

TECHNICAL RESEARCH REPORT

Simulation Studies of a Hybrid Network in Order to Enhance
the Performance of Hybrid Internet Service

by A. Gaid, J. Baras

CSHCN T.R. 98-9
(ISR T.R. 98-18)



The Center for Satellite and Hybrid Communication Networks is a NASA-sponsored Commercial Space Center also supported by the Department of Defense (DOD), industry, the State of Maryland, the University of Maryland and the Institute for Systems Research. This document is a technical report in the CSHCN series originating at the University of Maryland.

Web site <http://www.isr.umd.edu/CSHCN/>

Simulation Studies of a Hybrid Network in Order to Enhance the Performance of Hybrid Internet Service*

Ahmed Y. Gaid[†] and John S. Baras
Center for Satellite and Hybrid Communication Networks
Institute for Systems Research
University of Maryland
College Park, MD 20742

Abstract

The work reported here addresses the modeling and simulation of a hybrid network that conforms to Hybrid Internet Service, which is an Internet access currently commercialized by Hughes Network System. The objective is to research and discover performance enhancement techniques for Hybrid Internet Service. Hybrid Internet Service intends to provide Internet end-users with high bandwidth by delivering packets over a satellite channel. It takes advantage of the fact that the vast majority of Internet end-users don't send as much data as they receive. However, long delays experienced over the satellite channel (in the case of a GEO satellite) have catastrophic consequences on the overall throughput with Commercial-Off-The-Shelf TCP/IP stacks. This problem is addressed by the Asymmetric TCP/IP protocol implemented within a gateway, the hybrid gateway. The hybrid gateway acts as a go-between for the server and the end-user. The hybrid gateway both acknowledges packets on behalf of the end-user and advertises a larger window to the server. We have implemented the Asymmetric TCP/IP within the hybrid gateway with OPNET, which is an industrial strength, popular network modeling and simulation software. A discussion on the current results of the simulation in the broader perspective of hybrid networks development is provided following the presentation of both Hybrid Internet Service and OPNET along with the various built-in OPNET models used in the course of the simulation. The methodology we have followed is also included in this report.

*Supported by NASA NCC3-528 and Lockheed Martin Chair.

[†]Exchange Student from the Ecole National Supérieure des Télécommunications (ENST), Paris, France.

Contents

1	Hybrid Internet Service	1
2	OPNET	2
2.1	Overview of OPNET	2
2.2	Models used within OPNET	2
2.3	IP Fragmentation and Reassembly Strategy performed within OPNET	4
2.4	General considerations when modeling a network	9
3	Asymmetric TCP/IP Protocol	10
4	Implementation of the Asymmetric TCP/IP protocol within the hybrid gateway	13
4.1	Processing of segments arriving from the server	13
4.2	Processing of segments arriving from the hybrid host	14
4.3	The establishment of the connection	14
5	Structure of the model	16
6	Results of the simulation	22
6.1	Validation of the model	23
6.2	Guidelines for the simulation	23
6.3	Results and parametric curves	24
6.3.1	First set of results with standard parameters	24
6.3.2	Results with a higher speed modem	32
6.3.3	Second set of results with hybrid parameters and a large gateway window	39
6.3.4	Third set of values with large parameters	48
6.3.5	Fourth set of values with a smaller ACK timeout at the hybrid gateway	55
7	Future Enhancements of the simulation model	57

List of Tables

1	Value of the MTU depending on the network	7
2	Statistical outputs in the TCP process	22
3	Values of the standard parameters	24
4	Values of the standard parameters with a high speed modem	32
5	Values of the hybrid parameters with a large hybrid gateway window	39
6	Values of the large parameters	48

List of Figures

1	TCP Finite State Machine	3
2	TCP manager process	4
3	TCP connection process	5
4	Real IP datagram (a) versus OPNET IP datagram (b)	6
5	Fragmentation and reassembly schemes within OPNET	8
6	Asymmetric TCP/IP versus standard TCP/IP for acknowledgment policy	11
7	Processing of segments arriving from the server	13
8	Three hand-shake protocol	15
9	Server working over a LAN	16
10	Hybrid Gateway	17
11	Hybrid Terminal	18
12	Hybrid Network	19
13	Capacity of the channels at the hybrid gateway	19
14	Capacity of the channels at the hybrid terminal	20
15	Downstream and upstream channels in the hybrid network	21
16	Throughput at the end-user and at the hybrid gateway	25
17	Downstream data load	26
18	Sequence number received by the hybrid host and ACK number received at the hybrid gateway	27
19	Window advertised by the hybrid host at the hybrid gateway	28
20	Amount of data sent by the hybrid gateway to the hybrid host	29
21	Comparison between the window size and the amount of data eligible to be sent by the hybrid gateway	30
22	RTT measured at the server and the hybrid host	31
23	Throughput at the end-user and at the hybrid gateway	33
24	Sequence number received by the hybrid host and ACK number received at the hybrid gateway	34
25	Window advertised by the hybrid host at the hybrid gateway	35
26	Amount of data sent by the hybrid gateway to the hybrid host	36
27	Comparison between the window size and the amount of data eligible to be sent by the hybrid gateway	37
28	RTT measured at the server and the hybrid host	38
29	Throughput at the end-user and at the hybrid gateway	40
30	Downstream data load	41
31	Sequence number received by the hybrid host and ACK number received at the hybrid gateway	42
32	Example of a duplicated segment received by the hybrid host	43
33	Window advertised by the hybrid host at the hybrid gateway	44
34	Amount of data sent by the hybrid gateway to the hybrid host	45

35	Comparison between the window size and the amount of data eligible to be sent by the hybrid gateway	46
36	RTT measured at the server and the hybrid host	47
37	Throughput at the end-user and at the hybrid gateway	49
38	Comparison of the throughput at the end-user	50
39	Window advertised by the hybrid host at the hybrid gateway	51
40	Amount of data sent by the hybrid gateway to the hybrid host	52
41	Comparison between the window size and the amount of data eligible to be sent by the hybrid gateway	53
42	RTT measured at the server and the hybrid host	54
43	Throughput at the end-user and at the hybrid gateway	55
44	Comparison of throughput at the hybrid host and at the hybrid gateway	56

1 Hybrid Internet Service

The research work described in this report addresses the modeling and simulation of a hybrid network that conforms to Hybrid Internet Service as explained in [1]. Hughes Network Systems and the Center for Satellite and Hybrid Communication Networks of the University of Maryland at College Park have been collaborating in the development of a hybrid access to the Internet that provides high-bandwidth to the end-user [1], [2], and [3]. Now that the system is currently working, several enhancements are expected to be added to the service to further increase its performance, that is the bandwidth offered to the end-user. DirecPC TurboInternetTM is the commercial name of the service and is currently the only available successful technology that provides to end-users only connected to the Internet by a regular phone link with high bandwidth.

The service is based on the fact that the vast majority of end-users don't receive as much data as they send. Basically, all what the typical end-user sends is control data and commands to browse through the Internet. On the contrary, a typical end-user is really dependent on the available bandwidth when he is downloading web pages and images, sound files or even software; i.e. all the multimedia applications that have boosted the development of the Internet and made it so popular to the general public. Not only is the Hybrid Internet Service addressed to the private end-user, but it is also company-use (like business intranets) oriented and offers a very promising development for professional applications by allowing great amount of data to be sent to different locations, in a multicast way; for example transmission of overall databases, at no additional cost.

The DirecPCTM product line is part of the recent development of hybrid networks where different kinds of physical channels are used in order to provide to the end-user the bandwidth required by modern applications. Hybrid Networks require an adaptation of existing protocols and in the same time must fit existing system and commercial device specifications and operations.

2 OPNET

The simulation has been done within OPNET, Optimized Networks Engineering Tools, which is a hierarchical modeling and simulation package.

2.1 Overview of OPNET

OPNET is an event-driven simulation software system. That means that time advances whenever scheduled events occur. Modeling is done in a hierarchical fashion from the very lowest elements which are the Finite State Machines to the network as a whole.

The fundamental concepts within OPNET are Packets, Interrupts and Inter-process communications. The way a node works in a particular sub-network is programmed in a Finite State Machine with Proto-C. Proto-C is very close to C language. Proto-C just adds to C a large library of routines implemented within OPNET and dealing with the handling of packets, interrupts and inter-processes communication. OPNET offers, in addition to the modeling and simulation tools, powerful tools to analyze the results of the simulation.

2.2 Models used within OPNET

OPNET provides several communication network model modules ready to be used. Among them there is a TCP/IP stack working over a LAN. The TCP/IP stack provided with OPNET follows closely the RFCs 793 and 1122 which define the Internet protocols. The features implemented within OPNET are the following:

- Connection establishment and closing through three-way handshake protocols.
- Slow start congestion avoidance and flow control.
- Computation of the smoothed round trip time estimate using Jacobson's algorithm in order to update the value of the retransmission timeout.
- Fragmentation and reassembly of IP datagrams.
- Resequencing, persistent time-outs.

The TCP module invokes two processes. The first one, **tCP manager**, is a root process which works continuously. The second one, **TCP connection**, is invoked whenever a new connection is created. TCP connection is a child process of TCP manager which handles a TCB, Transmission Connection Table, and routes packets toward the adequate TCP Connection process. There are as many TCP Connection processes as connections. In that way, each TCP module can handle as many connections as made possible by the available resources, in a dynamic fashion.

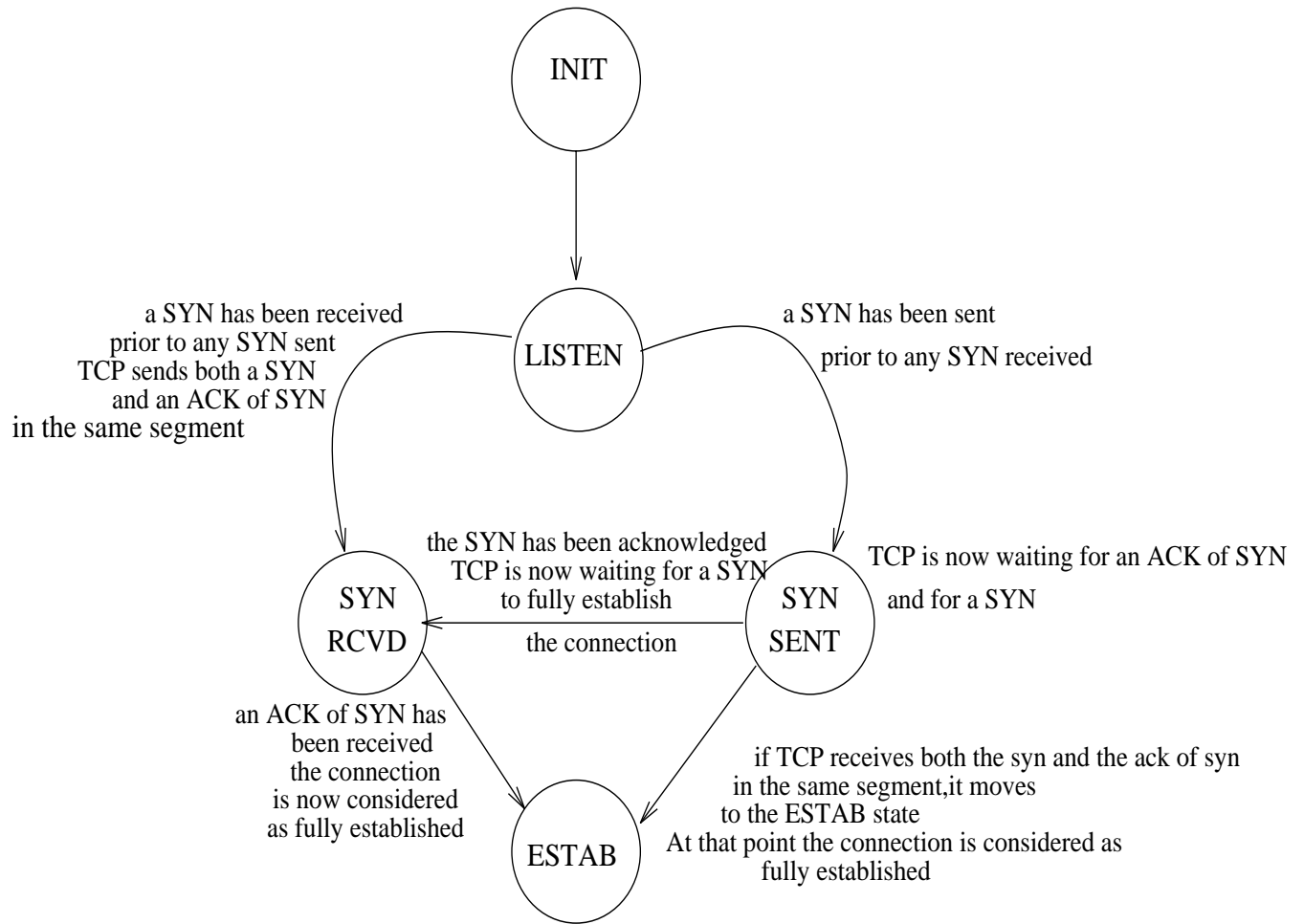


Figure 1: TCP Finite State Machine

The delivery of packets to a particular node respects the socket format:

- IP address
- port number
- and an additional feature: connection ID

IP performs the routing tasks according to routing tables loaded in files with the extension .gdf. The IP addresses are plain integers but the size of the field of the IP address within a datagram respects the real one, which is 4 octets. In the same way, the Ethernet addresses are integers and the ARP conversion tables are loaded in files.

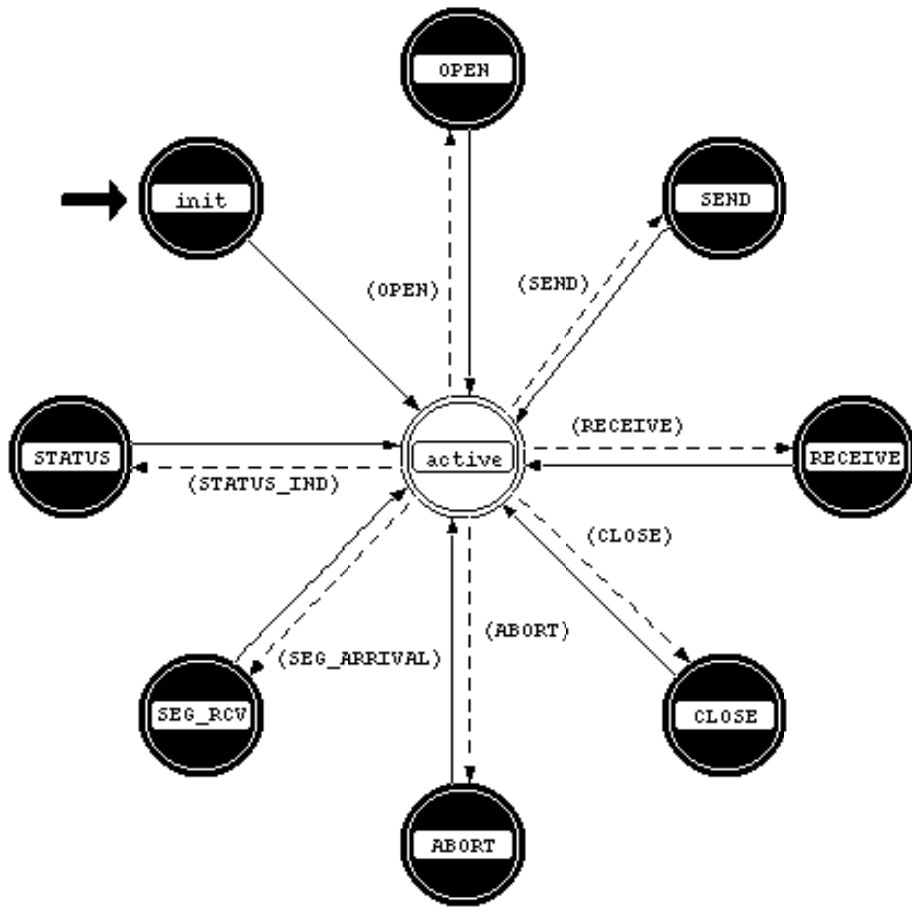


Figure 2: TCP manager process

2.3 IP Fragmentation and Reassembly Strategy performed within OPNET

IP datagrams forwarded toward an interface can face fragmentation if their length is higher than the Maximum Transfer Unit (MTU) maximum size of a packet allowed thru this interface. In the same way fragment's packets can experience fragmentation when passing thru another interface. When simulating an IP based network, two requirements must be fulfilled:

- perform fragmentation which induces effects on traffic flow by multiplying the number of packets handled with at the lower layers and in the routers;
- keep track of the data which is fragmented and provide reassembly at the end-user stage

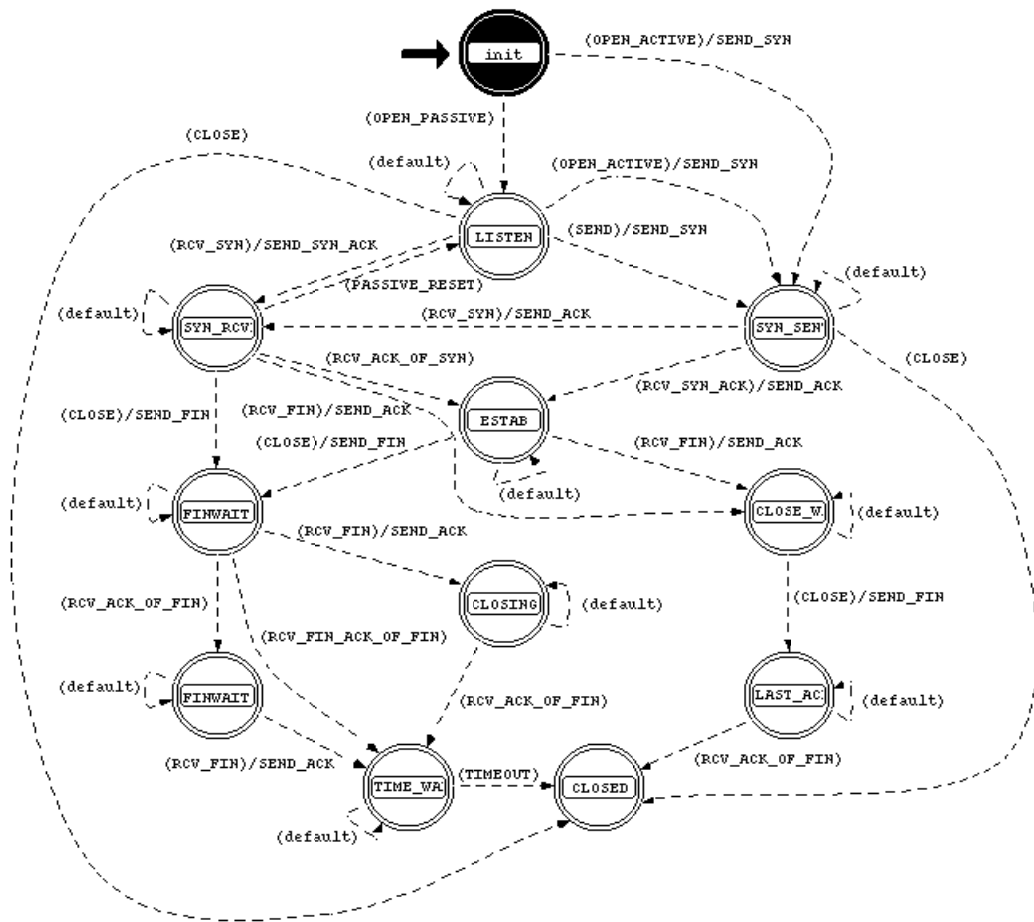


Figure 3: TCP connection process

The OPNET strategy to model fragmentation and reassembly is to generate the number of packets fragmented according to the size of the original packet and the MTU but to keep the entire packet in the last packet created by the fragmentation in order to make the reassembly easy. Thus OPNET complies with the two previous requirements. This is possible with the IP-datagram format which contains extra fields compared with the standard format of IP-datagram. This is allowed as long as it does not affect the logical size of the packet contained in the length size. The consequences of this strategy are: processing time is underestimated, required resources are not taken into account. A possible adjustment is to introduce an artificial time processing.

The packet or the segment being encapsulated within the IP datagram is affected at the field data of the last fragmented packet created. This field is a 0-bit field. The bulk size of the IP datagram is used instead to model the size of the encapsulated data.

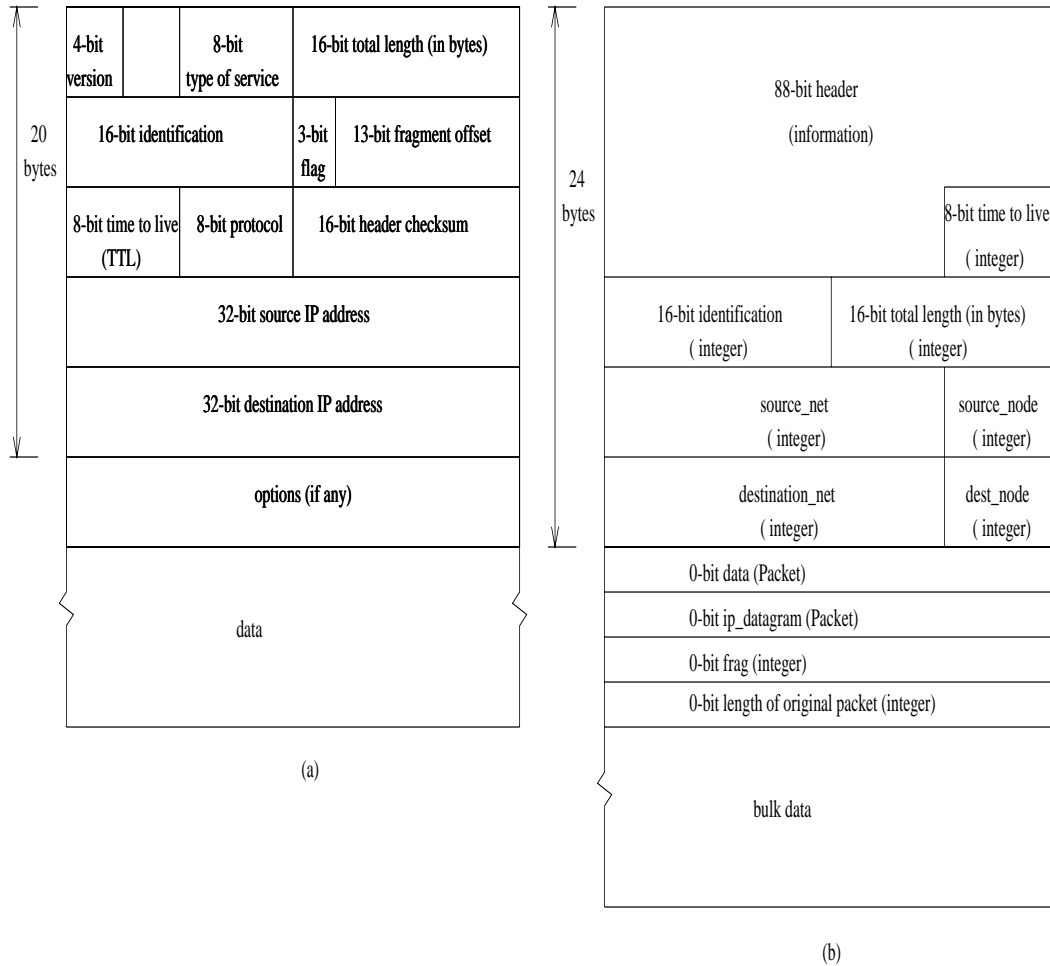


Figure 4: Real IP datagram (a) versus OPNET IP datagram (b)

An IP datagram can face several fragmentations at several nodes of the network. All the packets created during each of those fragmentations have in common the ID of the original packet in their 16-bit iden field and the size of the original packet. Thus when an end-user performs reassembly operation, what he has to do basically when he receives a packet with the flag frag on, is attach the packet according to its ID to an element in the reassembly list, increment a field of that element with the bulk data of the newly received fragment, compare the new value with the size of the original packet and check if the packet contains within one of the two fields *data* and *ip-dgrm* a TCP segment, an IP encapsulated datagram or the original datagram. When those two values match, the reassembly operation is over and the original datagram available.

