Performance limits of the tmm protocol : modelling and measurement

Erol Gelenbe, Patrice Mussard, Jean-Louis Grange

IRIA
laboria

laboratoire de recherche
en informatique
et automatique

PERFORMANCE LIMITS OF THE TMM PROTOCOL : MODELLING AND MEASUREMENT

Erol GELENBE
Jean Louis GRANGE
Patrice MUSSARD

Institut de Recherche
d'Informatique
et d'Automatique

Domaine de Voluceau
Rocquencourt
B. P. 105 78150 - Le Chesnay
France
Tél.: 954 90 20

Rapport de Recherche N° 230

Avril 1977

PERFORMANCE LIMITS OF THE TMM PROTOCOL : MODELLING AND MEASUREMENT

Erol Gelenbe*, Jean-Louis Grangé, Patrice Mussard****

Résumé :

TMM est une des procédures de transmission utilisées dans CIGALE et dont la principale caractéristique de fonctionnement est le contrôle de la transmission en cours par blocage des émissions suivantes jusqu'à réception d'un acquittement correct ou échéance d'un time-out. Ce rapport présente une évaluation de ce type de procédure permettant de rechercher les limites des performances de TMM compte tenu de la capacité de transmission de la ligne disponible et des caractéristiques du trafic. Des mesures effectuées sur CIGALE permettent de valider les prédictions d'un modèle mathématique de TMM que nous proposons. Ce modèle nous permet également de calculer la longueur moyenne des paquets qui maximise le débit de la procédure TMM.

Abstract :

The TMM protocol uses a stop-and-wait procedure with acknowledgements for error control. The purpose of this paper is to investigate the basic limits of TMM with respect to its utilisation of available line capacity. Measurement results are obtained and a mathematical model is presented in order to predict TMM's performance and to derive optimum average lengths for random length or fixed length packets.

* IRIA-LABORIA, B.P. 105 - 78150 Le Chesnay.
** IRIA-Projet CYCLADES, B.P. 105 - 78150 Le Chesnay.

1. Introduction

CIGALE [1] is the packet switching network of CYCLADES [2]. The basic service offered by CIGALE simply consists of transporting independent packets to their destination : this type of service has been called Datagram service within CCITT* .

In order to reach their destination, packets are forwarded from node to node through point to point synchronous lines. Since lines introduce delays and errors, line handling protocols are needed to provide for efficient and reliable transmissions. Standard manufacturer's protocols are made available in CIGALE so as to avoid any special piece of hardware or software for host connections. The purpose of this paper is to present some results on performance measurement and evaluation of the TMM protocol.

TMM is a CII standard protocol [3] . It allows full duplex symmetrical transmission, i.e. physical transmission in both directions at the same time. It uses a stop-and-wait scheme to control the packet transmission : a new packet can be transmitted over the line only when the previous one has been acknowledged. A timout is set before transmitting each packet. If it expires before receiving the acknowledgement, the packet is retransmitted. A simple packet numbering scheme allows to detect and discard duplicated packets. It is to be noted that, since the transmission of a new packet cannot occur before receiving the acknowledgment of the previous one, any delay in acknowledgment transmission leads to the waste of a portion of the bandwith available for the reverse traffic. In particular, short packet traffic may be considerably slowed down by long packets transmitted in the reverse direction : short packet acknowledgments are delayed due to the transmission time of long packets.

---

* Comité Consultatif International Télégraphique et Téléphonique

Figure 1 shows how a part of the bandwidth is wasted if packets are not of the same length in both directions of the line.

