a congestion indication and reduces both cwnd and slow start threshold to half of the current cwnd value.

5. TCP over Satellite: Resource Allocation Requirements

TCP performance over a link including a GEO satellite segment are mainly limited by the long latency, the large BDP and the presence of errors in the received flow [9][10]. In particular, since TCP interprets every packet loss as a clear indication of network congestion, the occurrence of transmission errors leads to unnecessary reduction of the congestion window, and then of the achieved rate. On the other hand, the long *Round-Trip Time* (RTT) does not allow TCP to quickly recover from errors and to fill easily the pipe (equal to bandwidth-RTT product). In a such context, the amount of resources really used by a TCP connection dynamically changes over the time, causing sub-optimal performance in terms of channel utilization when fixed allocation strategy are considered.

At the same time, the MAC protocol (Radio Resource Management) plays a fundamental role to guarantee good performance to higher-level protocols by managing the arbitration of return link access. In fact, RRM can significantly impact the end-to-end performance of TCP flows over a satellite network. To optimize RRM, it would be desirable that the resource allocation were driven by TCP congestion window evolution for each flow [11]. According to this concept, the NCC will allocate to each active RCST a number of slots as a function of the RCST requests. Unfortunately, in such a scenario, the allocation of new resources is very slow, in fact it needs about 500 ms (time between the transmission request and the NCC response's arrival instant). At the same time, the TCP window may double if the "Slow Start" phase is running. This causes basically two effects:

1. The terminal will have to send every RTT new resource allocation requests to NCC;
2. The number of packets stored in the MAC queue of the RCST will increase exponentially when the Slow Start phase occurs.

A possible solution, to reduce the time that a packet must wait in the queue before being considered for transmission, can be based on a cross-layer interaction between MAC and transport layer.

6. Design of the Cross-Layer Interaction Between TCP and MAC

We apply the cross-layer approach to “synchronize” the TCP window growth with the amount of resources dynamically assigned by NCC (layer 2 functionality) in a typical “star”-based satellite network. In particular, our main idea is to assign/remove resources taking into account the TCP trend of each active connection. Every TCP connection will inform time by time the MAC layer on the expected congestion window evolution, and then every terminal (RCST) will negotiate resources with the NCC on the basis of such information. Hence, we move from a terminal-based to a connection-based allocation scheme. Therefore, a single RCST can perform a large number of requests to NCC. In this paper, in order to analyse the effect of the TCP-MAC cross-layer interaction, we assumed a one-to-one relationship between RCSTs and connections. Nevertheless, such a choice does not entail a restriction for the application of the proposed approach. In fact, while scalability problems occur, the whole cross-layer & BoD process can be split: on the RCST the single connection allocation request will be managed and the assignment algorithm will run, while a “cumulative” resource request will be sent to the NCC.
The cross-layer information will be exchanged between TCP and MAC by ICMP message. The definition of a dedicate “hole” between transport and link layer saves layer-by-layer processing efforts. Furthermore, as far as the practical implementation of the cross-layer is concerned, the ICMP method is already supported by Linux OS.

7. Cross-Layer Based Resource Allocation Algorithm

The cross layer resource allocation algorithm is based on the use of TCP parameters, such as “congestion window” and “slow start threshold”, to carry out an estimate of the needed resources for a given TCP flow. From the comparison of these two parameters it is possible to determine the TCP congestion control status (i.e., *slow start* or *congestion avoidance*). Accordingly, the MAC layer can know the law according to which the congestion window is enlarged on an RTT basis and can predict the necessary resource allocation for each TCP flow. Hence, it is hopefully expectable a remarkable reduction of queuing delay (with a significant reduction of time out expirations) and consequently an efficient utilization of channel bandwidth.
To realize such a cross-layer-based allocation scheme, specific tasks have been assigned at both RCST and NCC sides.

7.1. RCST side

TCP communicates both the actual congestion window (a_cwnd) and the slow start threshold (ssthresh) value to a "Cross-Layer (CL) agent", which uses them to evaluate the TCP phase (Slow Start or Congestion Avoidance) and to estimate the expected congestion window (next_cwnd) value for the next RTT. Such information is delivered to the OSI layer 2 and then encapsulated into two new fields in the MAC header to be used by the MAC layer on the NCC side.