Network	MTU (bytes)
Hyperchannel	65535
16 Mbits/s token ring	17914
4 Mbits/s token ring (IEEE 802.5)	4464
FDDI	4352
Ethernet	1500
IEEE 802.3/802.2	1492
X.25	576
Point-to-point (low delay)	296

Table 1: Value of the MTU depending on the network

In our model we do not handle the 13-bit field fragment offset as TCP/IP does to perform reassembly. Nevertheless, this field is contained in the 88-bit field header of the OPNET IP datagram. In our model, the processing time and the required resource are underestimated as once a fragment has been received, the receiver can get rid of it once it has taken its size and for the last datagram created the original datagram, whereas in reality the receiver must keep all the fragments received and rebuild the original packet by sorting them in the correct order. Basically, in our model we just take into account the fact that all the fragments of a datagram have been received.

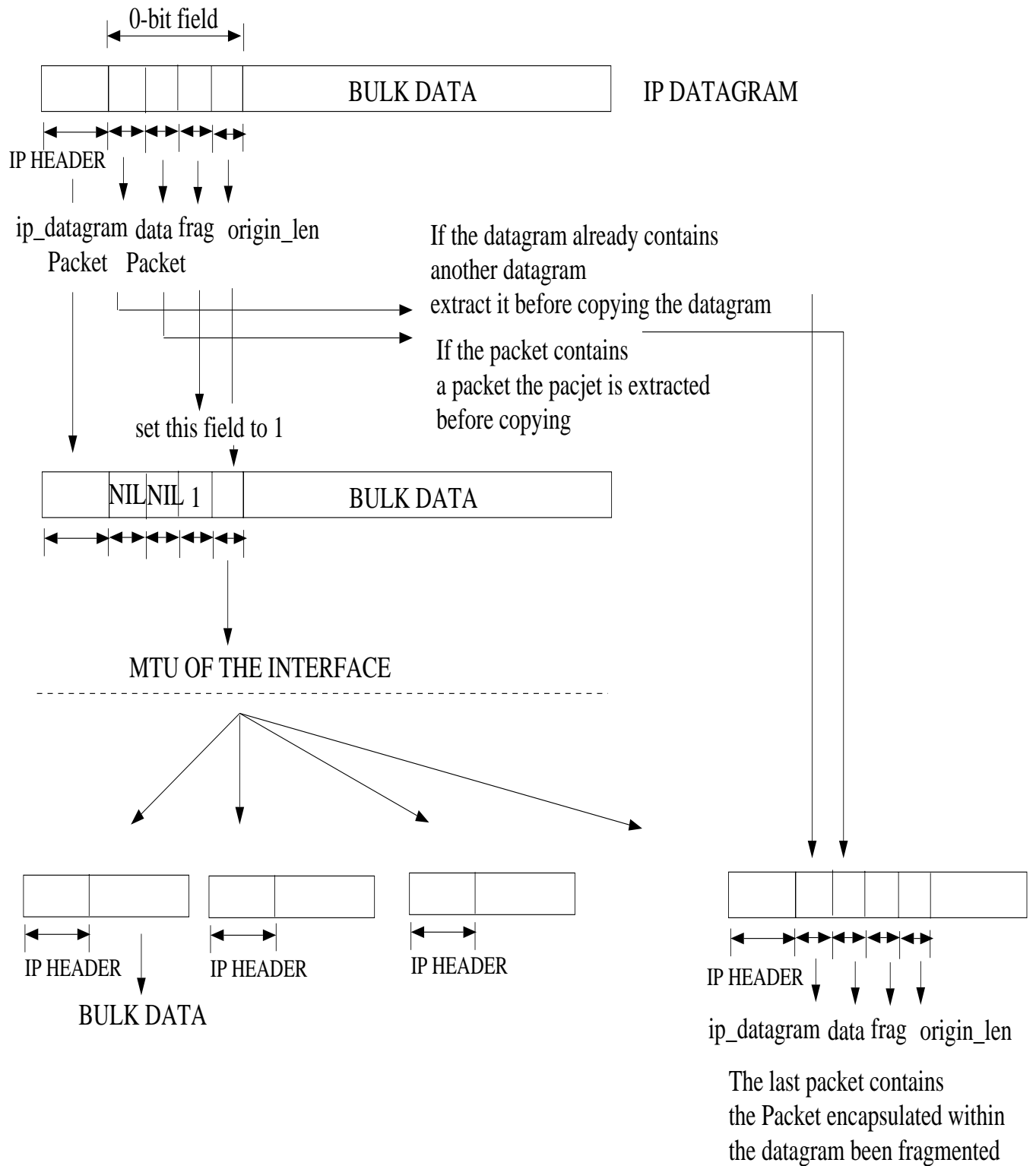


Figure 5: Fragmentation and reassembly schemes within OPNET

2.4 General considerations when modeling a network

When modeling a system, two main concerns drive the modeling scheme in two opposite directions: the fidelity to the reality and the complexity of the models adopted. The actual model results from a trade-off between those two concerns.

Another issue to be addressed is the predictability of the model. There is no point making a simulation whatever the complexity, if it is behaving according to an a priori scheme without introducing some kind of uncertainty. The purpose of a model is not to stick as much as possible to the reality but to model elements that really affect the system, and to see throughout the simulation how they affect the system in various contexts. Otherwise it is worthless modeling elements that are complex but their behavior is completely predictable and effect less influence on the parameter being studied. So in every case, the proximity to the reality must make sense in the particular perspective of a pre-defined study. For example, in our case, fragmentation performed by IP is essential from the point of view of the performance of the Hybrid Internet Access, because the introduction of several packets instead of one adds some processing-time and increases the probability of losing a segment, if we consider a probability of packet-loss for example. So it is necessary to model both fragmentation and reassembly operations.

3 Asymmetric TCP/IP Protocol

As the Hybrid Internet Service was for obvious commercial reasons planned to be used and to work with existing Commercial-Off-The-Shelf TCP/IP stacks in a completely transparent fashion, the satellite channel bandwidth could not be completely used with the standard size of the TCP receive window, currently 4096 kbytes. The satellite channel provides the hybrid host with high transmission rate for the reception of data. But what if this advantage is to be ruined by long-delays experienced over the satellite channel? The maximum achievable throughput with a standard 4096 kbytes receive window size is given by:

$$\text{maximum throughput} = \frac{\text{window size}}{T}$$

where T stands for the time between the transmission of a byte and the transmission of the next 4096th byte. A rapid calculation of this throughput, assuming a value of T equal to 0.250s (which is actually a lower bound to T taking into account only the delay experienced over the satellite channel), gives us a maximum throughput far below the expected capacity of the satellite channel:

$$\text{maximum throughput} = \frac{4096}{\frac{1}{4}}$$

$$\text{maximum throughput} \leq 16\text{kbytes/s}$$

By implementing the Hybrid Internet so that the hybrid gateway acknowledges TCP segments on behalf of the hybrid host, the hybrid host has merely an access to the T1 line on which the hybrid gateway is connected. Moreover, neither the transmission rate of the modem nor the delay over the satellite channel are the limiting factors for the overall throughput. The maximum achievable throughput is then:

$$\text{maximum throughput} = \frac{\text{window size}}{T - (T + T_1)}$$

$$\text{maximum throughput} = \frac{\text{window size}}{T_1}$$

where T_1 stands for the time required by the hybrid gateway for receiving and acknowledging packets before receiving the next 4096th byte. We see that actually the throughput for the hybrid gateway and the hybrid host are merely the same.

The hybrid gateway plays another important role by advertising a larger receive window to the server than the one advertised by the hybrid host. Thus segments of sequence number larger than the maximum sequence number allowed to be sent by the hybrid host can be on the channel. As the delay introduced by the satellite channel is important, those packets are expected once arriving to the hybrid host to be within the receiving window of the hybrid host. Thus the progression of the receive window is anticipated by the hybrid gateway as illustrated in figure 6.

STANDARD VERSION OF TCP/IP
VERSUS
ASYMMETRIC TCP/IP
WHEN EXPERIENCING LONG PROPAGATION DELAYS

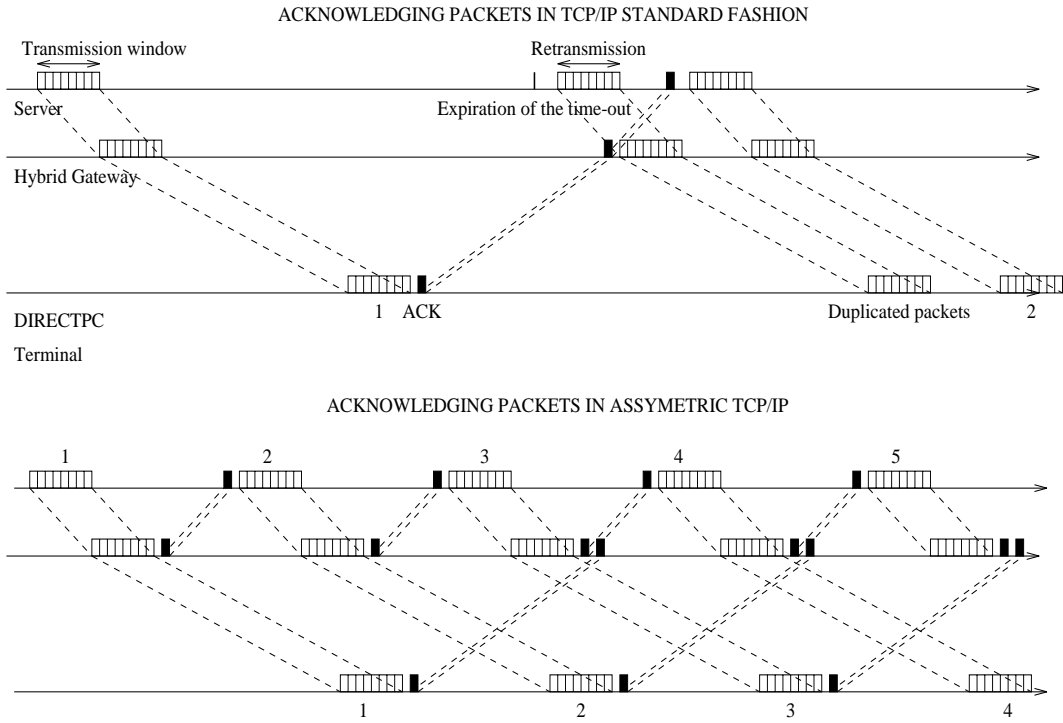


Figure 6: Asymmetric TCP/IP versus standard TCP/IP for acknowledgment policy

These are the two main features of the Asymmetric TCP/IP implemented within the hybrid gateway. What are the implications of these two features?

First, as the hybrid gateway acknowledges segments on behalf of the hybrid host, it must keep the segments received in sequence within its window and not yet acknowledged by the hybrid host, and consequently must deal with their retransmission until it has been made sure that they have been received by the hybrid host; that is until they have been acknowledged by the hybrid host. So large memory capacities are required in the hybrid gateway, on one hand to keep a record of all the packets sent unacknowledged, and on the other hand to keep track of several control data such as the current receive window of the server, the first unacknowledged sequence number, etc.

As the attribute “asymmetric” of the protocol implies, the management of the connection

by the hybrid gateway is not the same in the two directions of data streams. The hybrid gateway plays an active role in the downstream toward the hybrid host, whereas it is more passive in the upstream toward the server. The hybrid gateway acts as some kind of a representative (proxy) of the hybrid host in the Internet and helps integrate the hybrid host to the Internet. The hybrid gateway is the last node in the standard (terrestrial) network for packets before entering the hybrid part of it.

4 Implementation of the Asymmetric TCP/IP protocol within the hybrid gateway

4.1 Processing of segments arriving from the server

When the IP layer of the hybrid gateway receives a packet addressed to a hybrid host, it decapsulates it and sends the decapsulated TCP-segment to the Transport layer. Once the connection this segment belongs to has been identified in the Transmission Connection Block (TCB), if the segment fits in the receive window, the hybrid gateway has to send it to the hybrid host, and then to enqueue it in its unacknowledged segments buffer. At the same time, the hybrid gateway has to pick the value of the advertised receive window of the server. It will need it when retransmitting segments in order to fill the control field of the TCP segment with the current value. Thus the retransmission process is done in a transparent fashion to the hybrid host Transport layer. Therefore, we see that the hybrid gateway performs essential functions as flow control, retransmission process, i.e. reliable delivery of segments, etc.

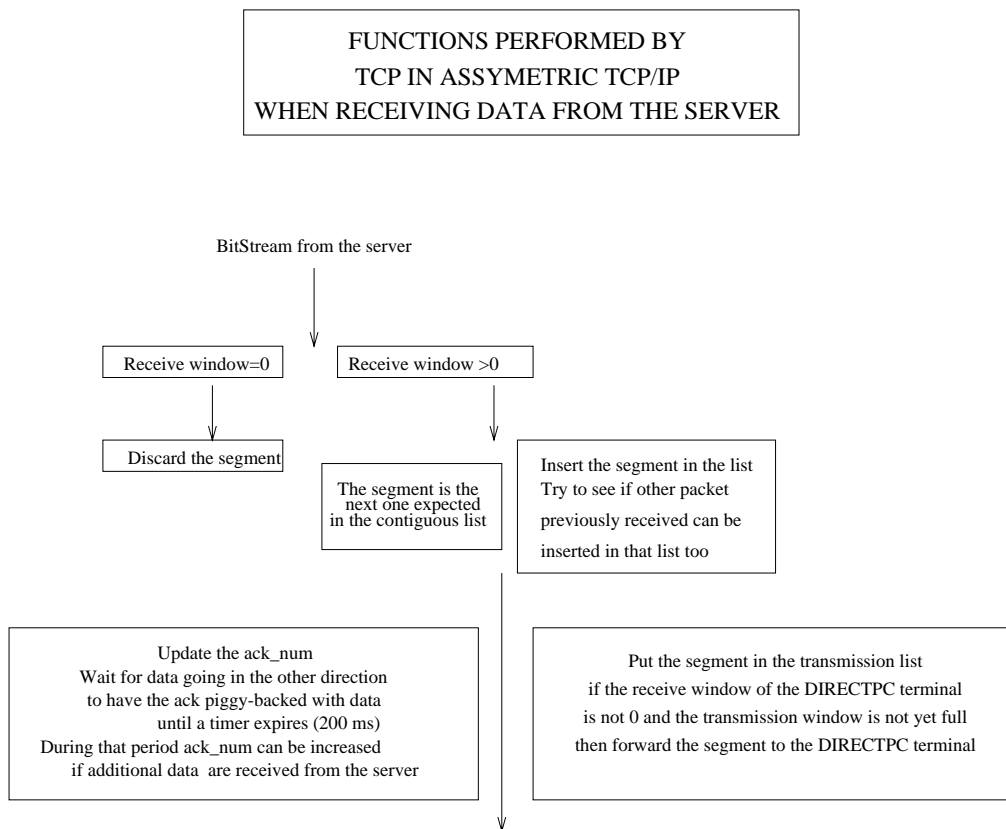


Figure 7: Processing of segments arriving from the server

4.2 Processing of segments arriving from the hybrid host

The hybrid gateway only forwards segments coming from the hybrid host to the server. However, it has to pick the acknowledge number, flush the data covered by this acknowledge number and replace it by its current acknowledge number, which must be higher. Thus this function performed by the hybrid gateway is completely transparent to the server. The hybrid gateway has to keep track of the window advertised by the hybrid host, but as a key point of the Asymmetric TCP/IP protocol regarding hybrid networks it is not required to conform to it in all cases. A trivial case for which it is mandatory to respect the size of the receive window is when the hybrid host advertises a zero window size.

4.3 The establishment of the connection

We now describe how connection establishment is addressed by the hybrid gateway. Besides the management of the connection between the hybrid host and the server according to the Asymmetric TCP/IP protocol, the hybrid gateway must go through the three-hand shake protocol for the establishment of the connection between the server and the hybrid host. Unlike when the connection is fully established, the hybrid gateway is not required to take any active part in the initiation of a connection and it is not required to acknowledge a SYN segment on behalf of the hybrid host as it currently does for common segments when the connection is fully established. The hybrid gateway must keep track of the control data passing through it during the initiation of the connection and decide when the connection is established; that is when both the server and the hybrid user's TCP connection Finite State Machine have moved to the state ESTABLISHED. The different segments going through the hybrid gateway during this stage are SYN alone, SYN and ACK of SYN, ACK of SYN alone, ACK of SYN and data segments. This phase is critical from the point of view of the connection. Indeed, many parameters of the connection must be initialized during this phase, as the ISS, Initial Sequence Segment, of the server. So even if the hybrid gateway doesn't play any active role but instead leaves the server and the hybrid host alone to establish the connection, it has to keep track of several parameters in order to be ready to play its required active role once the connection is fully established.

During the entire phase of connection establishment, the hybrid gateway, unlike the end-user and the server, remains in the same state and just updates the value of two variables according to the segment received. The first variable value changes when the first SYN issued by either the server or the end-user passes through the hybrid gateway. The first time the hybrid gateway receives a SYN, it opens a new entry in its Table of Connection Block (TCB) and gets ready to manage the new connection. The variable SYN_RECEIVED moves from 0 to 1. There are two variables named SYN_RECEIVED: one for the server and another one for the hybrid host. The second value to be considered concerns the ACK of the SYN that each end of the connection must send to the other one. Once the hybrid gateway receives from one end of the connection a segment containing an ACK acknowledging the SYN sent by the other end of the connection, the hybrid gateway changes the second variable CONNECTION_ESTAB of the end of the connection sending the segment to ON. The hybrid

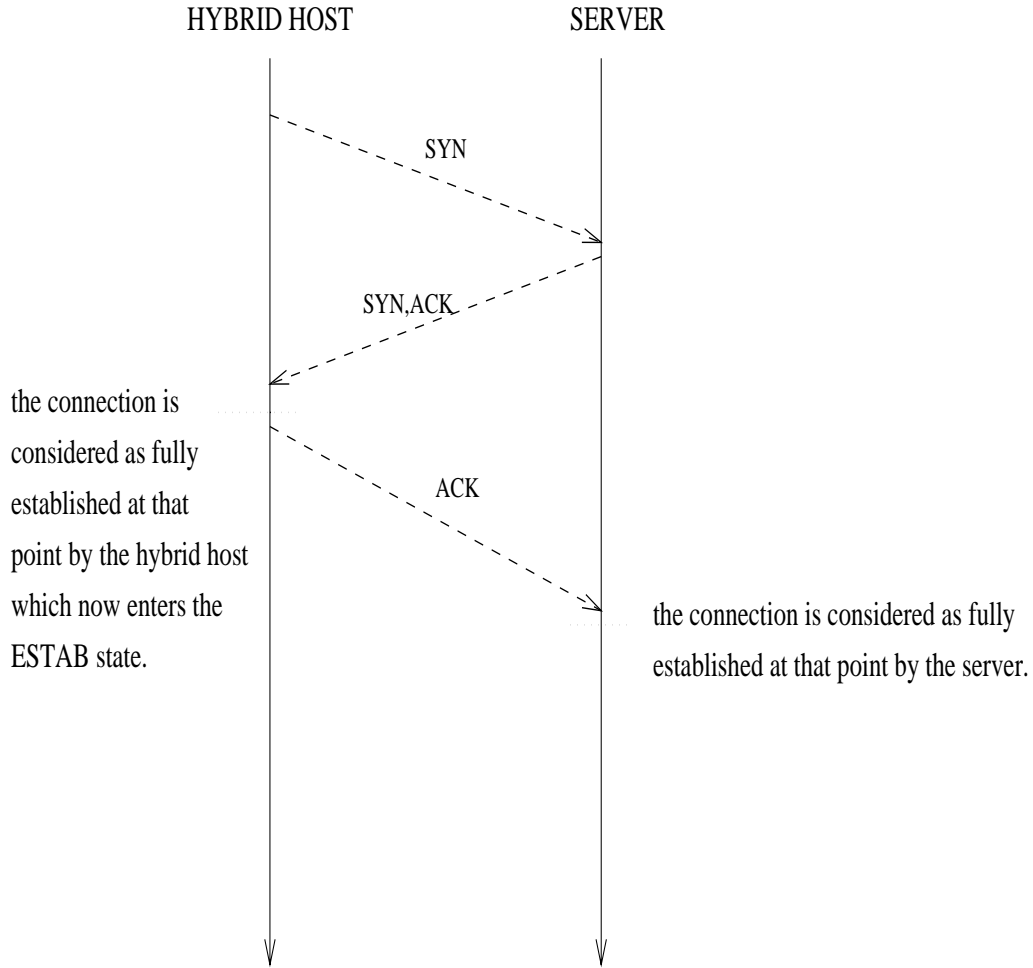


Figure 8: Three hand-shake protocol

gateway knows that this end of the connection has moved in the ESTAB state, but still has to wait for the time when both ends of the connection are in that state to be able to move itself in that state where the connection is considered as fully established. We see that unlike the two ends of the connection which move from one state to another during the establishment of the connection, all that the hybrid gateway does is interpret the arriving segment and update the value of variables that describe in what state the sender of the current segment is. That provides a non-ambiguous way to determine the very moment when the two ends of the connection are for the first time both in the state ESTAB. Only then the connection is considered as fully established.

5 Structure of the model

In this section we provide schematics of the OPNET modules of the model and their interconnections/structure. These schematics provide the details of the structure of the model.