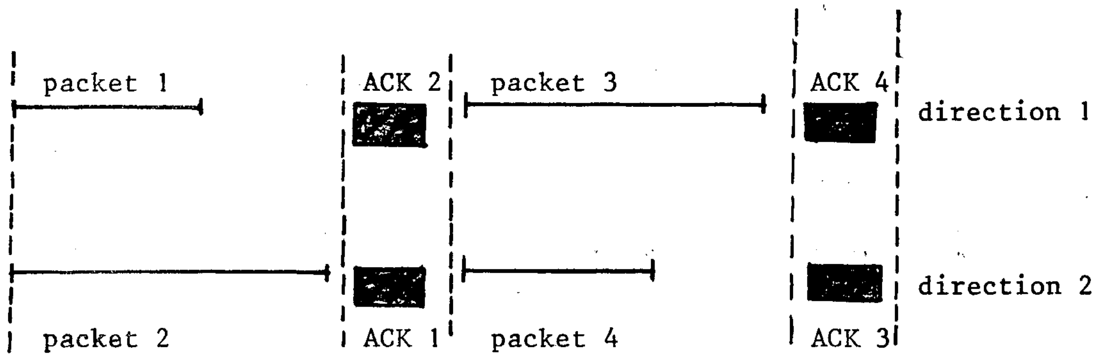
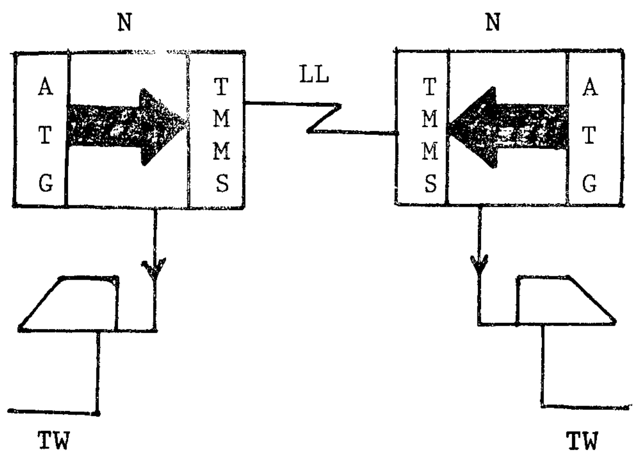


FIGURE 1

Transmission diagram in both directions for the TMM protocol

2. The measurements

In order to evaluate the performance of the TMM protocol various measurements have been performed. For that purpose, the configuration shown in Figure 2 has been set up.



N = CIGALE node
LL = local synchronous line
TMMS = TMM protocol software
ATG = Artificial traffic generator
TW = Control Typewritter.

FIGURE 2

Two nodes are connected by a local telephone line at 19 200 bits/sec. Base band modulation modems are used and the line length is extremely short (about 2 meters) : propagation delays through the hardware are negligible compared to the transmission delays.

Each node is loaded with the regular CIGALE node system including the TMM protocol software (TMMS) and an artificial traffic generator (ATG).

In all experiments, ATG is used to generate fixed length packets at the maximum rate in both directions, i.e. as fast as TMMS can accept them.

A typewritter is connected to each node and is used to control the ATG (e.g, start, stop) and to extract measurement data from the node system.

A number of experiments have been conducted with various combinations of packet length at each end. Throughput results obtained are summarized in Figure 3 and 4.

In the following sections of this paper we present a mathematical model of TMM and validate it with the measurement data obtained. We then obtain some basic properties of this protocol. In particular we compute the maximum throughput of TMM in the presence of packets of variable lengths.

A study of the time-out optimization for stop-and-wait schemes is presented elsewhere [6]. In this study we assume that error rates can be neglected.