7.2. NCC side

The NCC receives all incoming packets and compares the next_cwnd estimation with the amount of resources already assigned to the specific source. After comparison, the NCC may make available capacity in the resource space (next_cwnd < assigned resources) or generate requests of additional resources (next_cwnd > assigned resources). All the requests are inserted in one of two different priority-queues. In particular, if the TCP phase flag is set to 1, the NCC considers the corresponding TCP source in "Slow Start" and then forwards the request in a high-priority queue. If the TCP phase flag is set to 0, all the requests are

forwarded in a low-priority queue. As a matter of fact, the proposed assignment policy basically relies on a two-level priority strategy which first privileges the connections just started (first level of priority) and then the connections with a larger transmission window (second level of priority). If some pending requests can be satisfied within the current superframe, NCC stops the growth of the corresponding source until new resources will be available in order to limit the congestion on the transmitting MAC queue.
Furthermore, once NCC have assigned all the available resources, the allocation algorithm switches in a "fair" mode, in which, every RTT, a resource is taken from the connection with the larger amount of resources to be assigned to that one with less assigned resources.

8. Results

In order to evaluate the effect of the cross-layer interaction on the end-to-end performance, a simulation campaign has been carried out by utilizing the Network Simulator ns-2 (release 2.27) [12]. In particular, we have reproduced a typical GEO satellite network, where RCSTs are connected to a "core" network element, NCC. By considering the return link (from RCSTs to NCC), we have implemented a TCP transmitting agent on each RCST, while a receiving TCP agent is implemented on the NCC node. In our simulations, TCP sources start an FTP transfer (supposed of equal size) at regular time intervals. The main simulation parameters are listed in Table 1. Furthermore, we have modified the C++ code in order to simulate the MF-TDMA access scheme and the NCC tasks, and we have added two specific classes, "*Cross-layer*" and "*BoD-algorithm*", to implement respectively the "vertical" communication, between transport and MAC layer, and the proposed allocation algorithm.

RTT	~ 508 ms
Return link bandwidth	2 Mbit/s
Packet Error Rate	$[10^{-4}; 10^{-3}; 5*10^{-3}]$
TCP Packet Size	1500 bytes
Transport Protocol	TCP NewReno
Application Protocol	FTP
Maximum number of TCP sources	32
File size	5 Mbytes

Table 1. Simulation Parameters.

8.1. Analysis of the allocation process

Several simulations have been run to analyze the dynamic assignment/removal of resources when different TCP flows share the return link. For instance, Fig. 3 focuses on the variations on the allocated resources, in the case that two TCP connections start in different time instants. In particular, the obtained results allow the following considerations:

1. The assignment of the resources strictly follows the TCP congestion window trend: exponential when the SS phase is performed, and linear when TCP switches in the CA phase;
2. After about 23 seconds, all the available resources have been assigned. Then, the allocation algorithm enters the "fair" mode, and, every RTT, takes a resource from connection 1 to assign it to connection 2;
3. When the resources result fairly split between the two active connections, the NCC stops the TCP window growth of both the connections avoiding to fill the terminal queues;
4. After about 27 seconds, connection 2 suffers from a single loss, and consequentially its congestion window is halved. Therefore, the NCC accordingly reduces the

resources assigned to connection 2 allowing connection 1 to temporarily utilize the free resources.

As a result of this analysis, we can conclude that the proposed algorithm leads to an optimal utilisation of the whole bandwidth and fairness among connections is guaranteed as well.

8.2. Performance evaluation

To show the benefits of our cross-layer scheme, we run some preliminary tests to reproduce a scenario where a variable group of RCSTs access the return link to send files by using the FTP application protocol. In particular, we evaluated the end-to-end performance in terms of:

- Average time needed to transfer a file of fixed size vs packet error rate (Fig. 4);
- Channel utilisation as function of the time when a group of RCSTs starts the transfer of a sized file regular time intervals (Fig. 5).

In both cases, we compared the cross-layer scheme with a fixed allocation strategy, which assigns to each active connection an amount of resources (equal to the ratio between the total resource and the number of active connections) in static mode.
In particular, in Fig. 4 the average time to transfer a 5 Mbytes file is shown, in the case that 15 connections start at regular intervals of 5 seconds. The simulation outcomes show that the cross-layer-based allocation scheme leads to a transfer time reduction from 27% (PER=10^{-4}) up to 36% (PER=$5x10^{-3}$).

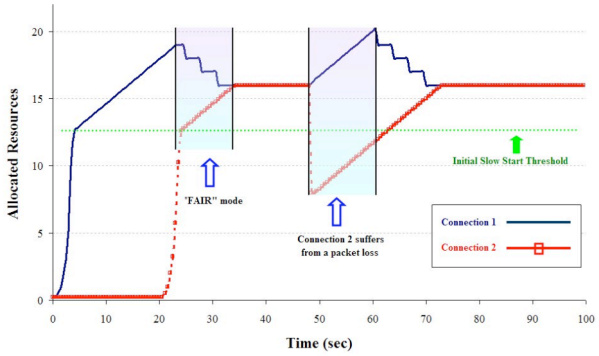

Fig. 3. Dynamic resource allocation/removal for two TCP connections.