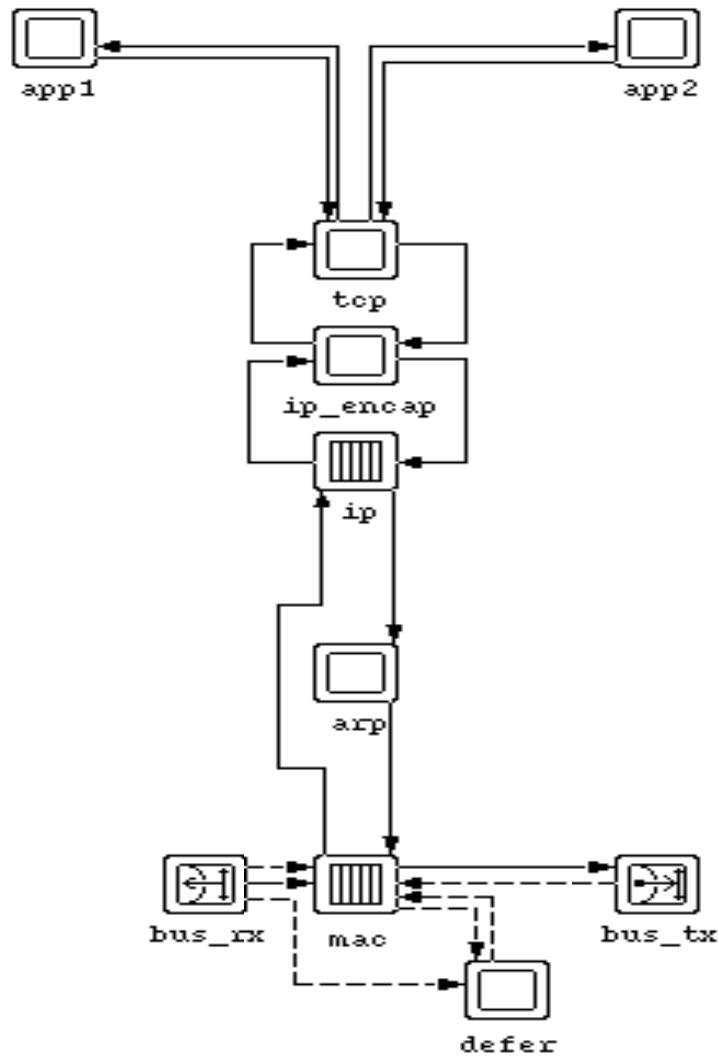


Figure 9: Server working over a LAN

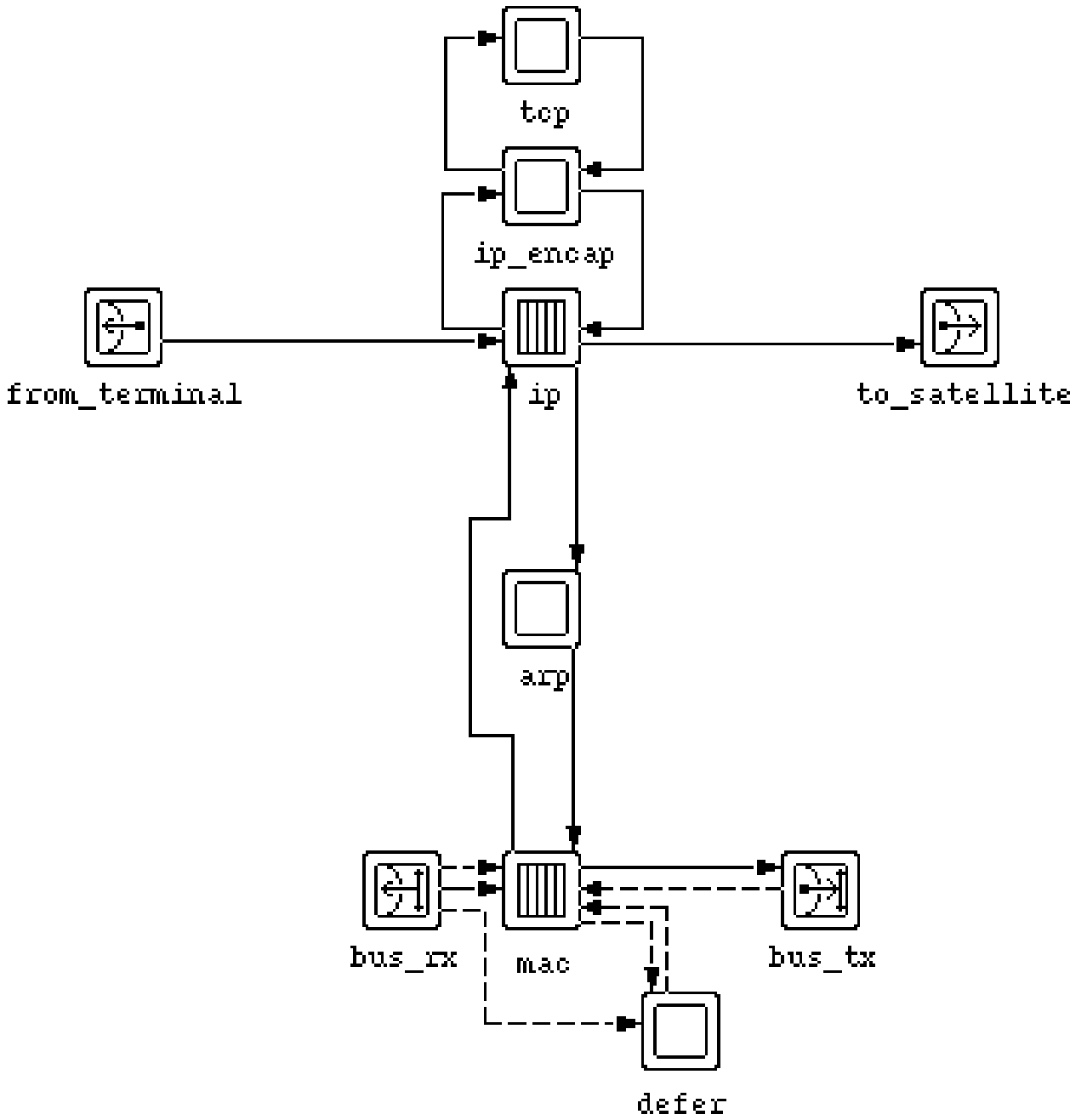


Figure 10: Hybrid Gateway

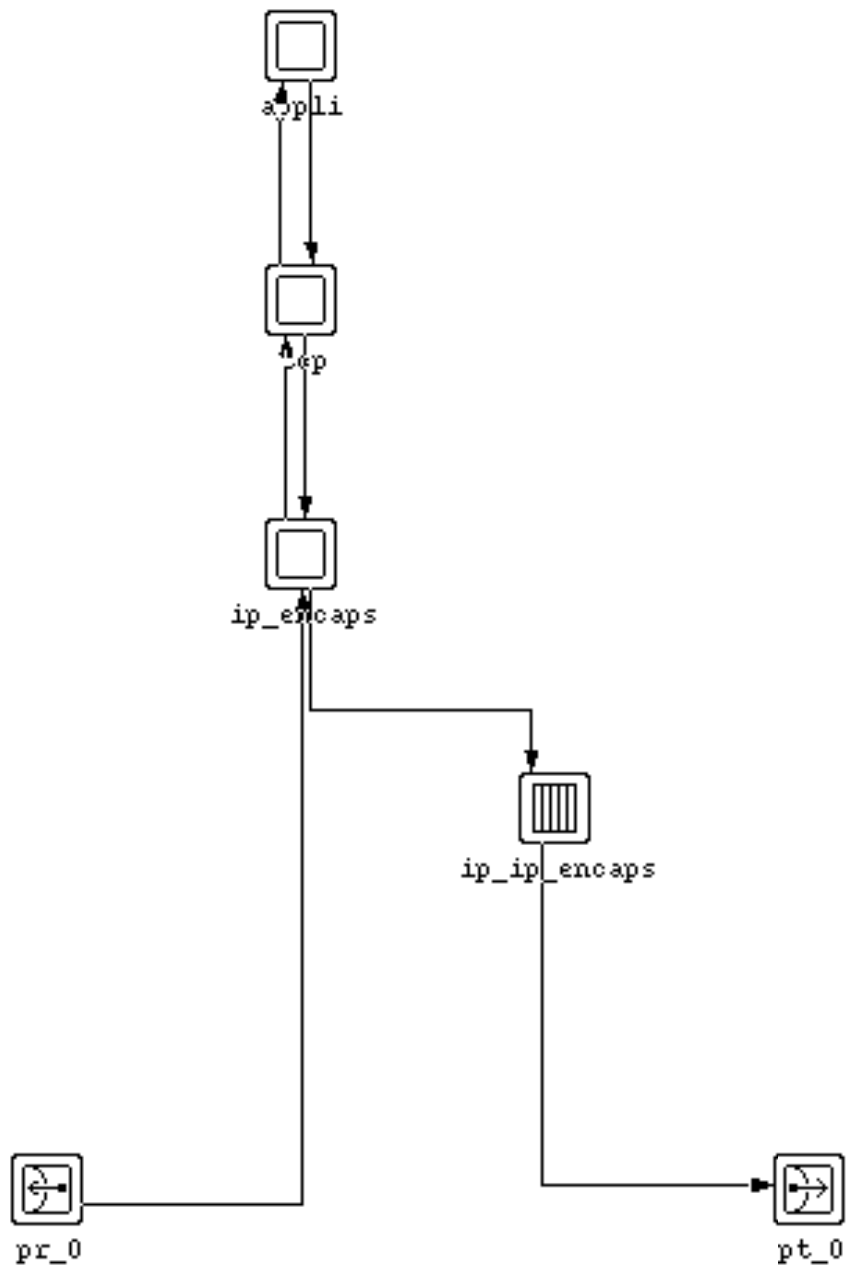


Figure 11: Hybrid Terminal

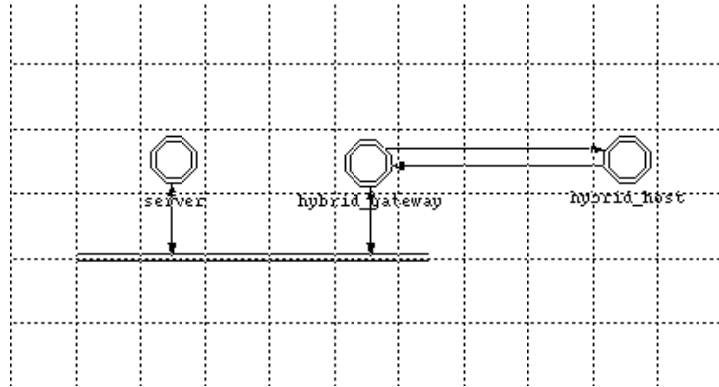


Figure 12: Hybrid Network

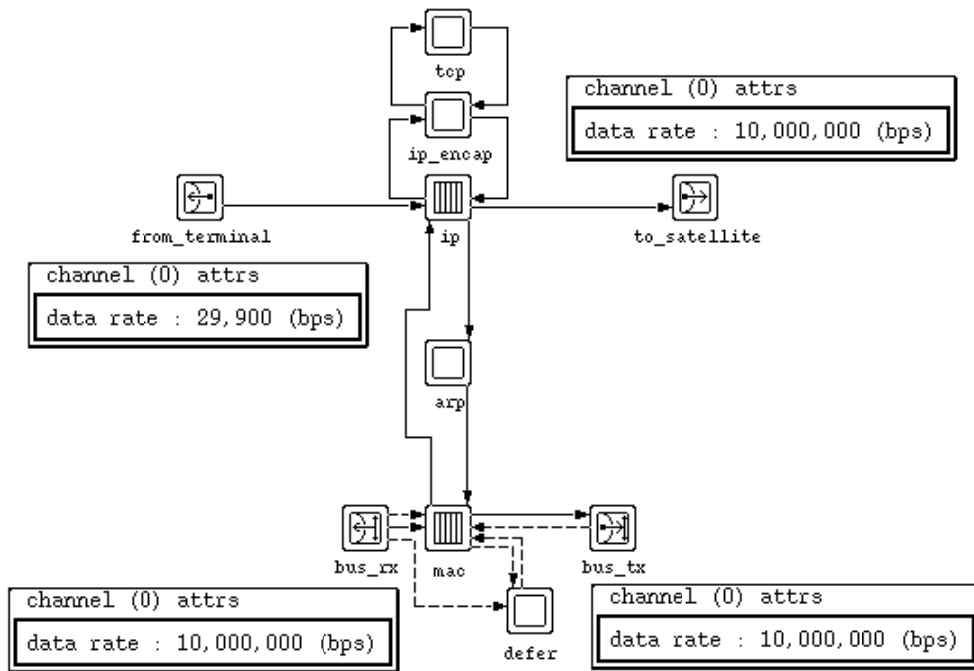


Figure 13: Capacity of the channels at the hybrid gateway

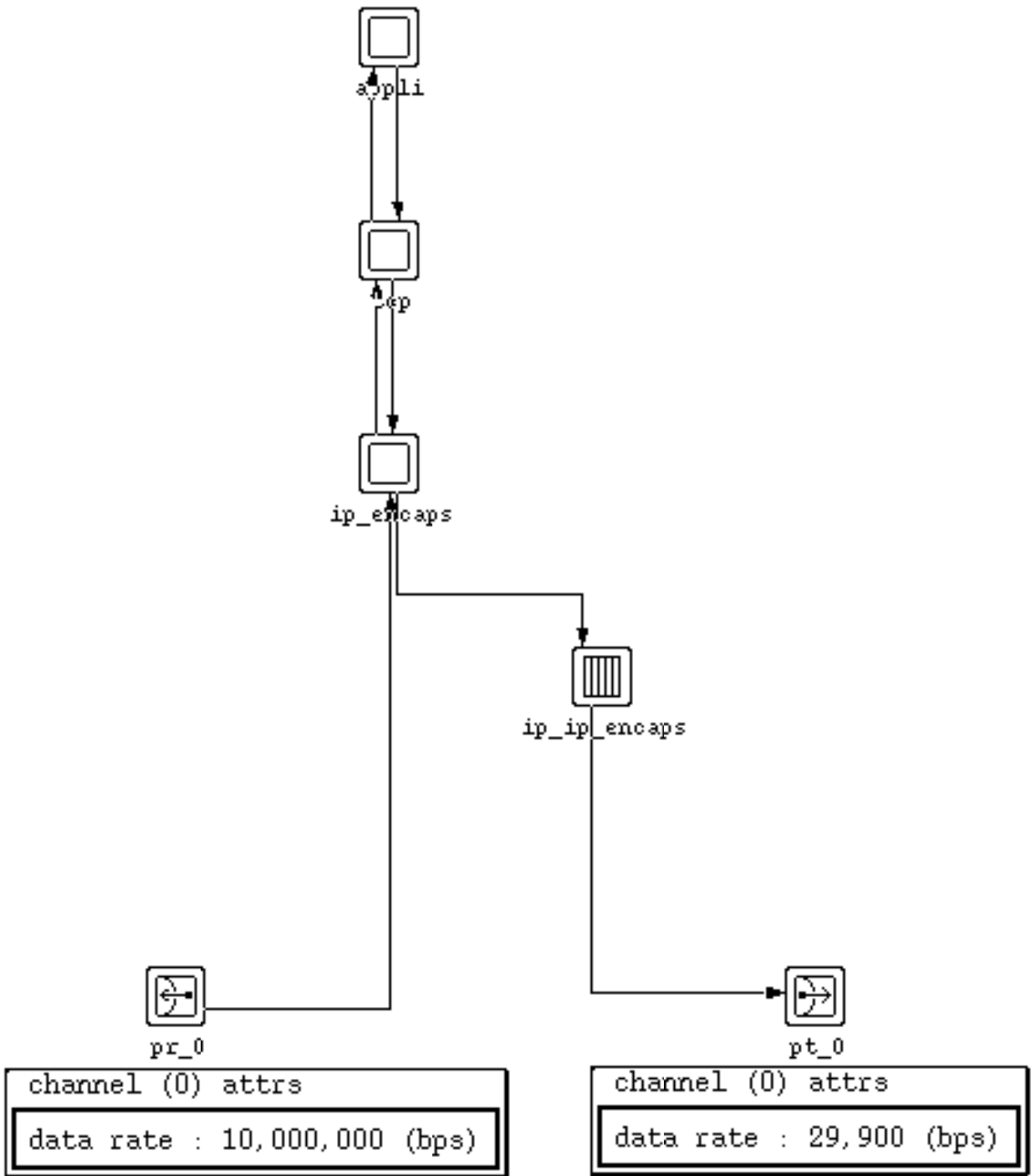


Figure 14: Capacity of the channels at the hybrid terminal

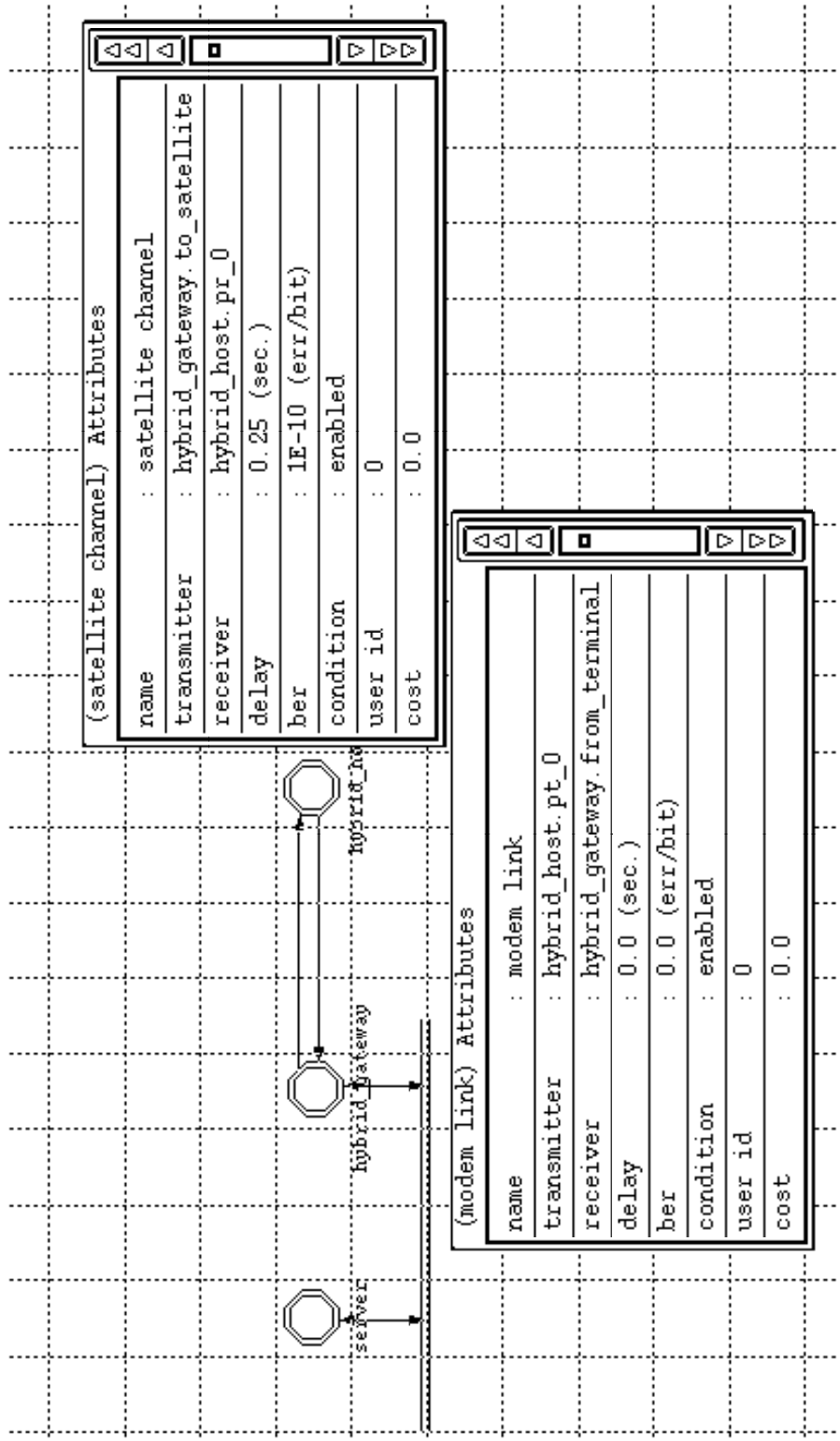


Figure 15: Downstream and upstream channels in the hybrid network

6 Results of the simulation

OPNET allows us to keep track of the evolution of several attributes throughout the simulation. A particular attribute of interest must be loaded in a probe which has to be assigned to the simulation file just before the simulation is launched. The default model of TCP/IP provided within OPNET gives the statistical outputs described in Table 2 below.

Name of the macro	numerical index	parameter being studied
TCPC_OUTSTAT_DELAY	0	End-to-end delay of the received segment
TCPC_OUTSTAT_RCV_SEG_SEQ	1	Sequence number of the received segment
TCPC_OUTSTAT_RCV_SEG_ACK	2	ACK sequence number of the segment received
TCPC_OUTSTAT_SND_SEG_SEQ	3	Sequence number of the segment being sent
TCPC_OUTSTAT_RCV_SEG_ACK	4	ACK sequence number of the segment being sent
TCPC_OUTSTAT_CWND	5	Congestion window
TCPC_OUTSTAT_RTT	6	Measured round trip time
TCPC_OUTSTAT_RTT_MEAN_DEV	7	Measured mean deviation of round-trip time

Table 2: Statistical outputs in the TCP process

We added the window advertised by the hybrid host and the amount of data the hybrid gateway can send each time it sends some data. By defining several probes, we can vary the values of key parameters and then analyze the implication on the throughput and other performance metrics of interest such as delay. We varied values of three parameters:

- the bit-rate of the modem used at the hybrid host to send back data;
- the value of the default window advertised by the hybrid host;
- the rate at which the application sends packets.

The study was highly motivated by the need to validate that as a result of the hybrid network and the asymmetric implementation one could have a difference between the throughput at the server and at the hybrid host that would match the one observed in reality. DirectPCTM claims to achieve 40 times better rate than standard telephone modem only Internet connection type. What this claim means is that DirectPCTM has been verified to reach data rates at the satellite dish of about 400 kbits/s which is 40 times higher than what can be achieved via a 9,600 bits/s modem under the best conditions. The 9,600 bits/s is a physical limit to the throughput achievable at the end user side and in the overall network constitutes a bottleneck for the end user when downloading data from the Internet. On the contrary, the rate achieved by the hybrid network (and the asymmetric implementation) is not a physical limit and stands well beyond the physical limit imposed by the satellite channel, which is 10 Mbits/s. Building the simulation model was highly motivated by these objectives. The validation of the model and the simulation are important parts of the continuing work on Internet over Satellite at the CSHCN.

6.1 Validation of the model

Even when the model works properly and generates a lot of performance curves, the model must still be questioned. There are a lot of mechanisms at stake in the model, TCP/IP, Asymmetric TCP/IP, not to mention all the lower layers working transparently to the transport layer we are focusing on. After having experienced the modularity of the TCP/IP stack according to the OSI model, we had to be concerned by the inter-dependence of the different modules when the model is at work. For instance, when interested in the throughput at the transport layer, we must keep in mind where other bottlenecks can occur in other layers. In our study, the IP module processes packets at a service rate specified as a promoted parameter, i.e. with a value specified in a file at the simulation level. Modeling a service rate adds to the realistic aspect of the simulation but should not alter the model by assigning to the service rate a value that would be a bottleneck. So it is of no concern as long as the value does not affect the parameter studied.