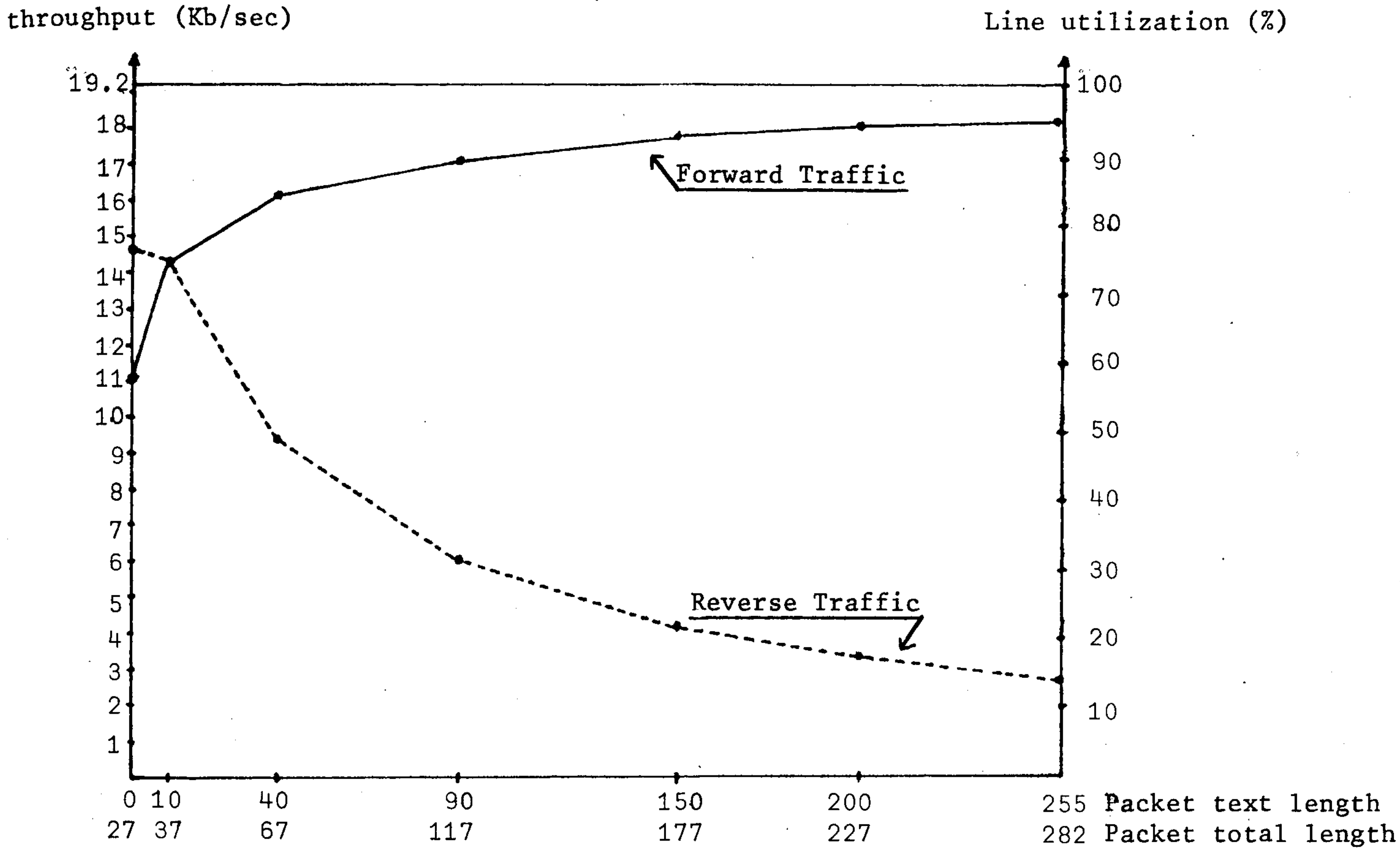


Figure 3 : TMM Performances (*) (Reverse Traffic Packet Length : 37 octets)