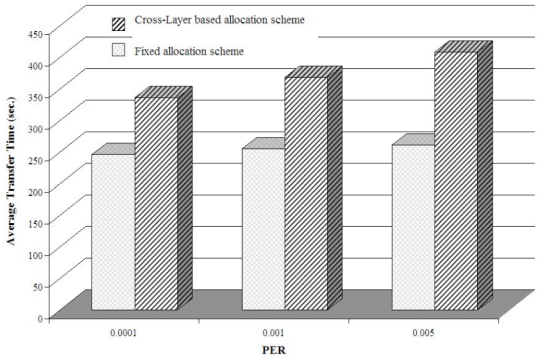

Fig. 4.: Average transfer time vs. PER**.**

Finally, we have configured a simulation scenario where 5 TCP connections, starting at time intervals of 5 seconds, share the return link (PER=10^{-3}) to transfer 5 Mbytes files. Then, we have evaluated the channel utilisation over the time. The results, depicted in Fig. 5, clearly highlight how the proposed cross-layer allocation scheme guarantees an optimal channel utilisation in spite of the presence of an high error rate.

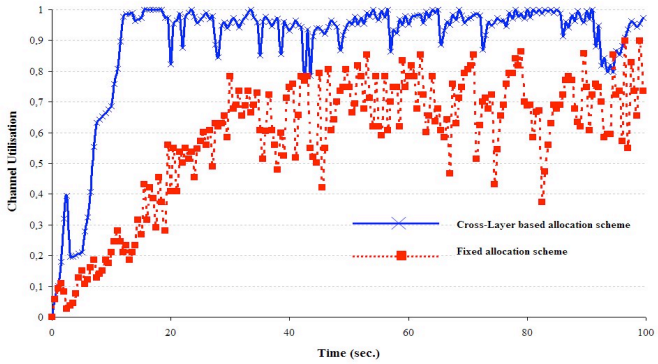

Fig. 5.: Channel utilization.

9. Conclusions and Future Work

The impact of the bandwidth allocation strategy on the TCP performance represents a critical issue in satellite networks. The paper presented an innovative approach, based on a cross-layer interaction between TCP and MAC. After analysis and preliminary simulations we have shown the potential benefits that are allowed by an information exchange between TCP and MAC layer. First of all we have proposed a new paradigm to dynamically allocate/remove network resources, no more based on the terminal but on the single connection state. Of course, in order to validate the proposed allocation scheme a more refined design of the algorithm and the comparison with existing allocation protocols (i.e., CF-DAMA) for the satellite return link are planned as future work.

Acknowledgements

This paper has been funded by the "SatNEx" NoE project (contract No. 507052) under the 6th framework of the European Commission - joint activity 2430 (A&TCP).

References

[1] ETSI EN 301 790, "Digital Video Broadcasting (DVB); Interaction Channel for Satellite Distribution System," V1.3.1, 2003.

[2] ETSI TR 101 790, "Digital Video Broadcasting (DVB); Interaction Channel for Satellite Distribution System, Guidelines for the use of EN 301 790," , V1.2.1, 2003.

[3] Z. H. Haas, "Design methodologies for adaptive and multimedia networks," Guest Editorial, IEEE Commun. Magazine, Vol.39, No. 11, pp. 106-107, November 2001.

[4] Qi Wang, M.A. Abu-Rgheff, "Cross-Layer Signalling for Next-Generation Wireless Systems," Wireless Commun. and Networking, Vol. 2, pp. 1084-1089, March 2003.

[5] J. Postel, "Internet Control Message Protocol," RFC 792, September 1981.

[6] A. Conta, S. Deering, "Internet control message protocol (ICMPv6) for the Internet Protocol version 6 (IPv6)," RFC 1885, December 1995.

[7] W. Stevens, "TCP/IP illustrated," vol. 1, Ed. Addison Wesley, 1994.

[8] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast retransmit and Fast Recovery Algorithms" Internet RFC 2001, 1997.

[9] C. Patridge, T.J. Shepard, "TCP Performance over Satellite Links," IEEE Network, vol. 11, No. 5, pp. 44-49, 1997.

[10] M. Luglio, C. Roseti, M. Gerla, "The Impact of Efficient Flow Control and OS Features on TCP Performance over Satellite Links," ASSI Satellite Communication Letter, vol. 3, No. 1, pp.1-9, 2004.

[11] E. Guainella, A. Pietrabissa, "TCP-Friendly Bandwidth-on-Demand for Satellite Networks," ASMS Conference, July 2003.

[12] NS-2 Network Simulator (Ver. 2) LBL, URL: http://www.mash.cs.berkley.edu/ns/