We were interested in two kind of results: those that would validate the TCP/IP model and those concerning directly the performance of the Asymmetric Internet in the hybrid network. One part of the results is dealing with validation, the other with modeling. Only the second part is of some interest from an external point of view taking for granted that the model works and that it is validated.

The curves obtained highlight the commonly noticed phenomenon of the SLOW-START over the throughput. The curve has a typical parabolic shape.

6.2 Guidelines for the simulation

The model we set up had to work with a large set of different values assigned to key parameters. We identified the key parameters to test with different values as the following:

- the overall application load, which depends on the interarrival rates of packets to be sent to the Transport layer along with the size of the packets; as those two values are expected to follow a random distribution, the overall load can be expressed as:

$$\frac{E(\text{mean-size-of-a-packet})}{E(\text{interarrival-time})};$$

- the window size advertised at the server, hybrid gateway and hybrid host;
- the default retransmission time-out at the hybrid gateway;
- the speed of the modem used by the hybrid host, i.e. the bit rate of the link back to the hybrid gateway.

These parameters were given three kind of values:

- the standard values found in the literature;
- the values that match the specifications of the Hybrid Internet Service;
- arbitrary large values in order to test the robustness of the model and to get the maximum output for the key parameter queried as the throughput at the end-users. For example by assigning arbitrary large values for the window advertised by the gateway in a completely unrealistic fashion, we get the maximum achievable throughput from the server within our model, which gives us an upper bound to the expected throughput at the hybrid user.

6.3 Results and parametric curves

6.3.1 First set of results with standard parameters

The first simulation was performed with a standard value for the window advertised by the hybrid host: 4096 bytes. The window advertised by the hybrid gateway to the server was set to 64000 bytes. As shown in figure 16, the throughput for incoming data is not satisfactory at both end-users. The Asymmetric TCP/IP protocol plays its role as the upstream data load is 7 Megabytes in 100 seconds. Therefore the large window advertised by the hybrid gateway to the server allows the server to send a large amount of data which is still only 7% of the overall application load. The windows advertised by the server and the hybrid host are both 4096 and still, even with those values we can notice that the throughput at the hybrid host is slightly higher than the throughput at the server and thus that Asymmetric TCP/IP protocol again works the way we wanted.

parameter	value assigned
hybrid_host RCV_BUFFER	4096
hybrid gateway RCV_BUFFER	64 kbytes
server RCV_BUFFER	4096 bytes
hybrid gateway retransmission timeout	200 ms
modem speed	9,600 bits/s

Table 3: Values of the standard parameters

Even if this set of curves is of some interest because they do confirm certain mechanism at work in the Internet, the results are still not satisfactory. The window advertised by the hybrid host to the hybrid gateway has to be assigned a greater value along with the speed of the modem.

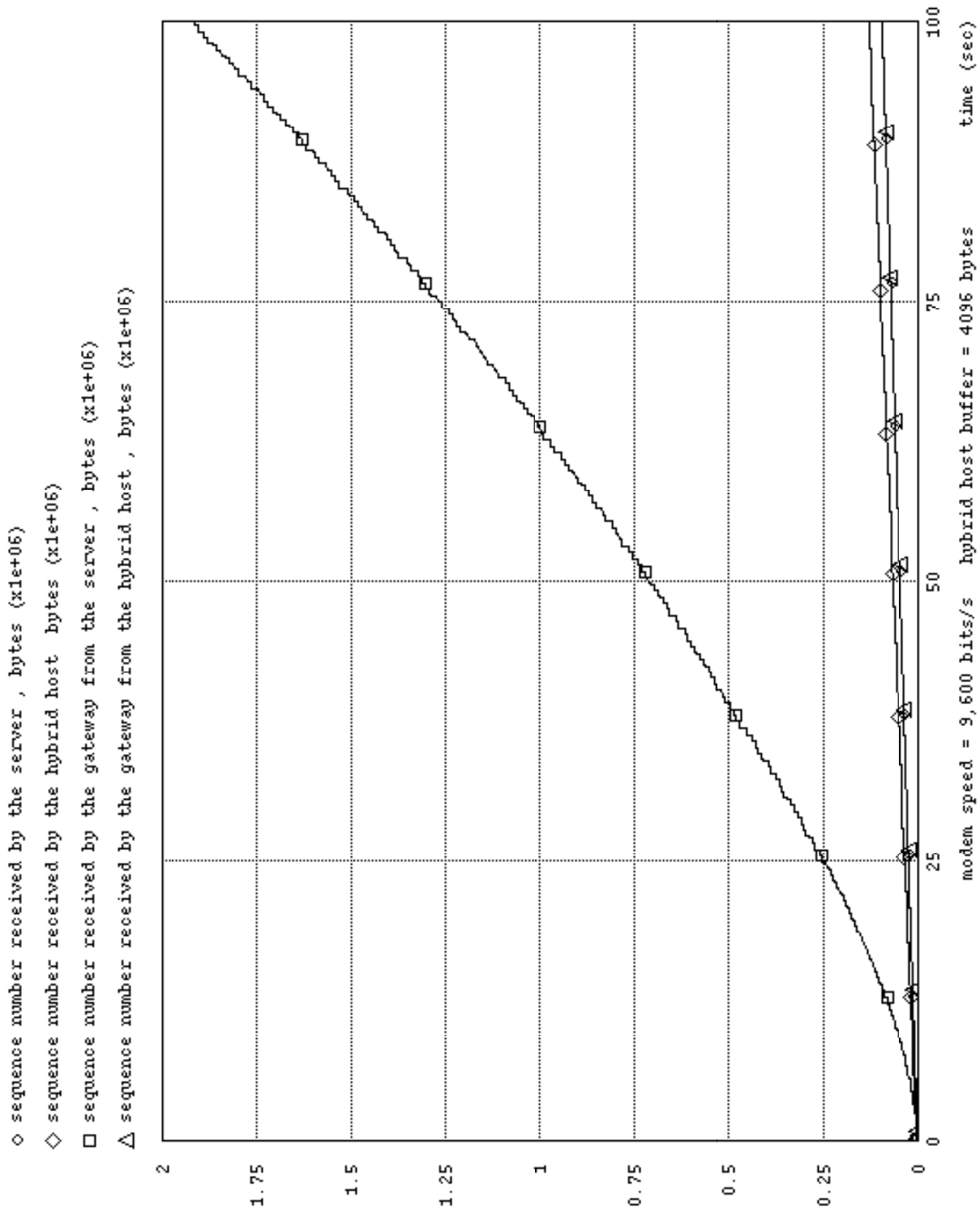


Figure 16: Throughput at the end-user and at the hybrid gateway

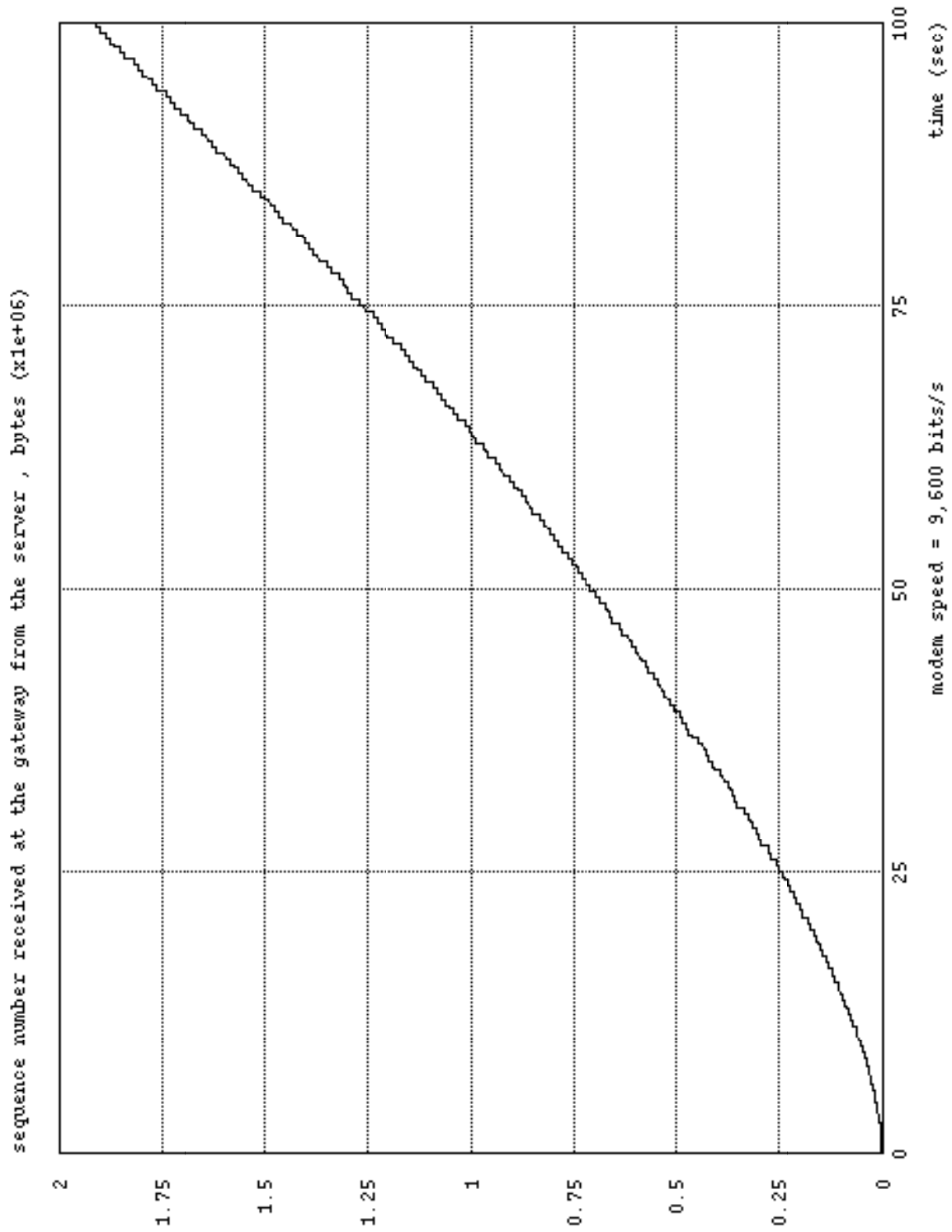


Figure 17: Downstream data load

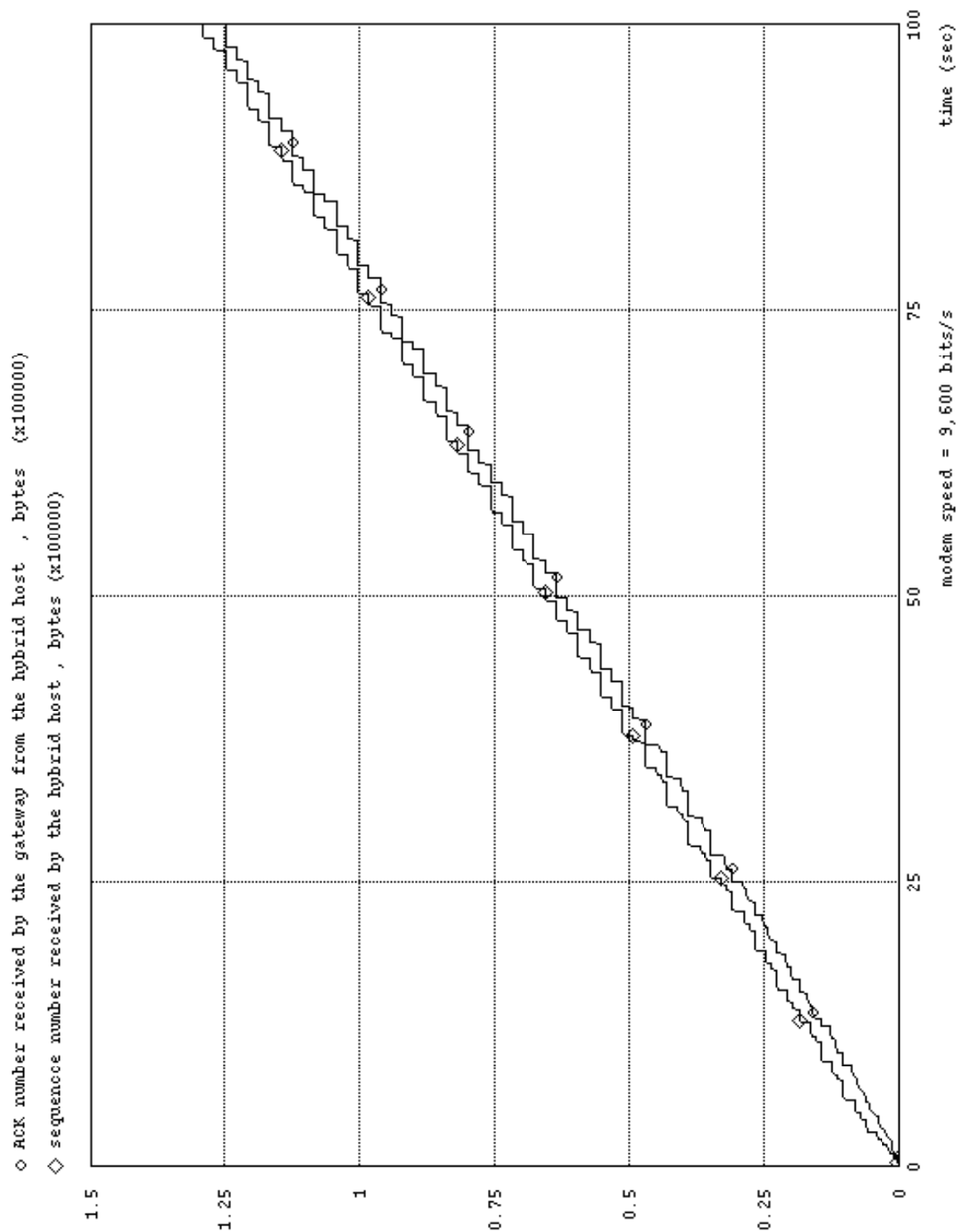


Figure 18: Sequence number received by the hybrid host and ACK number received at the hybrid gateway

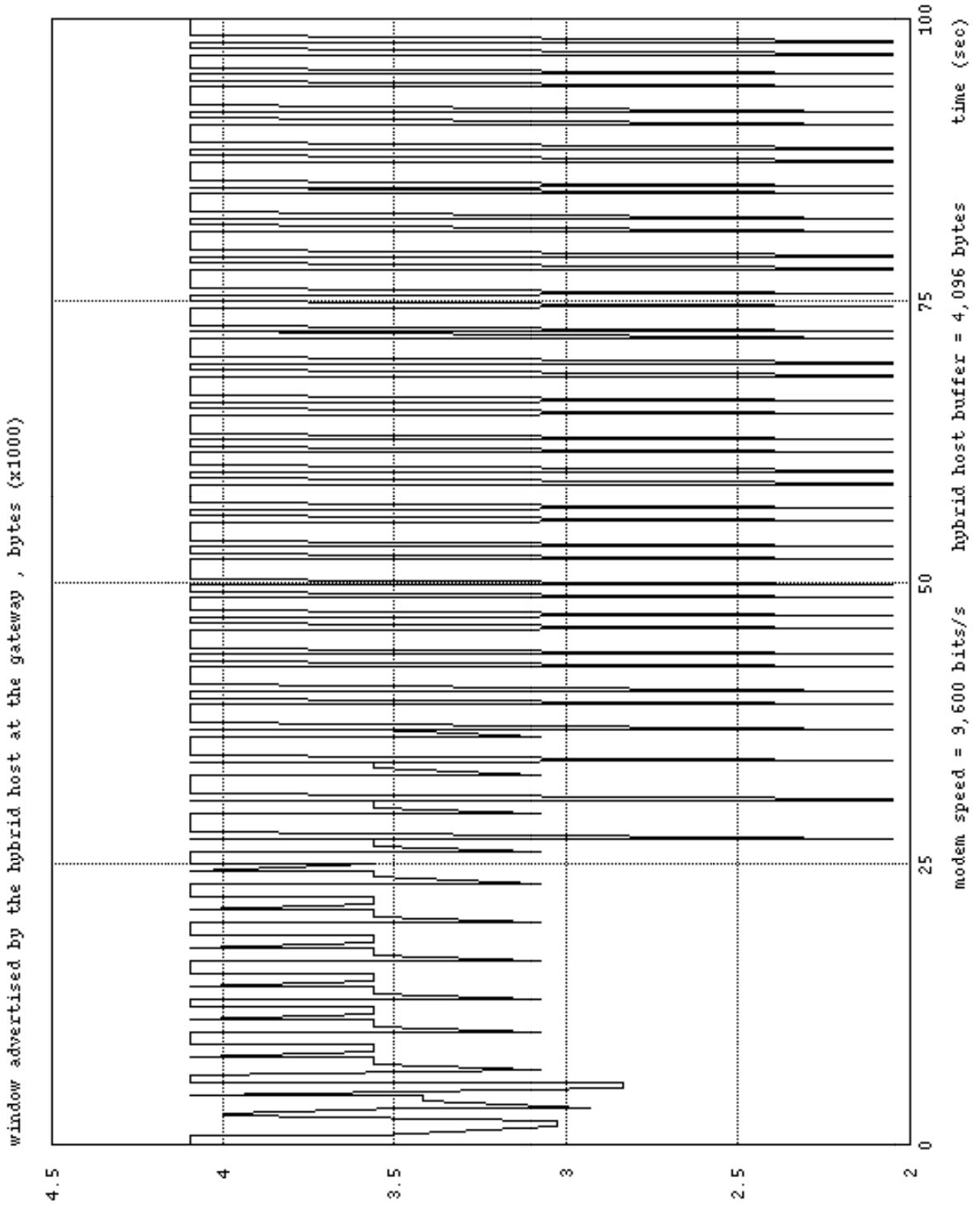


Figure 19: Window advertised by the hybrid host at the hybrid gateway

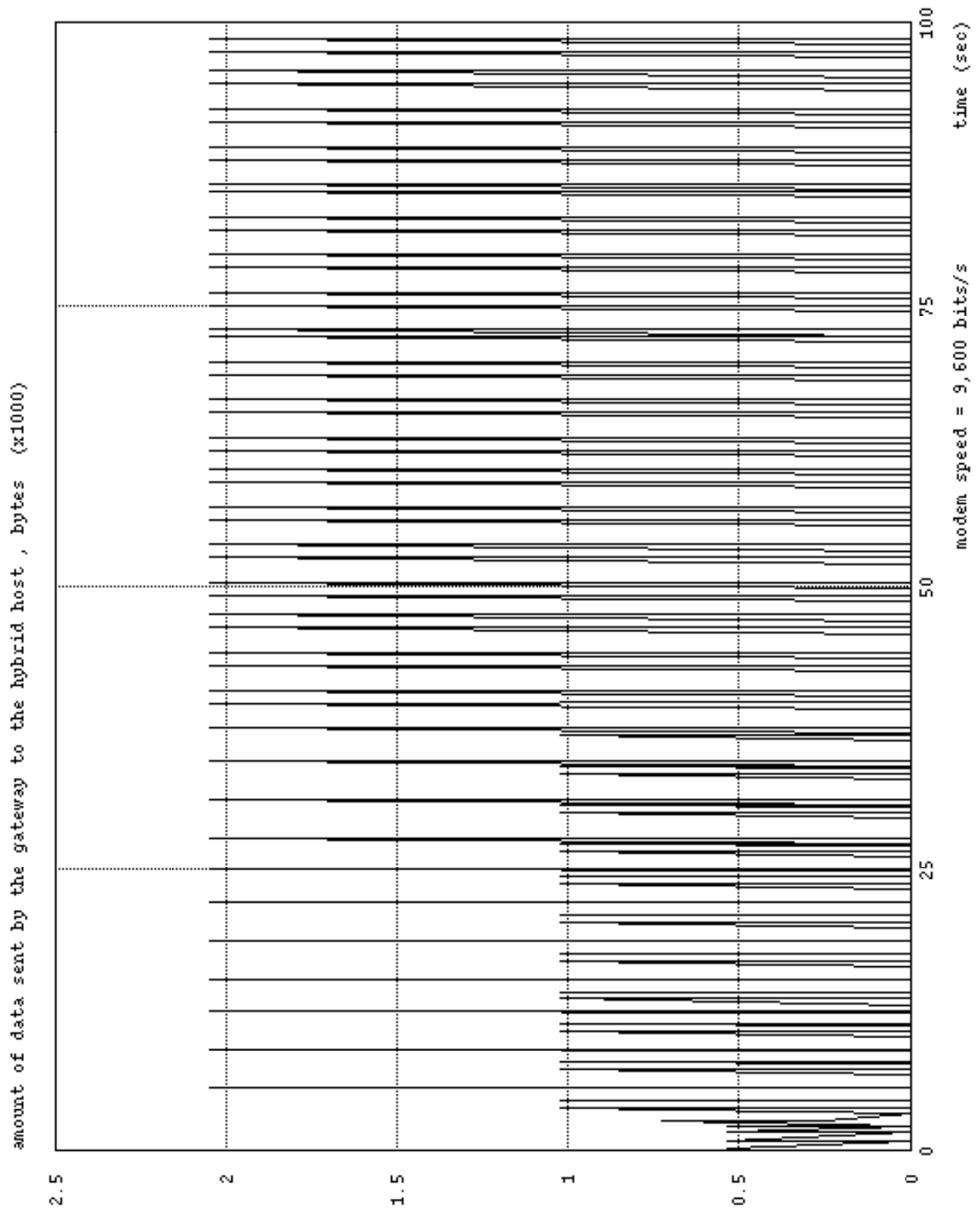


Figure 20: Amount of data sent by the hybrid gateway to the hybrid host

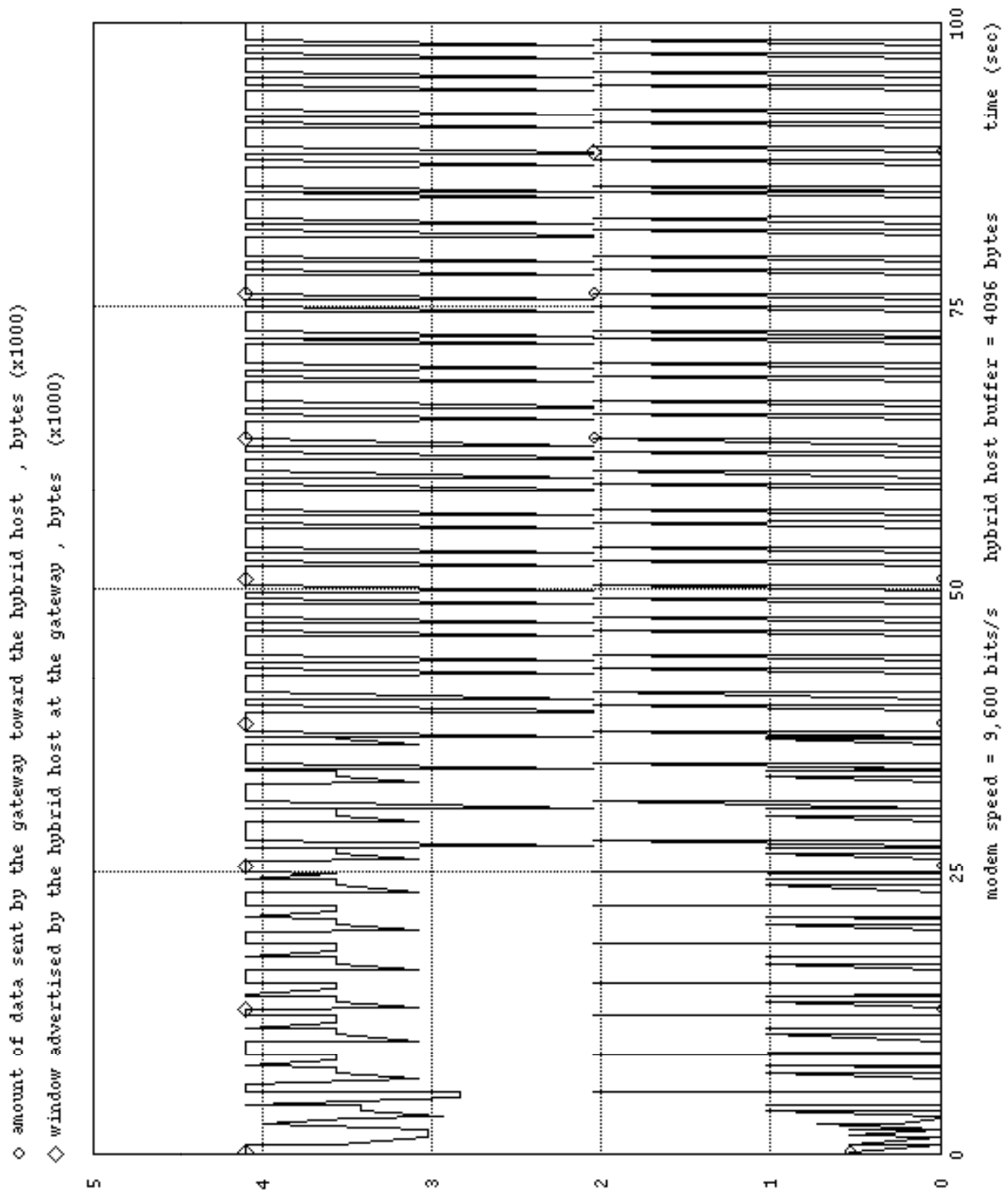


Figure 21: Comparison between the window size and the amount of data eligible to be sent by the hybrid gateway