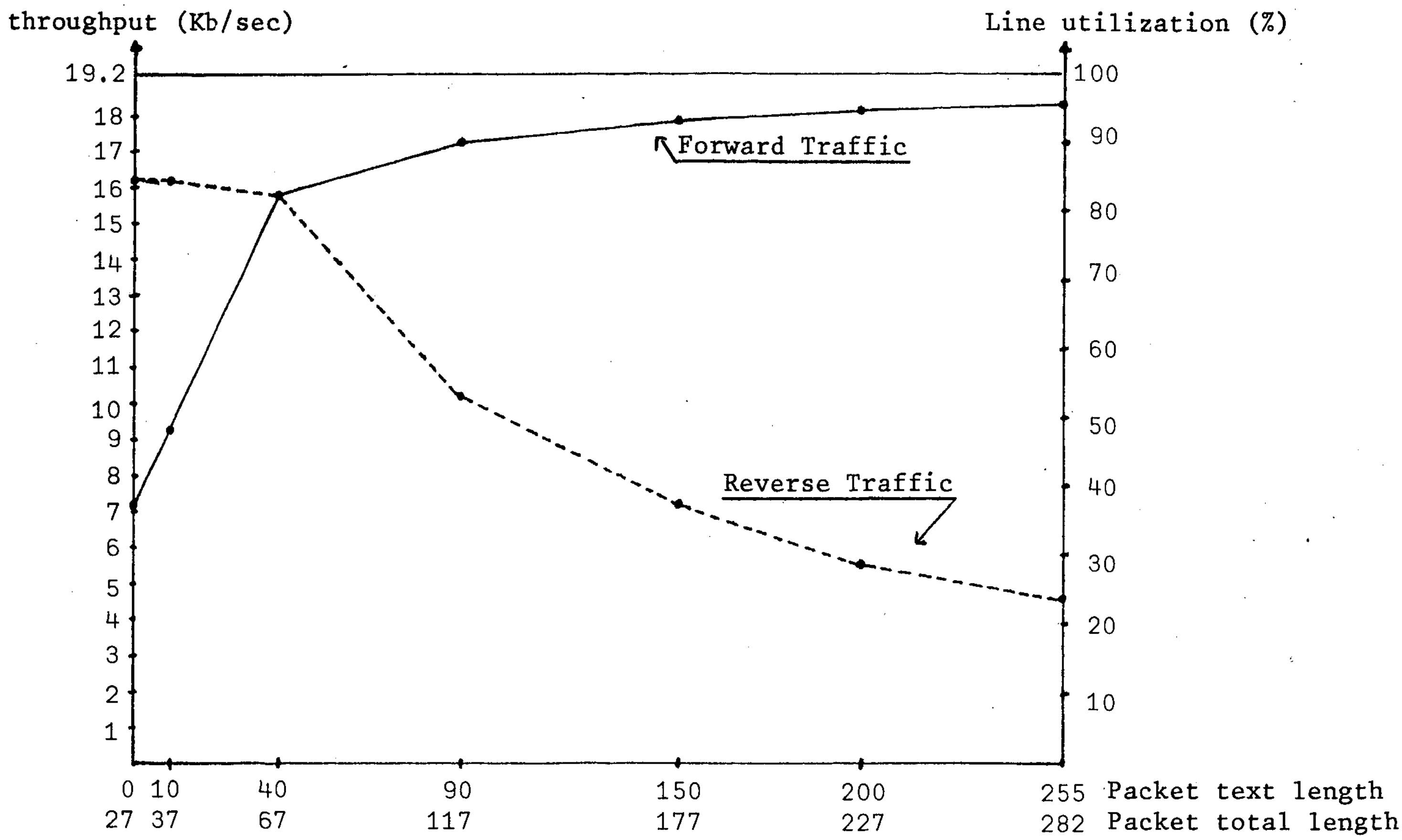


Figure 4 : TMM Performances (*) (Reverse Traffic Packet Length : 67 octets)

(*) The plotted throughput and line utilization include overhead (i.e. synchronization characteres, packet header, acknowledgements).

3. The model and its validation

The behaviour of the protocol is modelled by examining the state of each of the two nodes : let S_1 and S_2 be the state node 1 and 2 respectively. S_1 can take one of three values :

- state P corresponds to a packet transmission from 1 to 2
- state A corresponds to the transmission of an acknowledgement from 1 to 2
- state W is the state in which the sender of node 1 waits for an acknowledgement to arrive from node 2.

The states P, A, W of node 2 are defined in similar fashion.

We are assuming here that pure software times between state transitions of the senders are negligible. We shall see via our measurement results that this assumption is justified.

If both nodes are considered we obtain the state transition diagram of Figure 5.