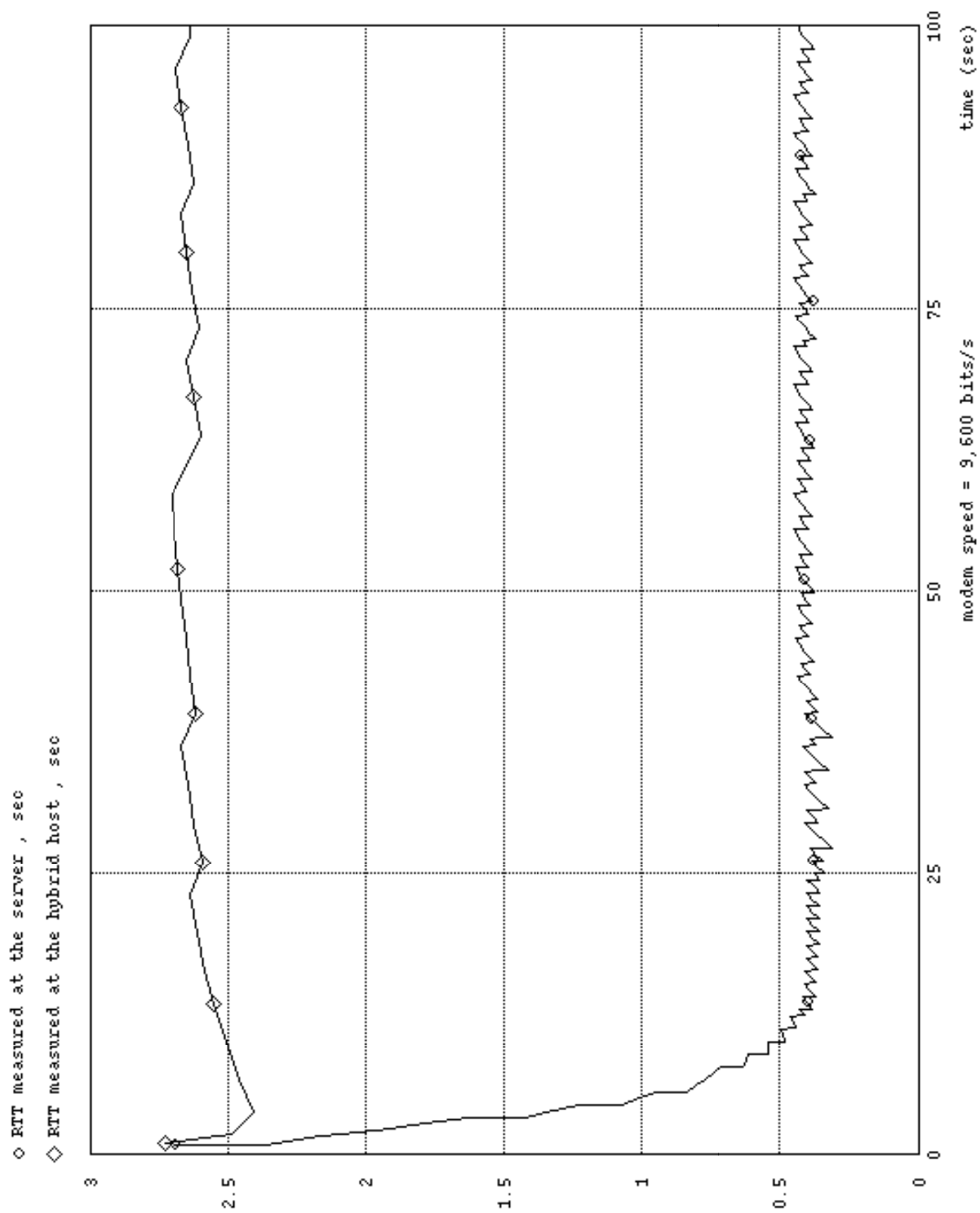


Figure 22: RTT measured at the server and the hybrid host

6.3.2 Results with a higher speed modem

parameter	value assigned
hybrid_host RCV_BUFF	45 kbytes
hybrid gateway RCV_BUFF	64 kbytes
server RCV_BUFF	4096 bytes
hybrid gateway retransmission timeout	200 ms
modem speed	29,900 bits/s

Table 4: Values of the standard parameters with a high speed modem

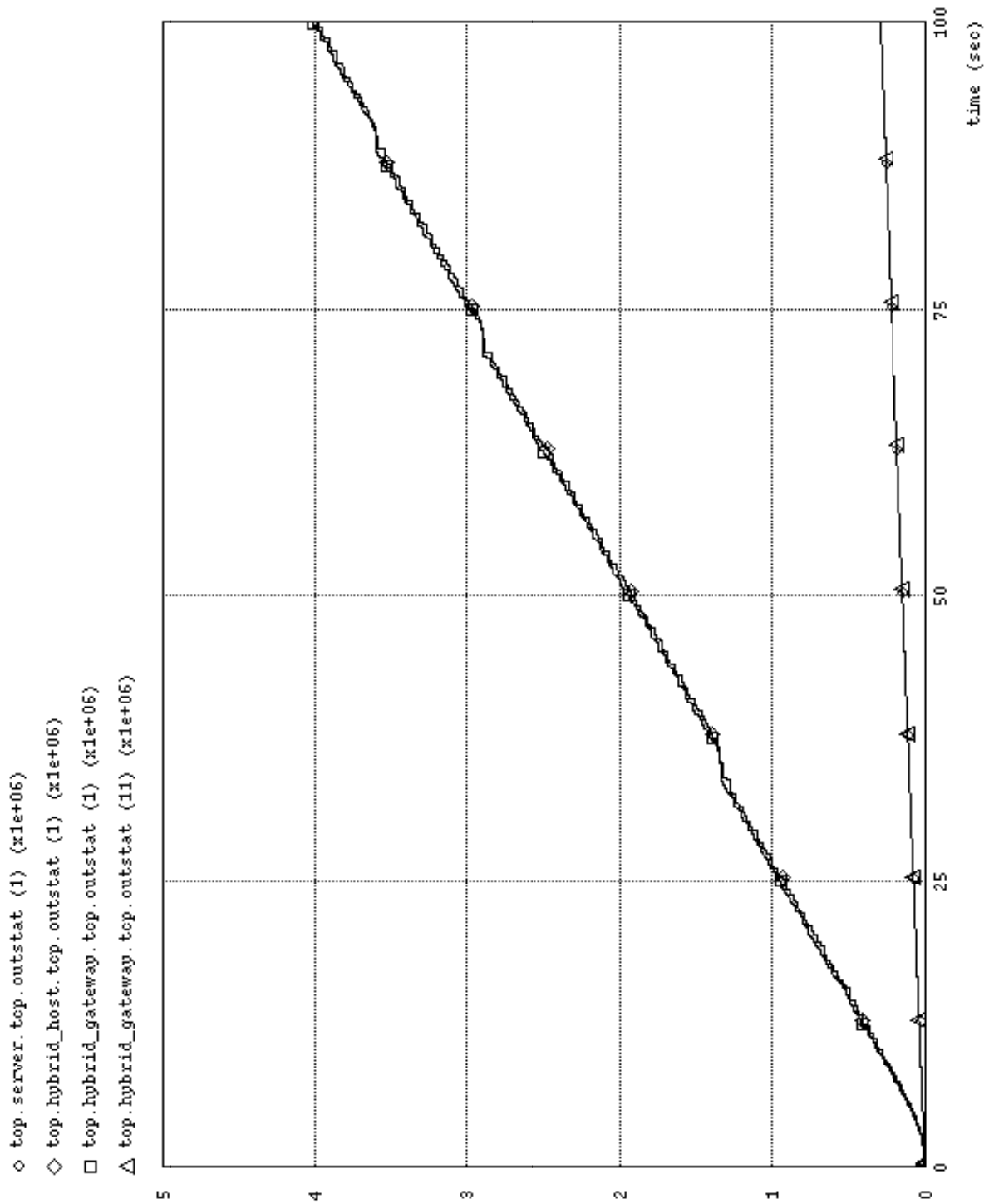


Figure 23: Throughput at the end-user and at the hybrid gateway

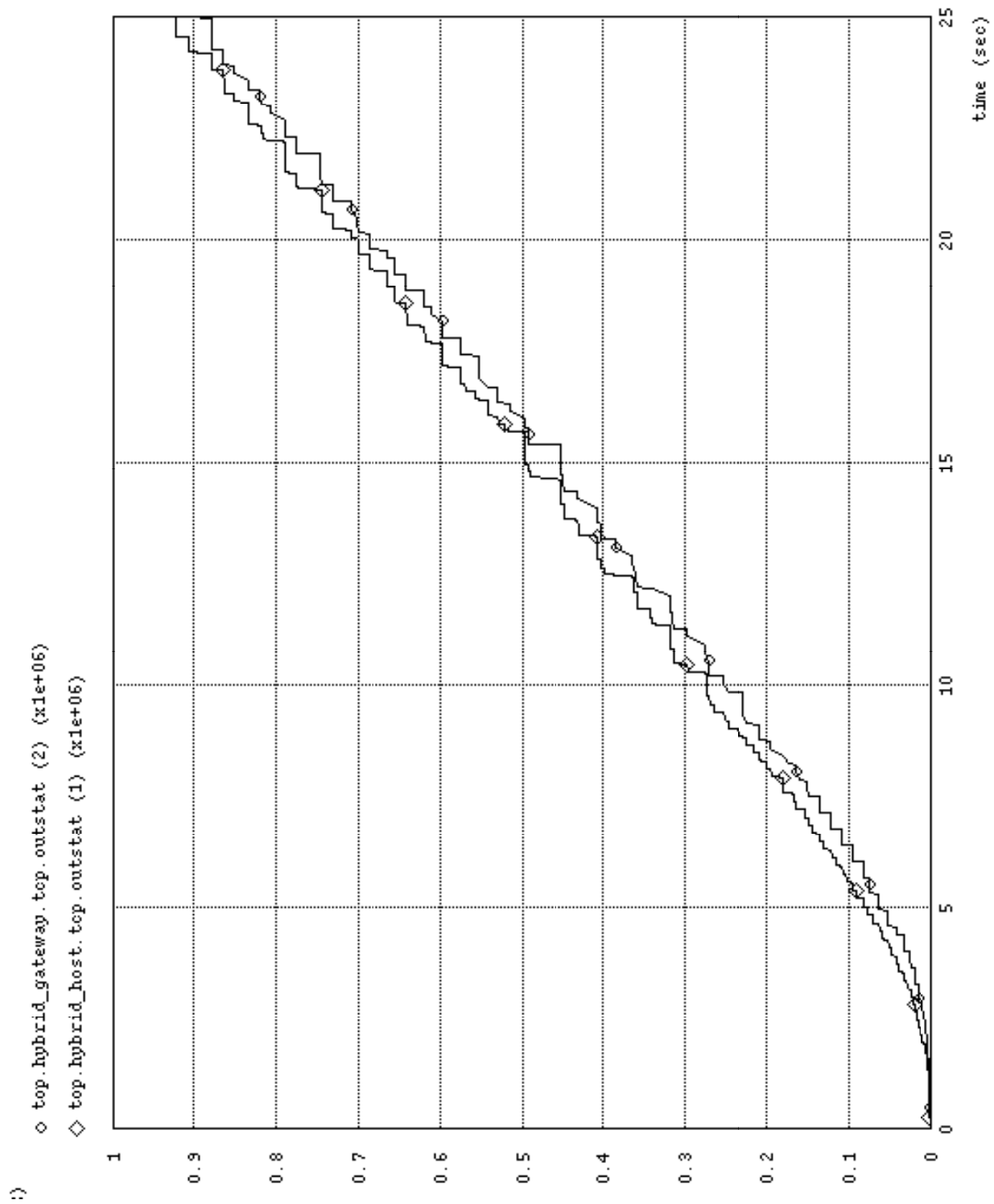


Figure 24: Sequence number received by the hybrid host and ACK number received at the hybrid gateway

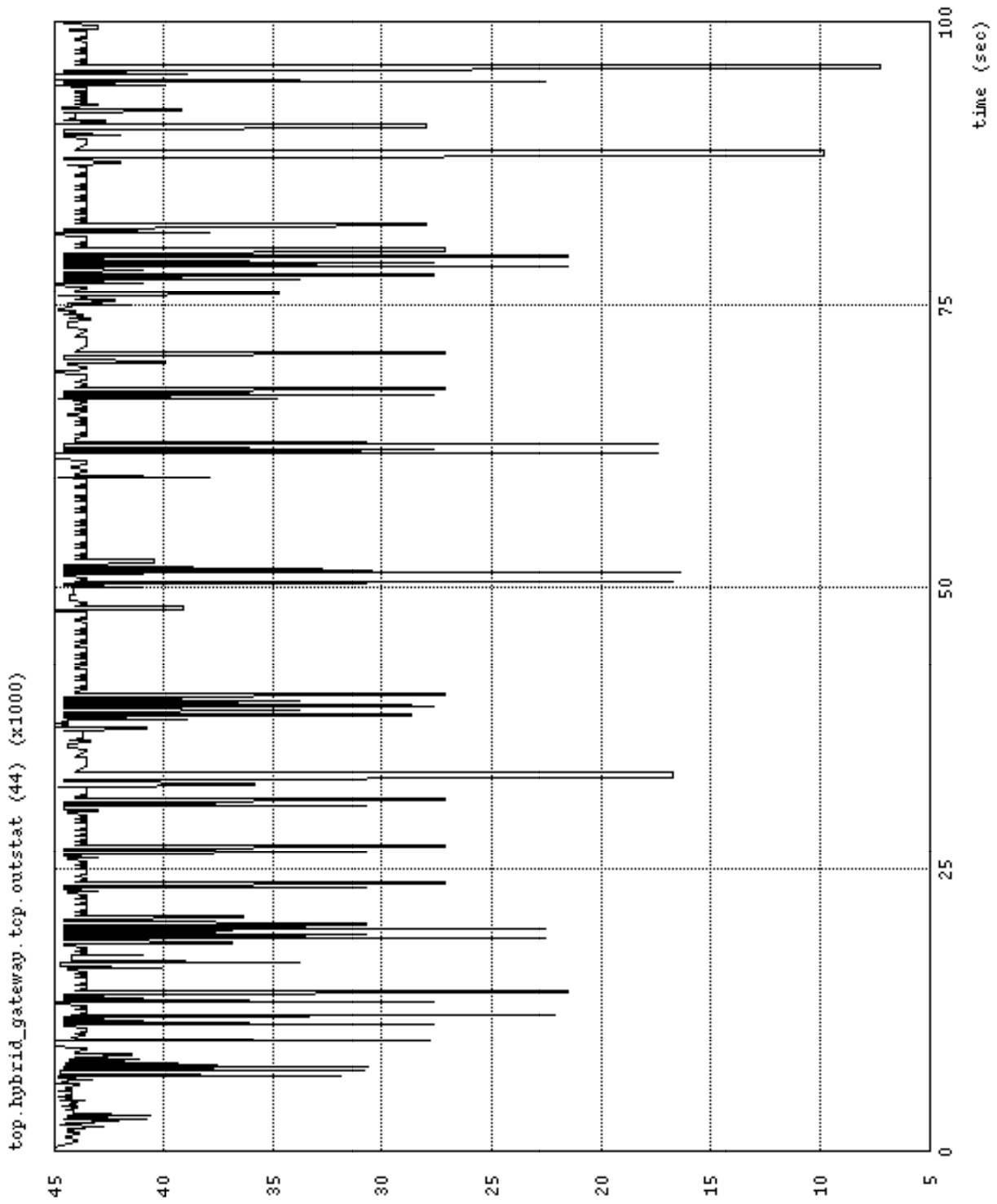


Figure 25: Window advertised by the hybrid host at the hybrid gateway

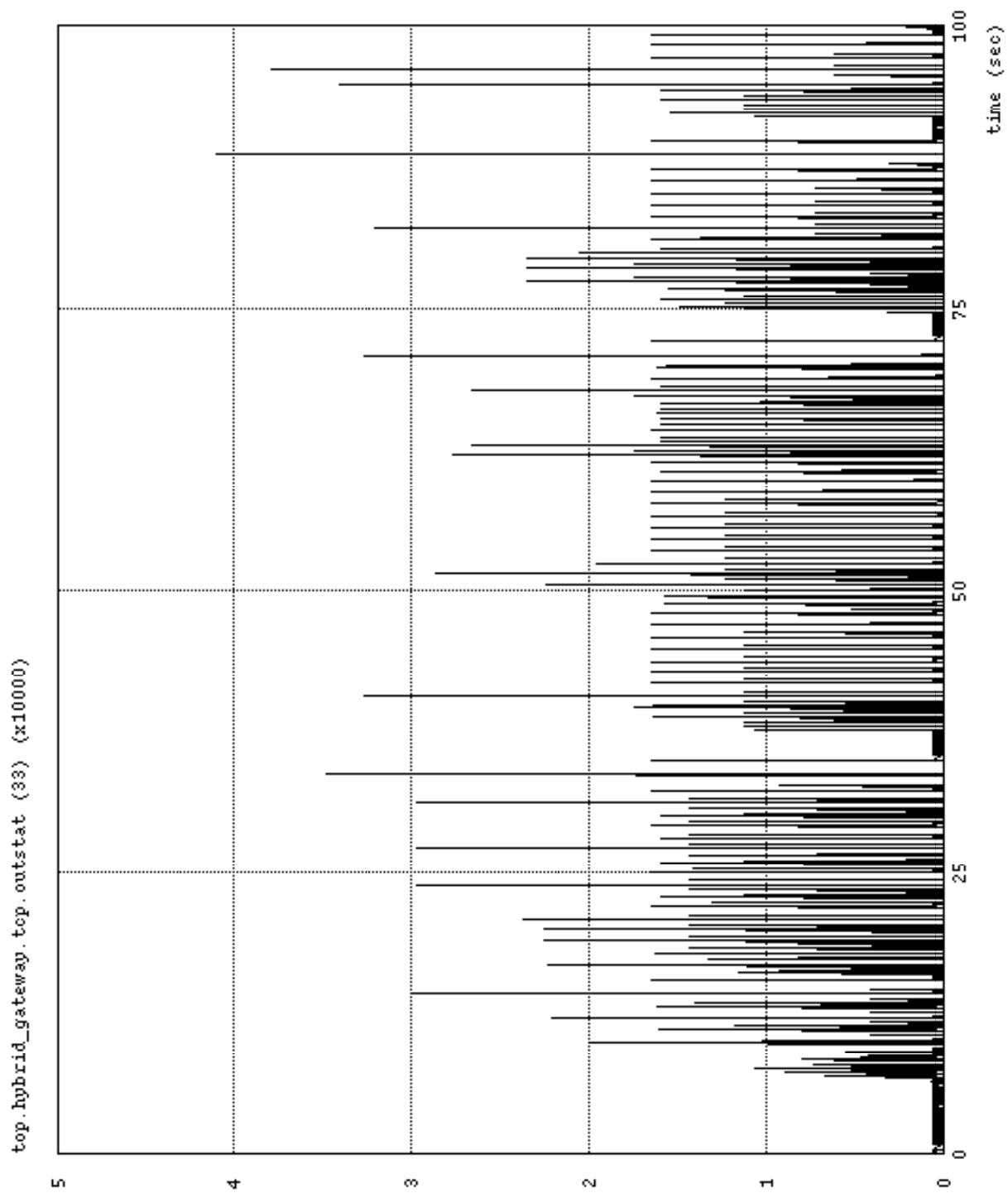


Figure 26: Amount of data sent by the hybrid gateway to the hybrid host

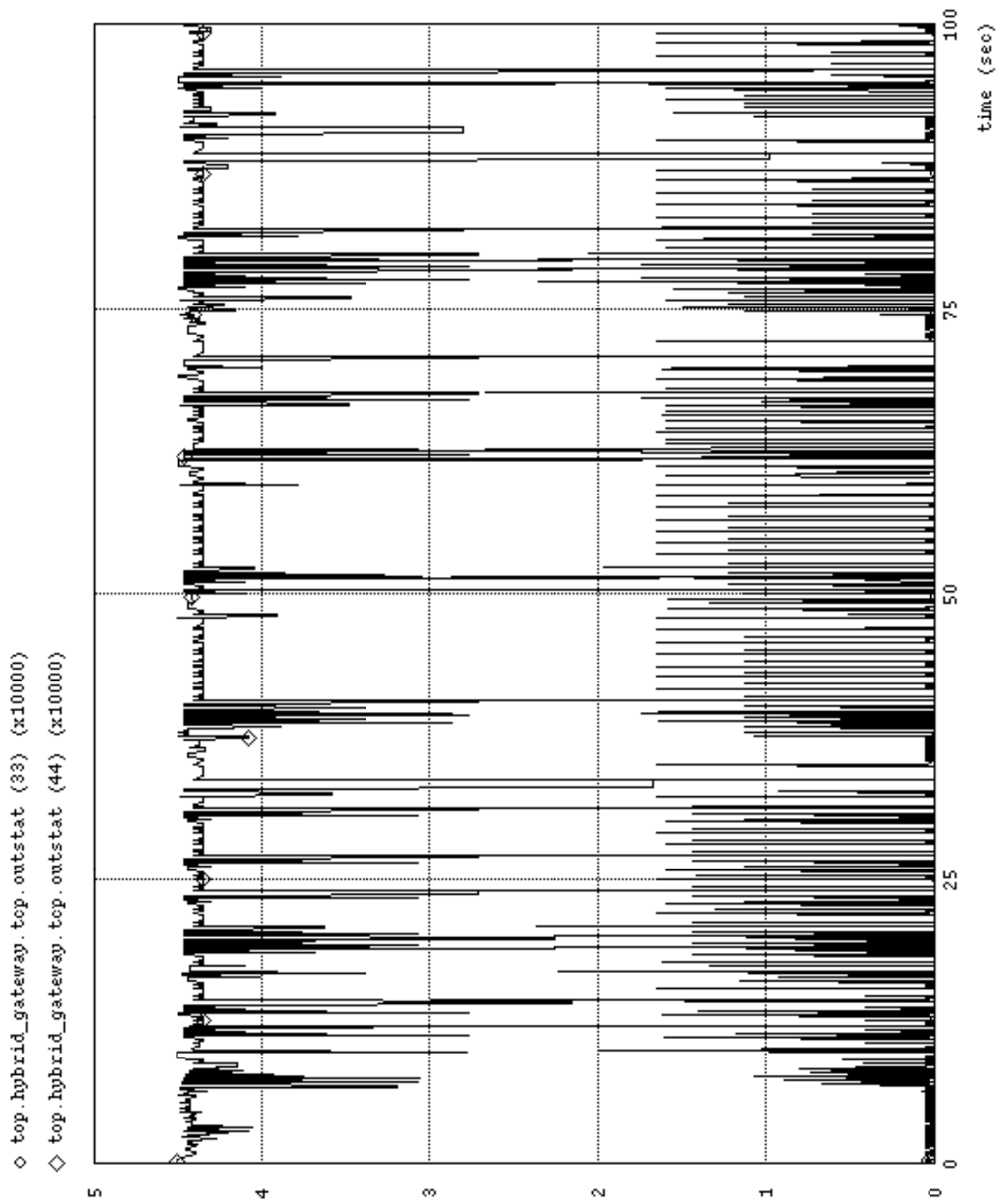


Figure 27: Comparison between the window size and the amount of data eligible to be sent by the hybrid gateway

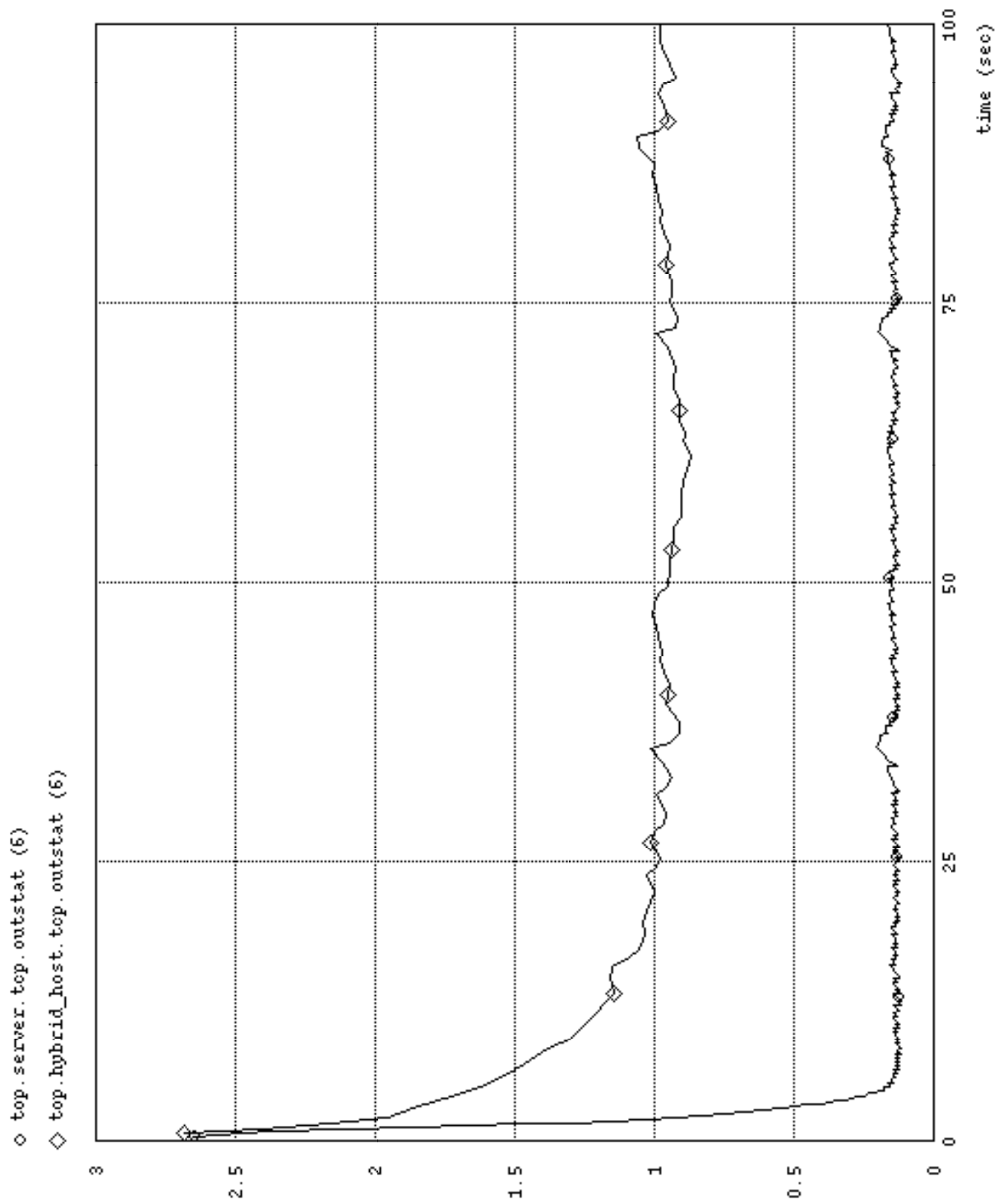


Figure 28: RTT measured at the server and the hybrid host

6.3.3 Second set of results with hybrid parameters and a large gateway window

parameter	value assigned
hybrid_host RCV_BUFFER	45 kbytes
hybrid gateway RCV_BUFFER	1000 kbytes
server RCV_BUFFER	4096 bytes
hybrid gateway retransmission timeout	200 ms

Table 5: Values of the hybrid parameters with a large hybrid gateway window

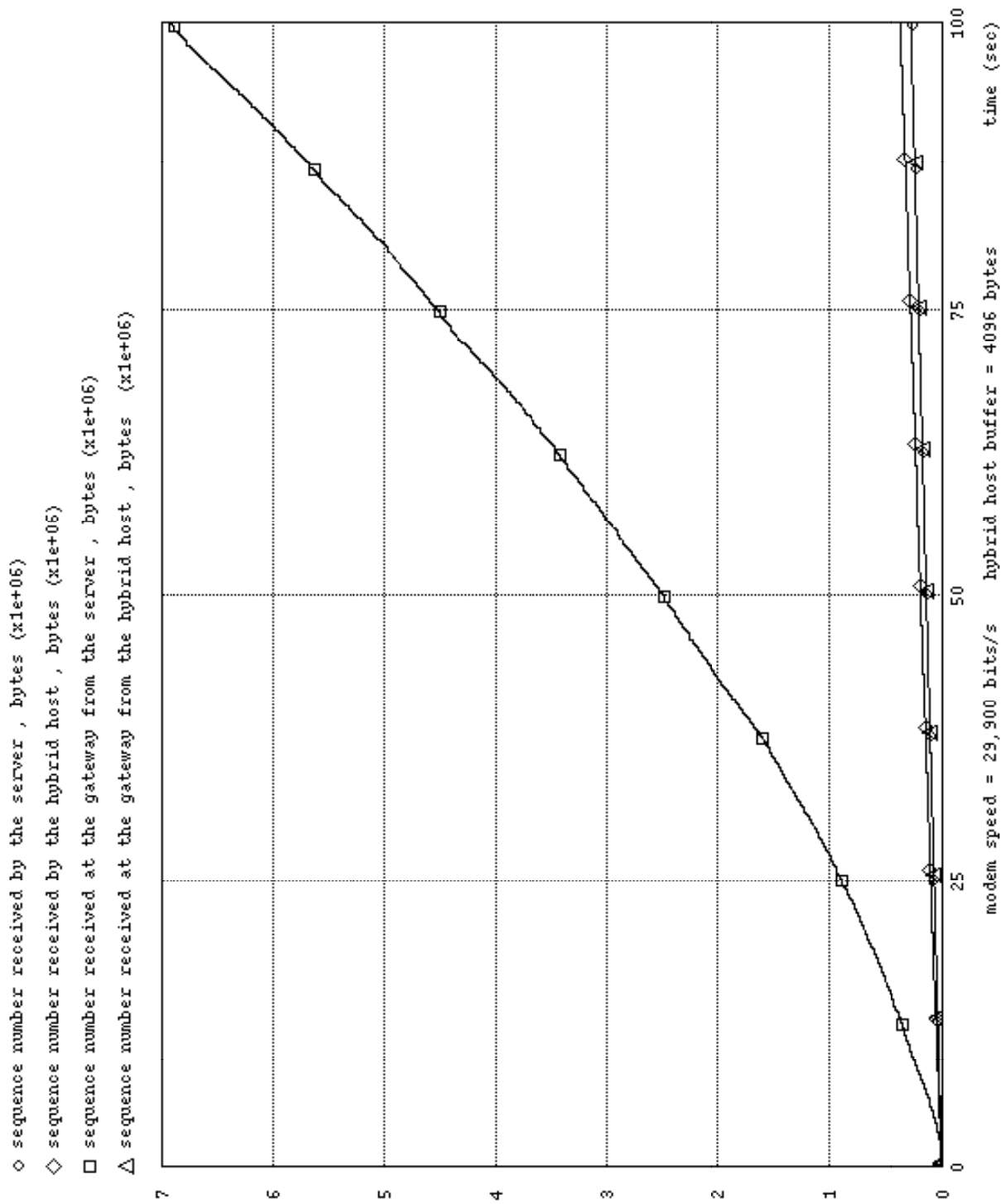


Figure 29: Throughput at the end-user and at the hybrid gateway

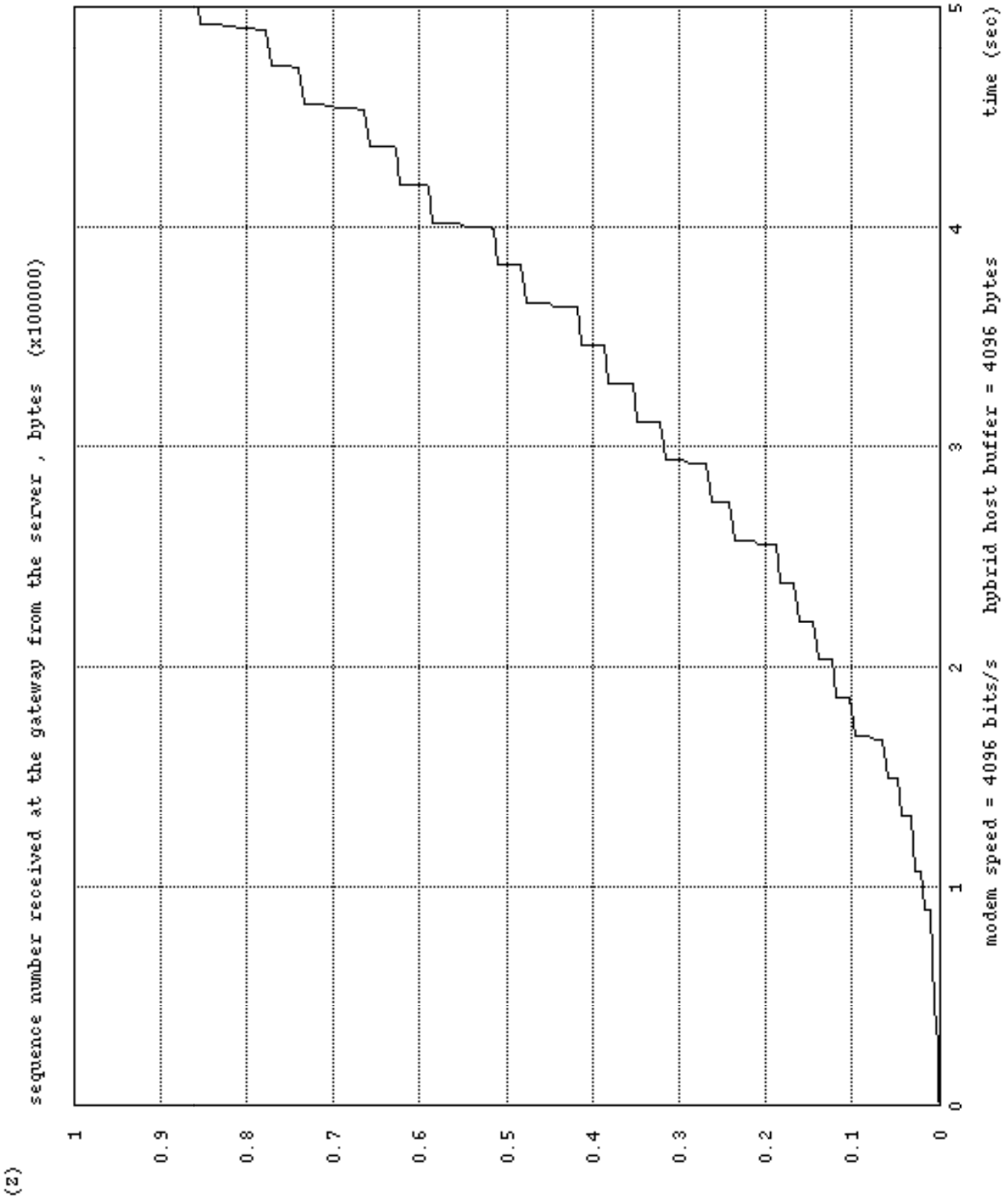


Figure 30: Downstream data load

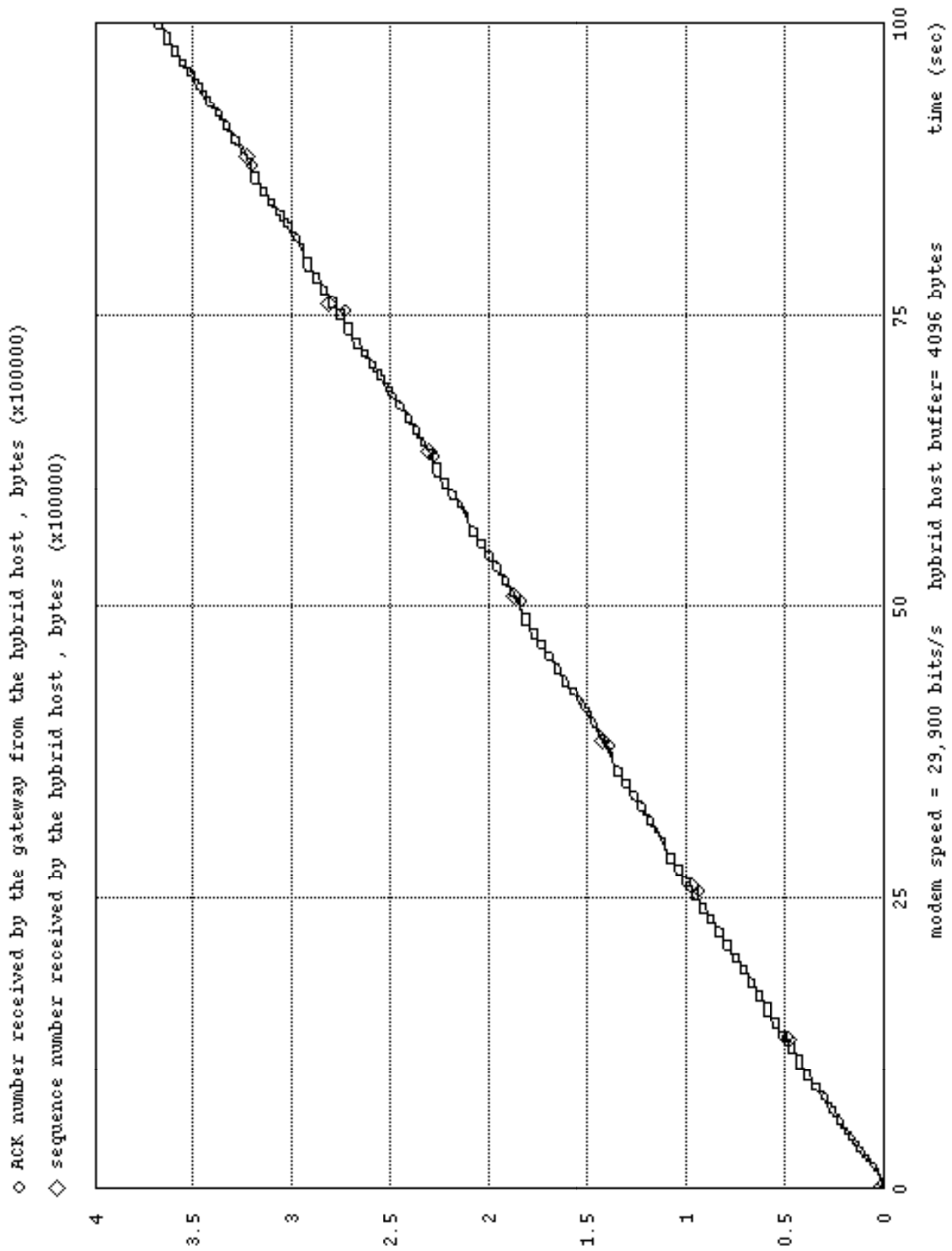


Figure 31: Sequence number received by the hybrid host and ACK number received at the hybrid gateway