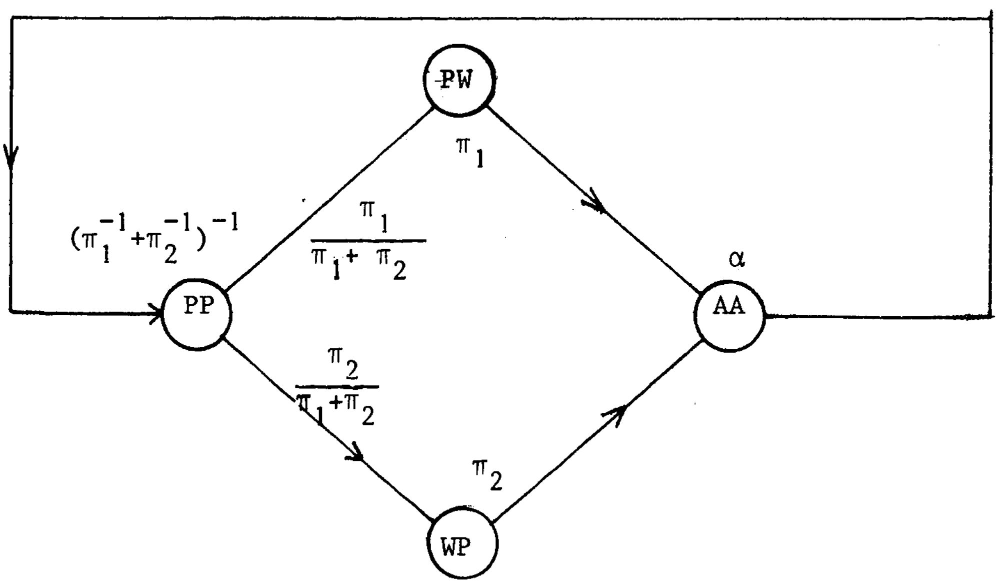


FIGURE 5

In state PP both senders are transmitting user packets, and a packet is travelling on each of the two lines. Depending on which packet transmission terminates first, state PW or state WP is entered. For instance, in state PW the packet transmission of sender 2 is complete and it enters the wait state (waiting for an acknoledgement to arrive from node 1). From state PW the system moves to state AA when the packet transmission from 1 to 2 is complete, since receiver 1 has received a packet and now proceeds to acknowledge it and so does receiver 2. Finally, after this last acknowledgement is transmitted the system will once more enter state PP.

Notice that the system cannot be in state PA or in state AP : if an acknowledgement is being transmitted from 1 to 2 then 2 <u>must</u> be waiting for it in state and vice-versa. Similarly, we do not allow state WW since we have excluded the possibility of software waiting times (such times have been assumed to be negligibly short compared to packet or acknowledgement transmission times; an estimate is obtained in Section 5).

The time spent in state AA is α (i.e. the transmission time of an acknowledgement). In the computations everything will be normalized so that α will also be the length in bytes of the acknowledgement packet. Let π_1 and π_2 be the average length of user packets sent from 1 to 2, and from 2 to 1, respectively. We shall assume in this section that user packets are exponentially distributed ; this restriction, which simplifies the analysis is not essential and is removed in the computations given in the Appendix. Our purpose in keeping this restriction in this section is to avoid complications which may confuse the reader.

With this assumption we show (in the Appendix) that the average time spent by the system in state PP at each passage through that state is $(\pi_1^{-1} + \pi_2^{-1})^{-1}$. The average time spent in state PW and WP is π_1 and π_2, respectively. Similarly we show that from state PP the system will next enter state PW with probability $\pi_1/(\pi_1 + \pi_2)$ or state WP with probability $\pi_2/(\pi_1 + \pi_2)$.