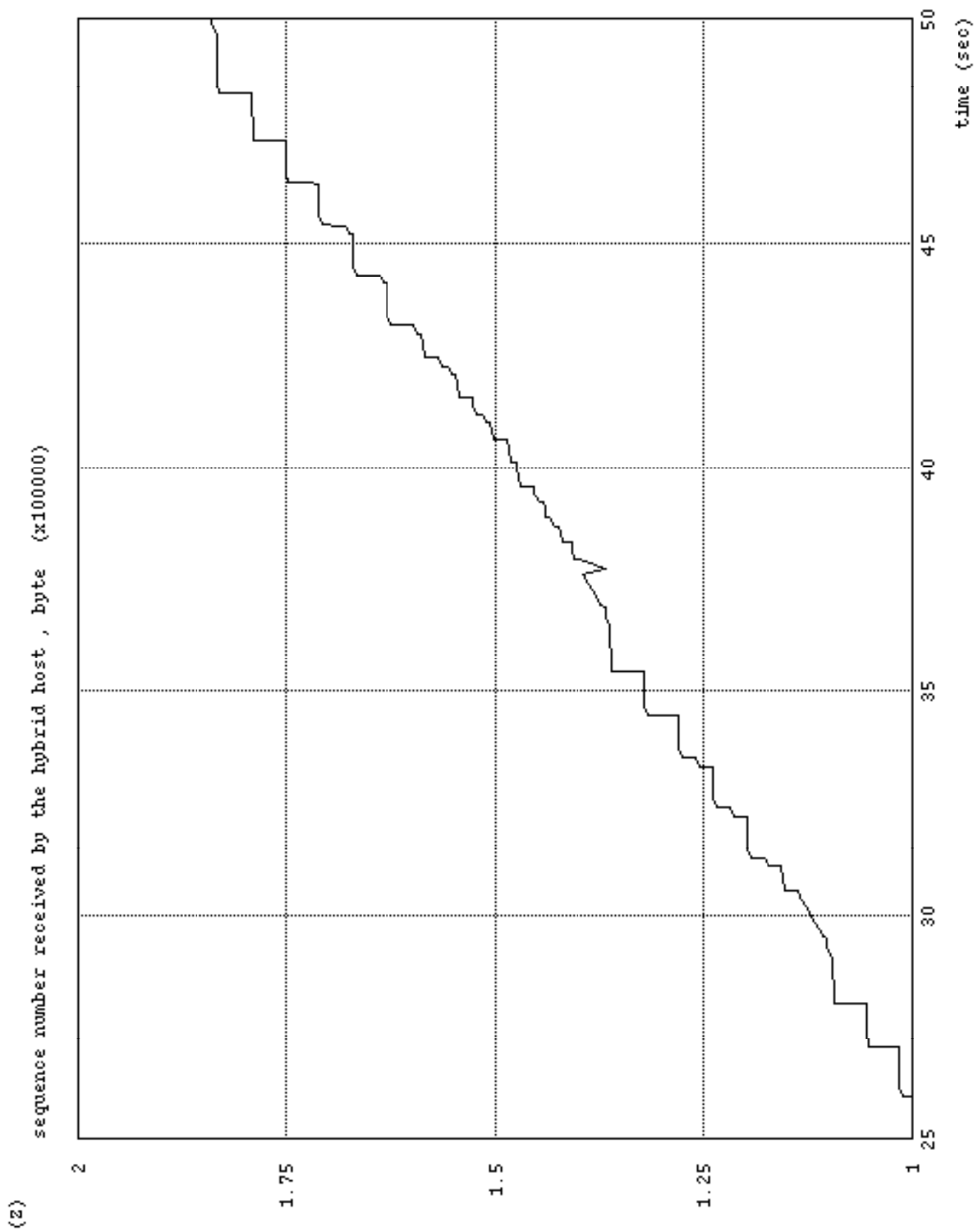


Figure 32: Example of a duplicated segment received by the hybrid host

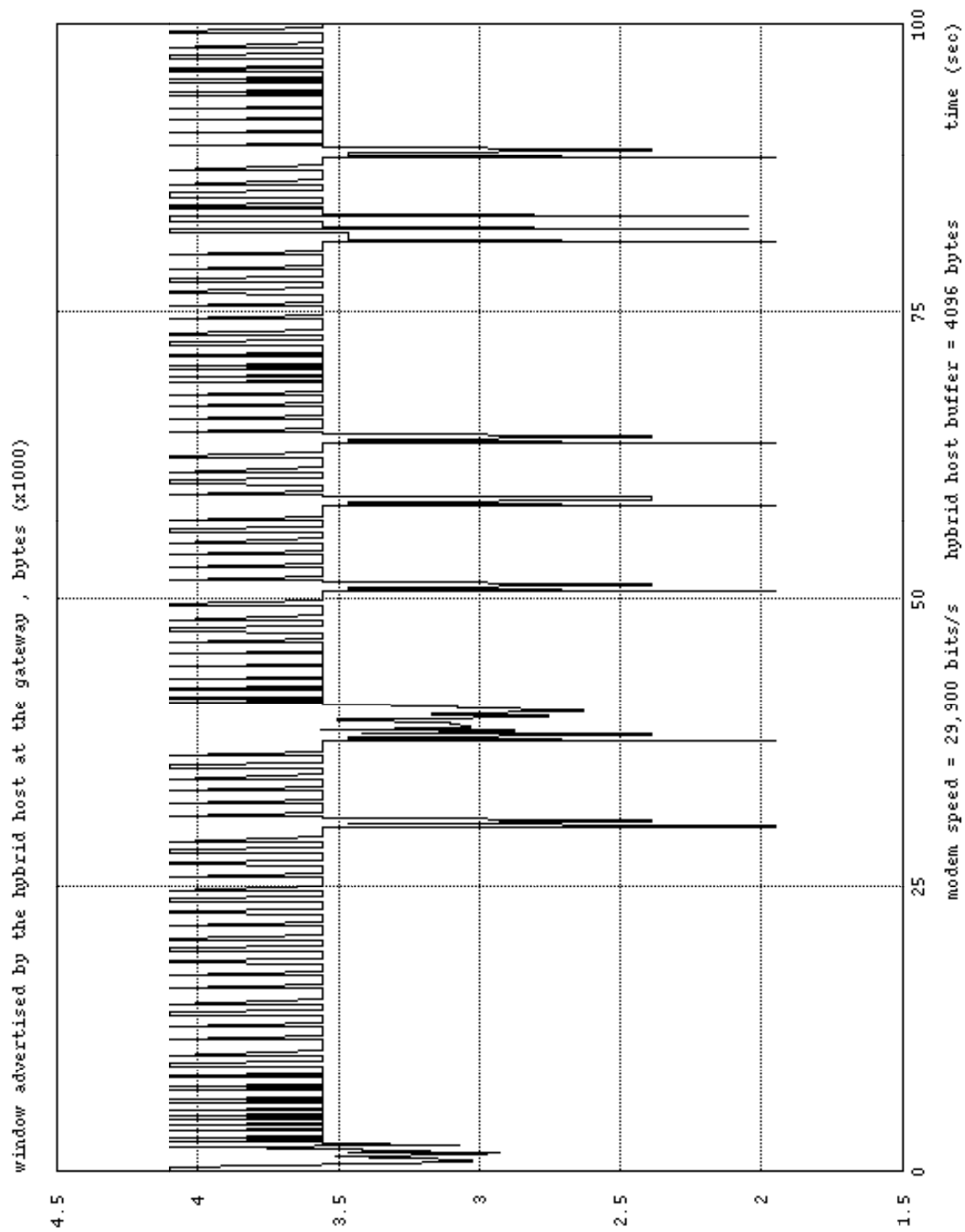


Figure 33: Window advertised by the hybrid host at the hybrid gateway

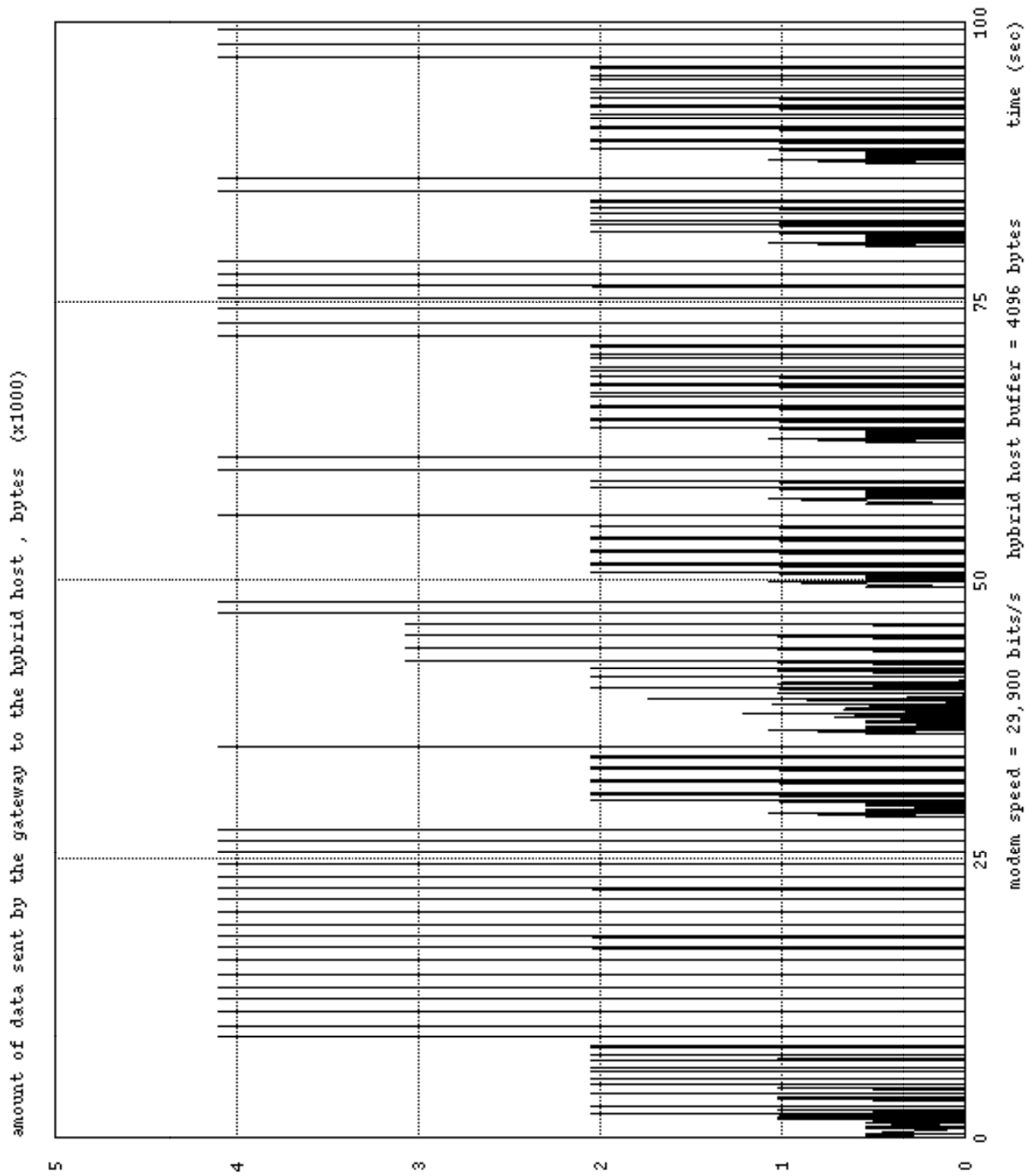


Figure 34: Amount of data sent by the hybrid gateway to the hybrid host

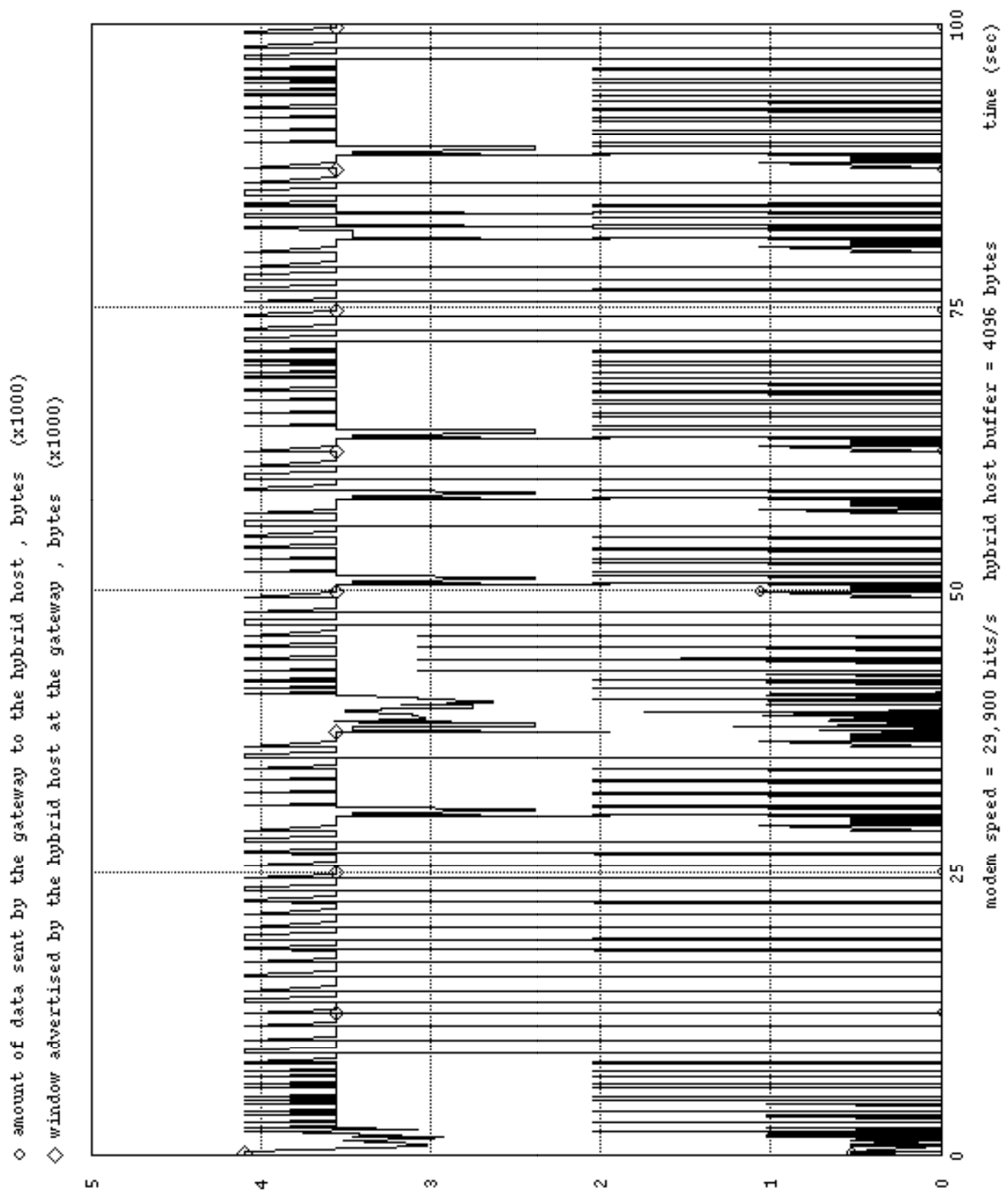


Figure 35: Comparison between the window size and the amount of data eligible to be sent by the hybrid gateway

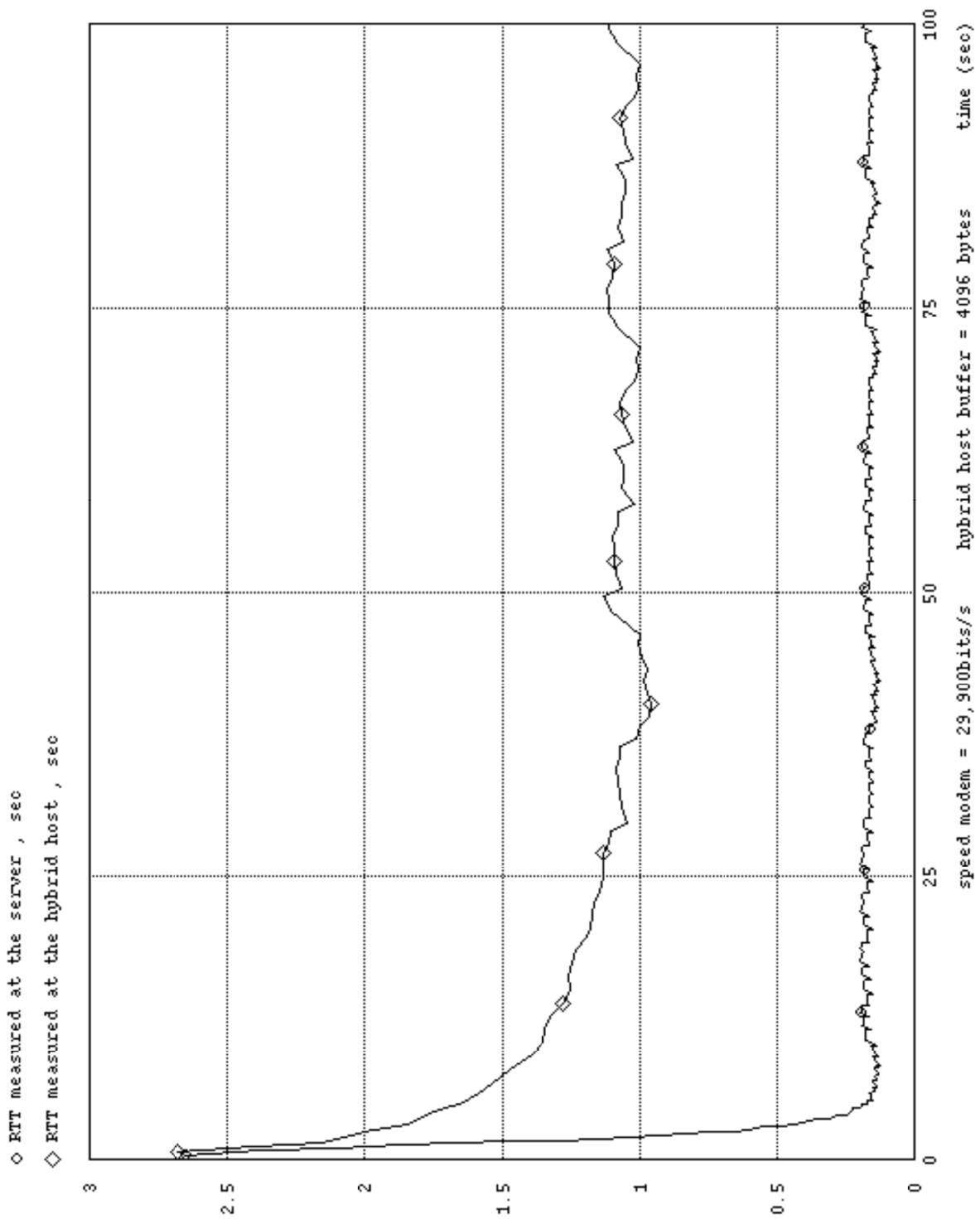


Figure 36: RTT measured at the server and the hybrid host

6.3.4 Third set of values with large parameters

parameter	value assigned
hybrid_host RCV_BUFFER	1000 kbytes
hybrid gateway RCV_BUFFER	1000 kbytes
server RCV_BUFFER	4096 bytes
hybrid gateway retransmission timeout	200 ms