We notice that the model shown on Figure 5 is a semi-Markov process [4]. The steady-state probability that it will be in state S is denoted by $p(S)$: we shall need $p(PP)$, $p(PW)$, $p(WP)$ since these states are the only ones which contribute to the throughput. We define D_1 the throughput of sender 1 as

$$D_1 = p(PP) + p(PW)$$

and D_2 the throughput of sender 2

$$D_2 = p(PP) + p(WP)$$

Notice that in state PP both senders are busy. The total throughput is

(1) $$D = D_1 + D_2$$

These quantities are easily obtained as [3]

(2) $$D_1 = \frac{1 + \pi_1/\pi_2}{1 + \frac{\pi_1}{\pi_2} + \frac{\pi_2}{\pi_1} + \frac{\alpha}{\pi_1}\left(1 + \frac{\pi_1}{\pi_2}\right)}$$

(3) $$D_2 = \frac{1 + \pi_2/\pi_1}{1 + \frac{\pi_1}{\pi_2} + \frac{\pi_2}{\pi_1} + \frac{\alpha}{\pi_1}\left(1 + \frac{\pi_1}{\pi_2}\right)}$$

To simplify the notation we shall use $x = \pi_1/\pi_2$, $y = \alpha/\pi_1$. We notice the following after some simple algebra :

- For fixed π_2, D_1 is a monotone increasing function of π_1 which becomes 1 as $\pi_1 \to \infty$. Similarly, for fixed π_2 it is monotone decreasing in π_2 tending to 0 as $\pi_2 \to \infty$.

- The above statements can be made also for D_2 if we interchange π_1 and π_2 (see Figure 6).

These theoretical results are illustrated on Figure 6. We have taken three values of π_1 (10, 20 and 100 bytes) and have set $\alpha = 7$ bytes (this is the length of an acknowledgement packet in CIGALE [1]. We show D_1 and D_2, the forward and reverse traffic rate as a function of π_2 (the average length of packets in the reverse traffic. We also show D the total throughput (sum of forward and reverse traffic rate). Notice that equations (1), (2), (3) give the normalized traffic rate ; the true traffic is obtained by multiplying the normalized traffic rate by the link capacity in bits per second.

On Figure 7, 8, we validate the analytical model by providing a comparison with measurements on the CIGALE packet switching network.

Consider first the measurement results presented on Figure 5. The values are $\pi_1 = 37$ bytes (27 bytes of header, plus 10 bytes of text), and π_2 taking the values 27, 37, 67, 117, 227, 282 bytes. What is shown is the total traffic in one direction. $\hat{D}_1$ and in the other $\hat{D}_2$ in bits per second. $\hat{D}_1$, $\hat{D}_2$ include the acknowledgements.

We also show the measured number of acknowledgements per second A_m, and the theoretical values A_t computed from the following formula :

(4) $$A_t = \frac{\hat{D}_1 - 56\, A_t}{8\, \pi_1}$$

where 56 is the number of bits in an acknowledgement. Notice that we could also use the formula

(5) $$A_t = \frac{\hat{D}_2 - 56\, A_t}{8\, \pi_2}$$

The small discrepancy between A_t and A_m indicates the existence of measurement errors. In Figure 6 we show the useful traffic D_1', D_2' given in bits/second, where