Table 6: Values of the large parameters

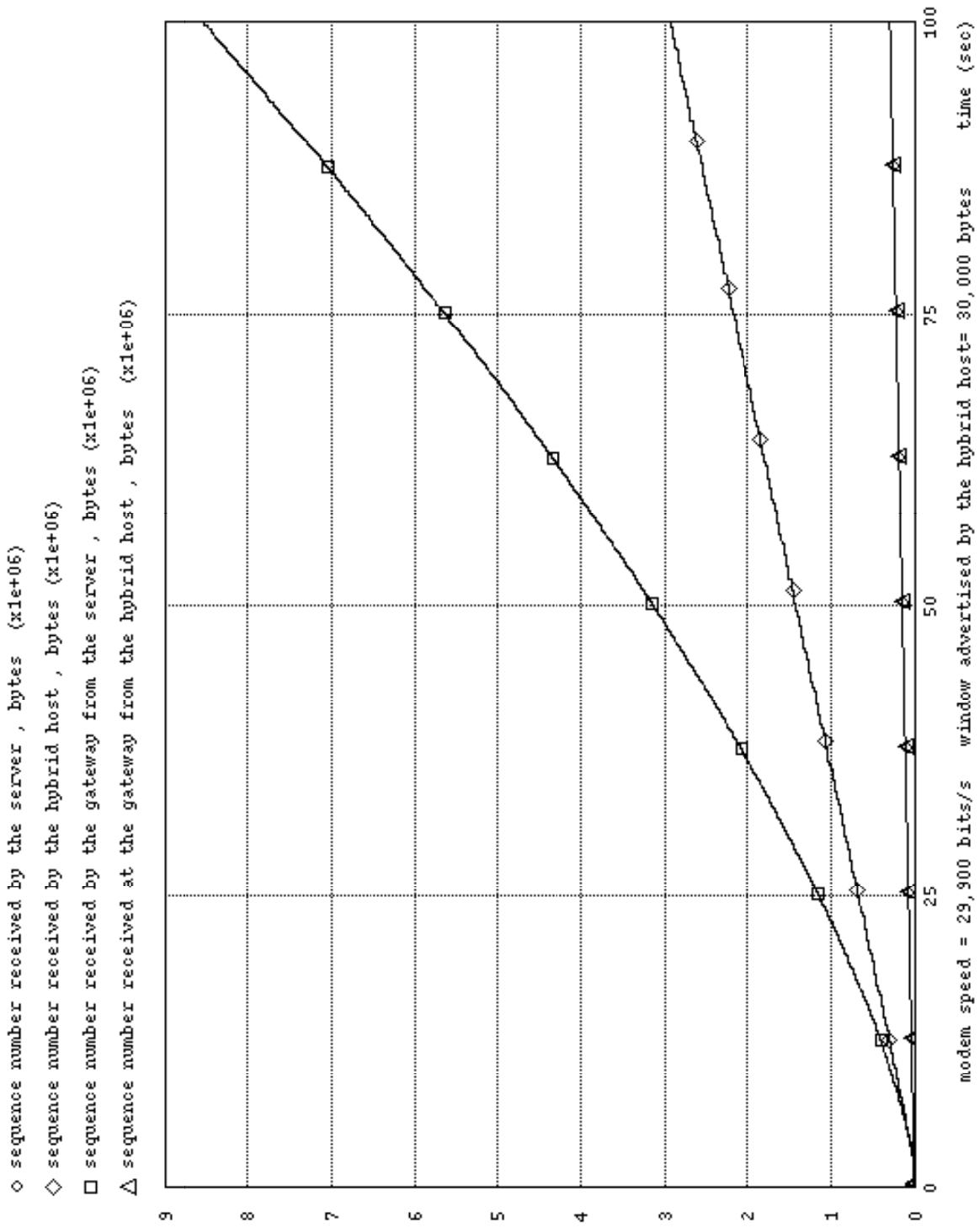


Figure 37: Throughput at the end-user and at the hybrid gateway

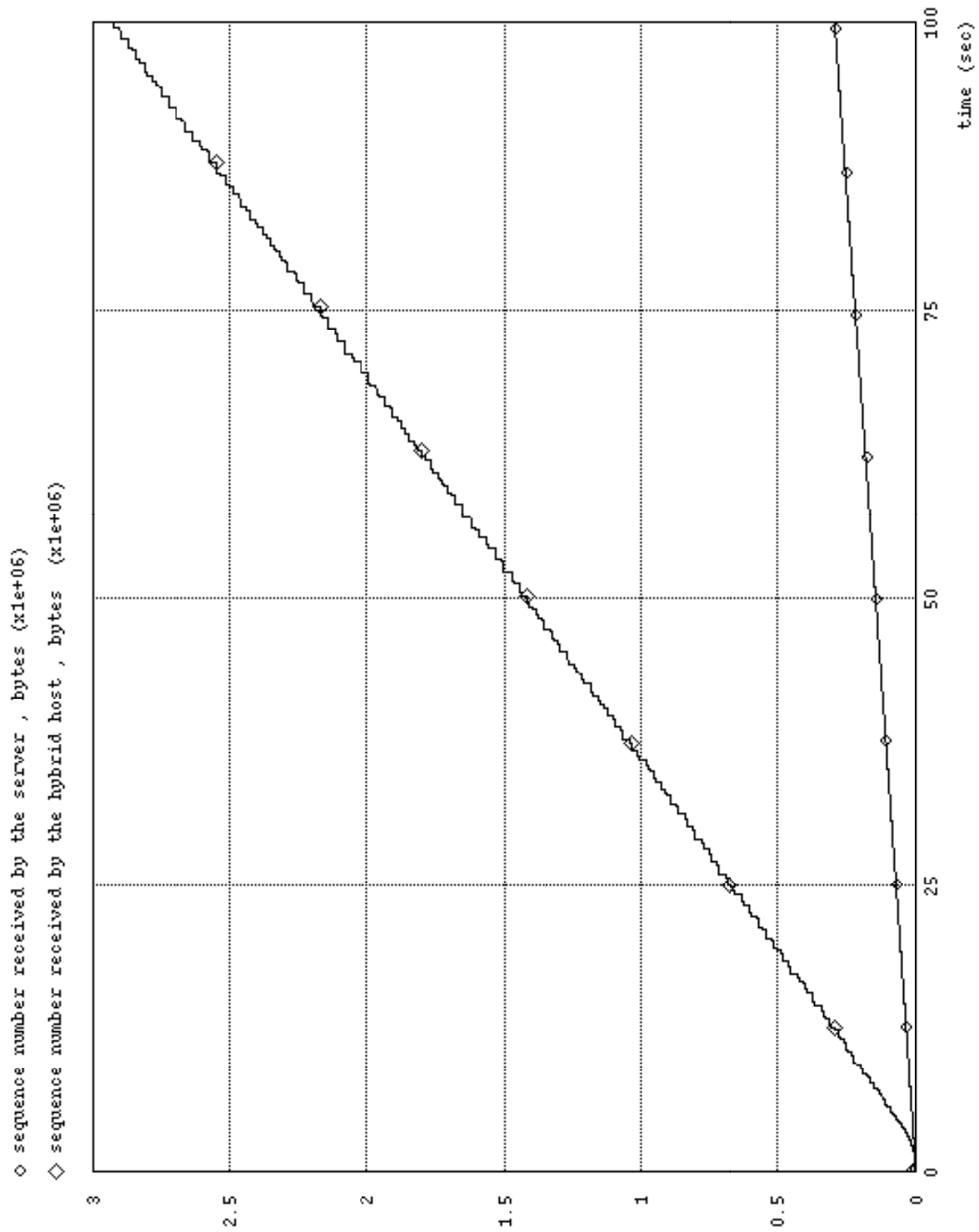


Figure 38: Comparison of the throughput at the end-user

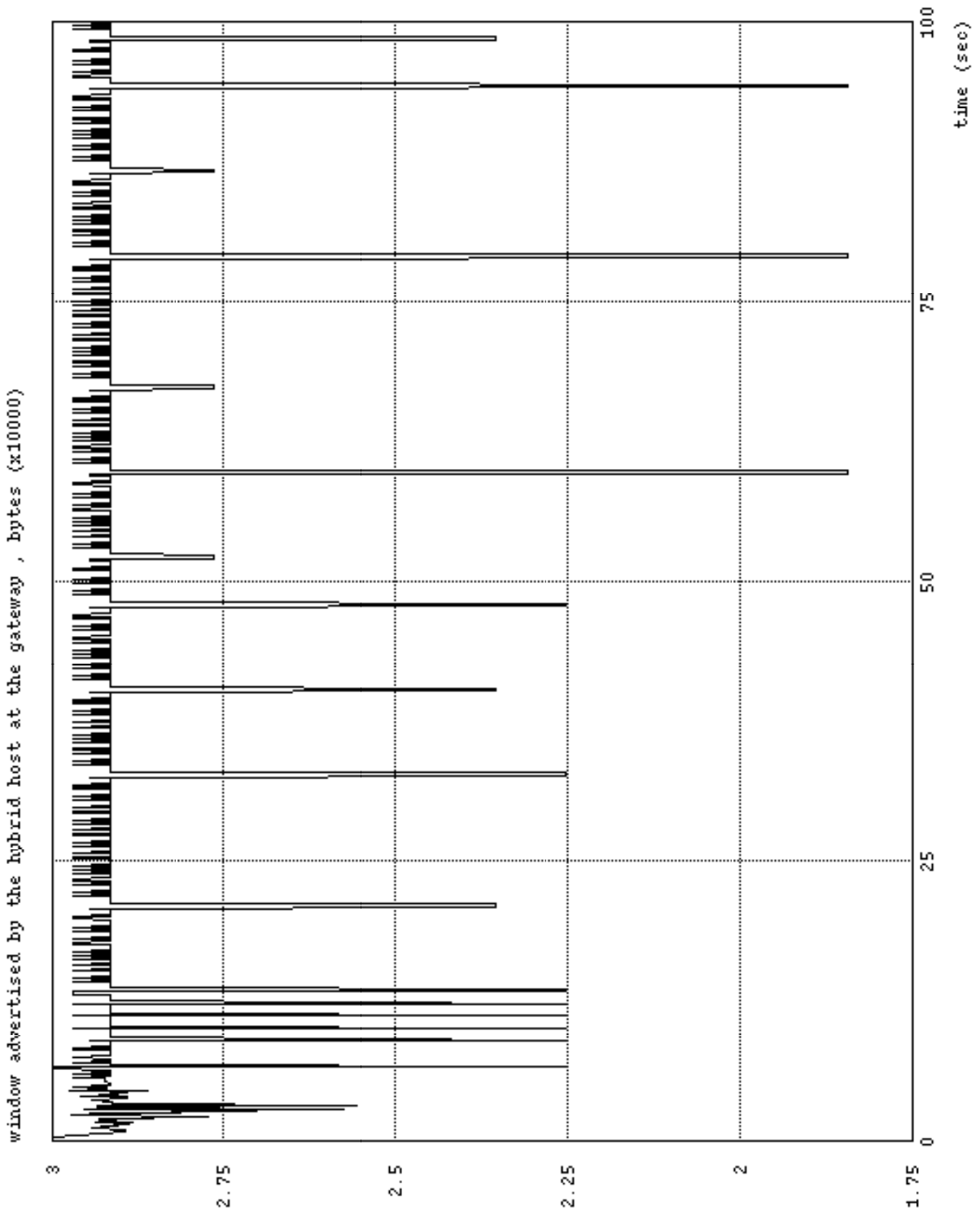


Figure 39: Window advertised by the hybrid host at the hybrid gateway

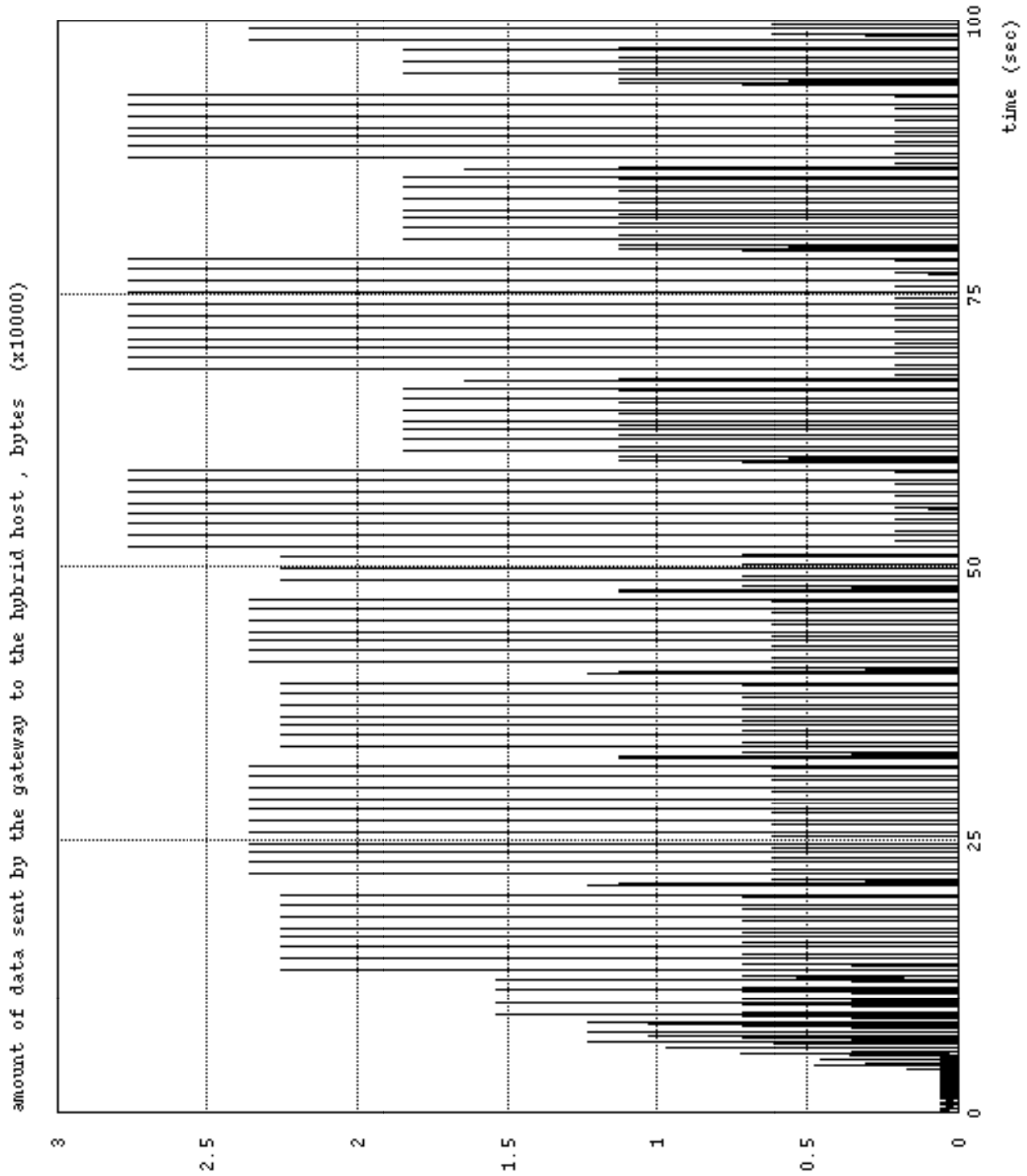


Figure 40: Amount of data sent by the hybrid gateway to the hybrid host

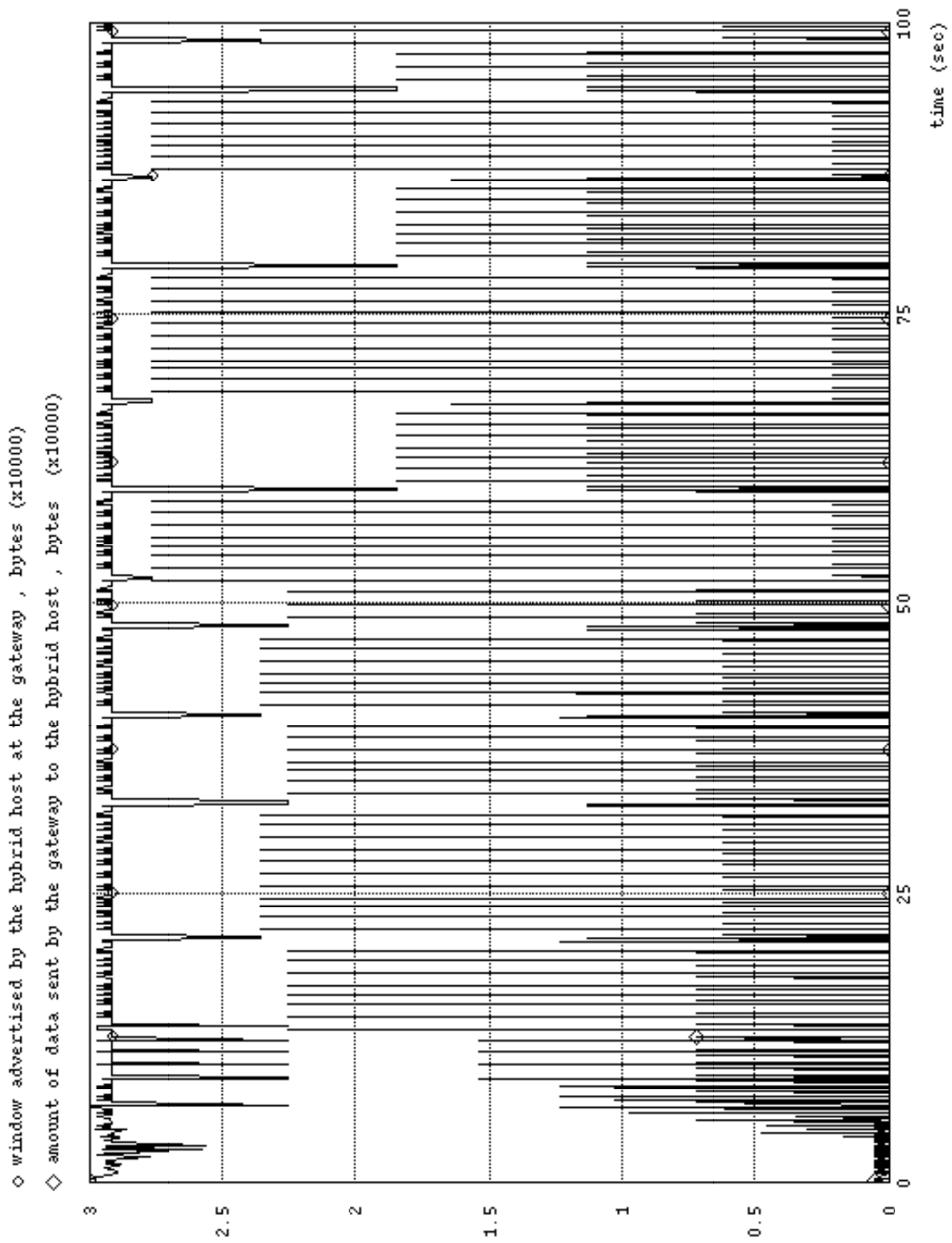


Figure 41: Comparison between the window size and the amount of data eligible to be sent by the hybrid gateway

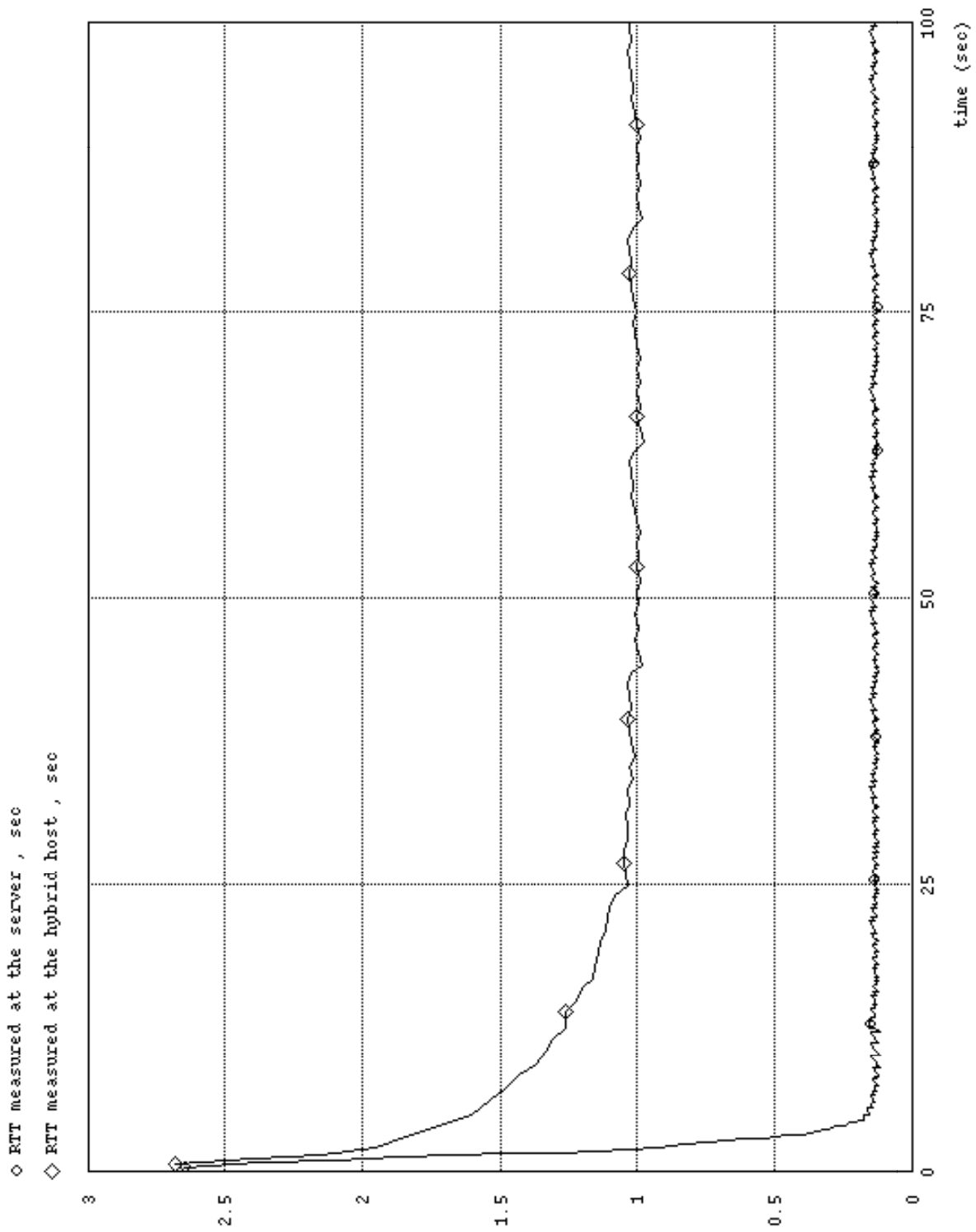


Figure 42: RTT measured at the server and the hybrid host

6.3.5 Fourth set of values with a smaller ACK timeout at the hybrid gateway

The hybrid gateway doesn't acknowledge segments right away. Instead it waits for some data to go in the same direction as the ACK, to have the ACK piggy-backed until a time out expires. This set of curves have been obtained with an ACK time out of 10 ms instead of 200ms in the previous results.

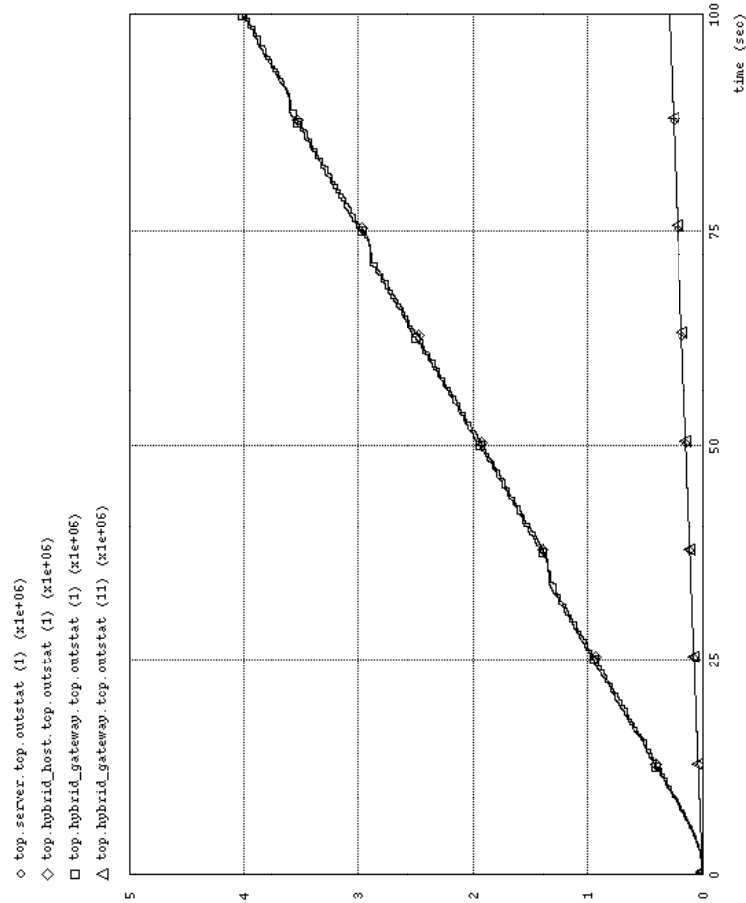


Figure 43: Throughput at the end-user and at the hybrid gateway

When the value of the default ACK time out is smaller, we see in figure 43 that the overall throughput is better.

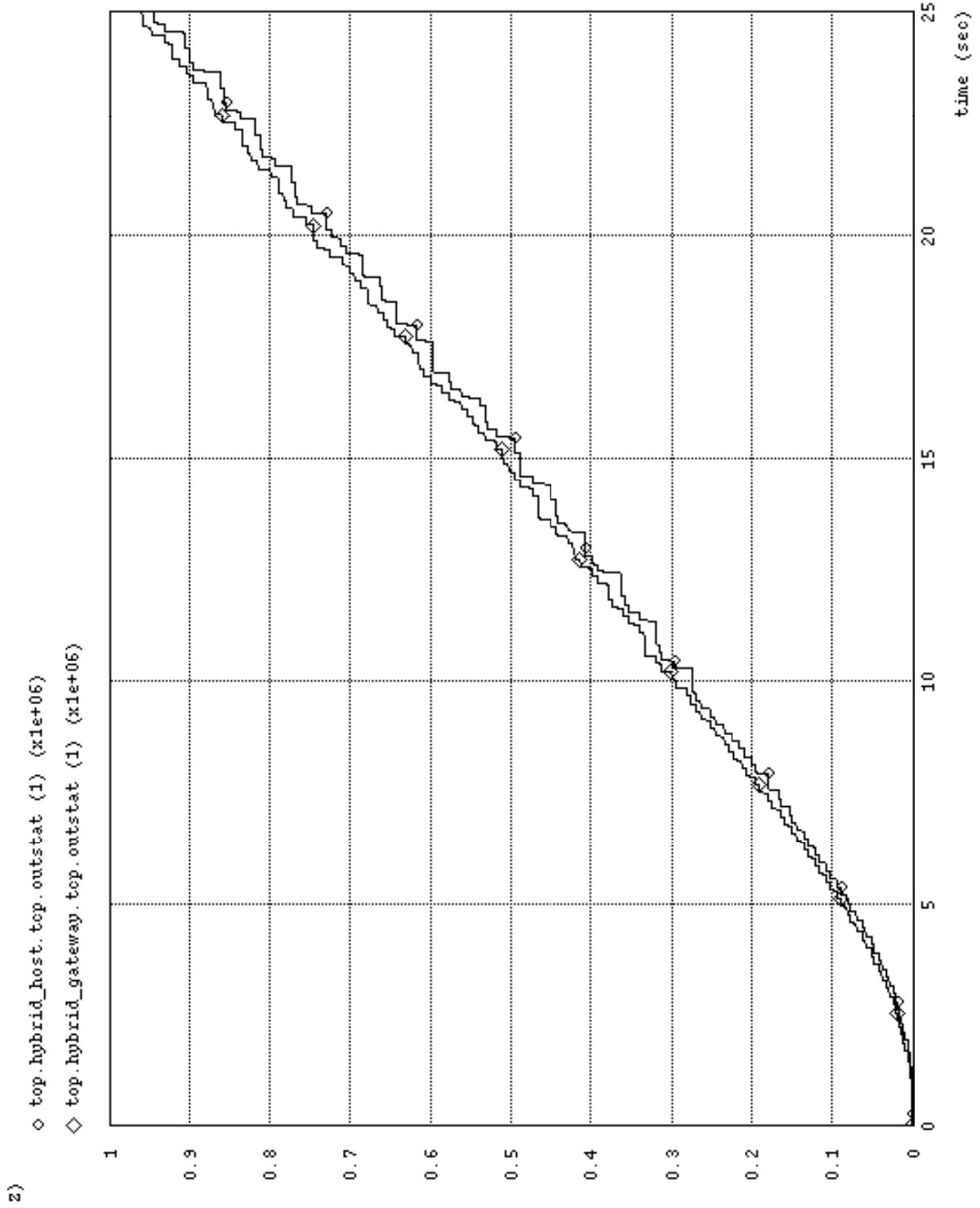


Figure 44: Comparison of throughput at the hybrid host and at the hybrid gateway

7 Future Enhancements of the simulation model

One of the main interests of this work, besides the results, is to confirm the possibility to handle within a model a very accurate description of the mechanisms at work in the Internet. Several enhancements are being added to the model in order to make it more realistic and to: (i) match the results obtained in the real system; (ii) develop future improvements of performance. These additional features include:

- merge hybrid network traffic with standard Internet traffic
- define a more realistic rate of packet generation by the application layer
- increase the size of the network, i.e. the number of users
- add different kinds of traffic: periodic broadcast, periodic multicast, interactive, etc.
- implement the function of the Network Operations Center (NOC) in order to manage different priorities according to the type of traffic
- find out the optimal policy to maximize the throughput along with the best pricing policy

In order to increase the performance of the system, several enhancements can be implemented within the hybrid gateway. Fast retransmit and Fast Recovery algorithms should be implemented within the hybrid gateway [4]. Additional techniques such as RED, SACK, and queue management are also being added.

References

- [1] V. Arora, N. Suphasindhu, J.S. Baras, and Douglas Dillon. “Asymmetric Internet Access Over a Satellite-Terrestrial Networks”. *AIAA Proceedings of the 16th International Communications Satellite Systems Conference and Exhibit*, pages 476–482, 1996.
- [2] V. Arora, J.S. Baras, D. Dillon, A.D. Falk, and N. Suphasindhu. “Hybrid Internet Access”. *Proceedings of the AIP Conference on NASA Centers for Commercial Development of Space*, pages 69–74, 1995.
- [3] V. Arora, N. Suphasindhu, D. Dillon, and J.S. Baras. “Effective Extensions of Internet in Hybrid Satellite-Terrestrial Networks”. *Proceedings of the 1st Conference of Commercial Development of Space*, pages 345–350, 1996.
- [4] W.R. Stevens. “*TCP/IP Illustrated Volume 1*”. Addison-Wesley, 1996.