(6) $$D_1' = \hat{D}_1 - 56\, A_m$$

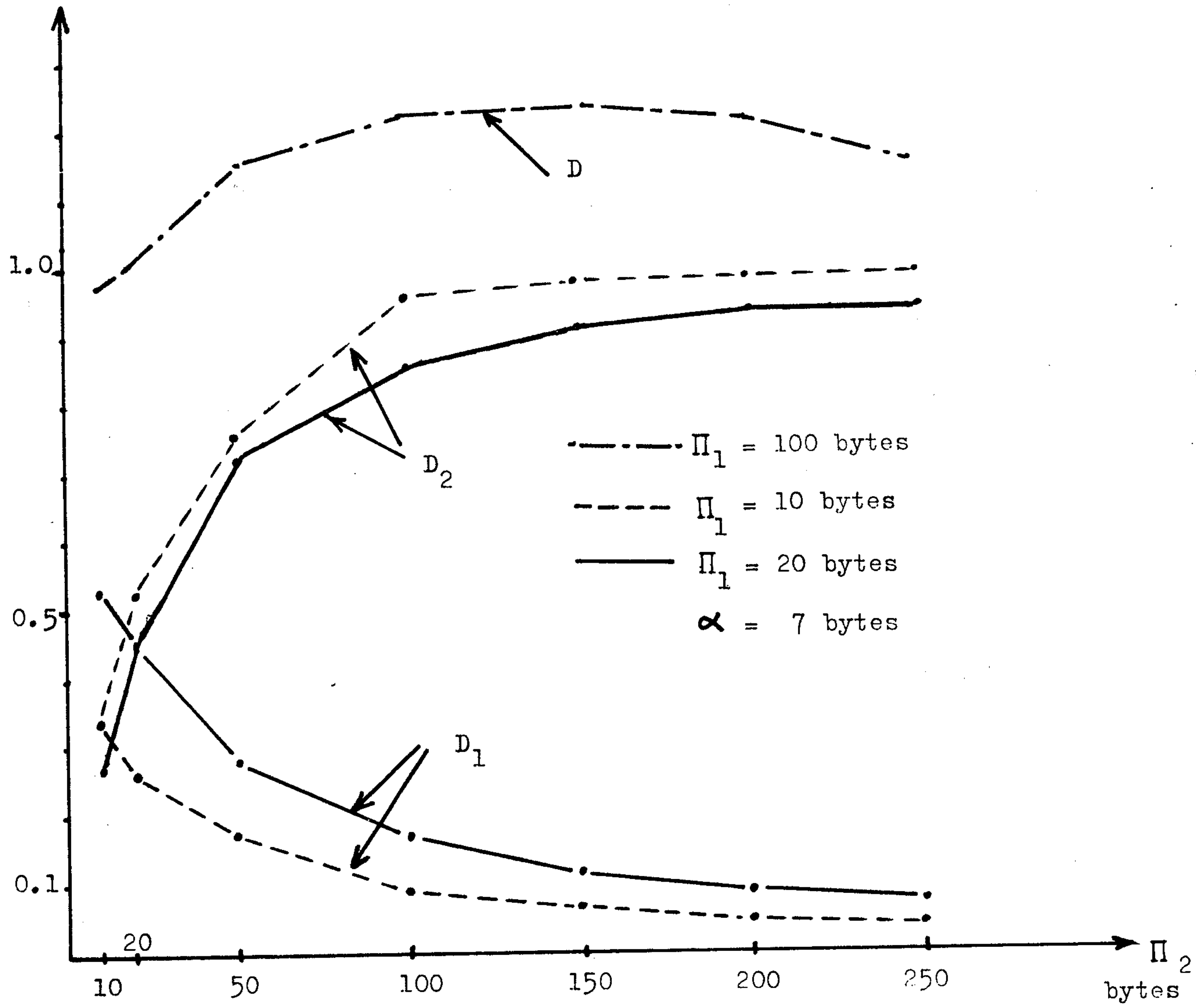

D
1.0
D_2
Π_1 = 100 bytes
Π_1 = 10 bytes
Π_1 = 20 bytes
α = 7 bytes
0.5
D_1
0.1
20
10
50
100
150
200
250
Π_2
bytes

FIGURE 6

π_2 bytes	27	37	67	117	177	227	282
A_m ACKS/sec.	41.6	41.6	27.3	17.1	12	9.7	7.9
A_t ACKS/sec.	41.5	40.6	26.2	17.33	11.8	9.66	8.09
D_1' bits/sec.	14320	12000	8000	5100	3600	2800	2300
D_1 bits/sec.	12590	11366	7400	5345	3733	2968	2422
D_2' bits/sec.	10720	12000	14700	16250	16800	17500	17800
D_2 bits/sec.	9185	11366	15401	16906	17849	18225	18462
D' bits/sec.	25040	24000	22700	21350	20400	20300	20100
D bits/sec.	21775	20442	22801	22251	21582	21193	20884
$\hat{D}_1$ bits/sec.	14600	14300	16200	17200	17400	18093	18270
$\hat{D}_2$ bits/sec.	11000	14300	9500	6040	4212	3342	2731

π_1 = 37 bytes, π = 7 bytes

Figure 7

π_2 bytes	27	37	67	117	177	227	282
A_m ACKS/sec.	27.3	27.3	27.3	17.1	12	9.7	7.9
A_t ACKS/sec.	27.35	26.7	26.72	17.36	12.07	9.55	7.84
D_1' bits/sec.	14665	14631	14297	10143	7054	5580	4578
D_1 bits/sec.	15740	14764	11970	8659	6361	5159	4269
D_2' bits/sec.	5711	7845	14282	17167	17809	17969	18231
D_2 bits/sec.	6345	8154	11970	15148	16790	17488	17935
D' bits/sec.	20375	22476	28576	27310	24863	23549	22809
D bits/sec.	22085	22918	23940	23807	23151	22647	22204
$\hat{D}_1$ bits/sec.	16194	16160	15823	10280	7150	5658	4641
$\hat{D}_2$ bits/sec.	7240	9374	15811	17304	17905	18047	18294

π_1 = 67 bytes , α = 7 bytes

FIGURE 8

(7) $D_2' = \hat{D}_2 - 56\ A_m$

For the links being measured, the maximum capacity is 19 200 bits/second. In Figure 7 we also show the theoretical traffic obtained from equations (1), (2), (3). We see that the maximum error in the model prediction is 14 % (for the case $\pi_1 = 27$), while generally the error is of the order of 5 %. Similar results for other sets of measurements are given on Figure 8., where the largest error is 16 %.

The high degree of accuracy of our model predictions confirm the validity of our simple mathematical model as well as its robustness : in the measurements all packet lengths are constant (and not exponential), and software wait times exist. The agreement between the theory and the measurements may be due to the fact that due to lack of synchronisation between nodes : a jitter is introduced which is perceived by the "other" node as a random packet length.

If we examine the data of Figure 5 we obtain that D, the total two-way throughput has a definite maximum value which is in the neighbourhood of $\pi_2 = 67$ bytes. This indicates that if we have $\pi_1 = 37$ we should set $\pi_2 = 77$ in order to maximize the total throughput. This does not seem to have an intuitive explanation and motivates our optimization study of the following section. We also notice that the maximum value of D, which is 22 801 bits/second, is well below the 38 400 bits/second which is the theoretical upper bound of the two-way link in this case. This point merits some analysis.

4. Packet size for throughput maximization

We have already noticed that D_1 is an increasing function of x, while D_2 is a decreasing function of $x (= \pi_1/\pi_2)$. We shall presently see that D given by (1) :

(8) $$D = \frac{2 + x + x^{-1}}{1 + x + x^{-1} = y(1 + x)}$$

has a maximum as a function of x. It is easily obtained by taking the derivative of D with respect to x and setting it to zero. D is maximized by

(9) $$x^* = \frac{1 - y}{1 + y}$$

and its maximum value is

$$(10) \qquad D^* = \frac{4}{3 + y(2 - y)}$$

assuming $y < 1$, i.e. that the length of the acknowledgement is shorter than the average length of a packet. From (9) and (10) we have that :

- the maximum throughput cannot exceed 2/3 of two-way channel capacity (obtained when $y = 0$) ; this is similar to Pouzin's predictions [5].
- when $y > 0$, the optimum value of π_2 is larger than π_1. That is, the system is optimized when the traffic is not identical in both directions.

Let us apply these results to the example of Figure 7 ; we see that the optimum value of π_2 computed from (9) is 54.35 bytes, and $D^* = 22\,978$ bits/second. On Figure 8, the optimum value of π_2 is close to 67 bytes and the corresponding value of the traffic is 22 700 bits/second.

On Figures 9 and 10 we illustrate the theoretical results of this optimization study.

5. Long constant packet lengths

In the previous analysis we have examined the case of asymmetric traffic with exponentially distributed packet lengths. This model seems to be satisfactory when either :

(i) packets in each direction are of different length (most of the measurements given pertain to this case), or

(ii) packets in both directions are of equal length, but are small enough so that the random fluctuations introduced by the system become significant.

Our previous model does not seem to account accurately for the case when both nodes are transmitting long packets of equal length. This is the case we will treat here.

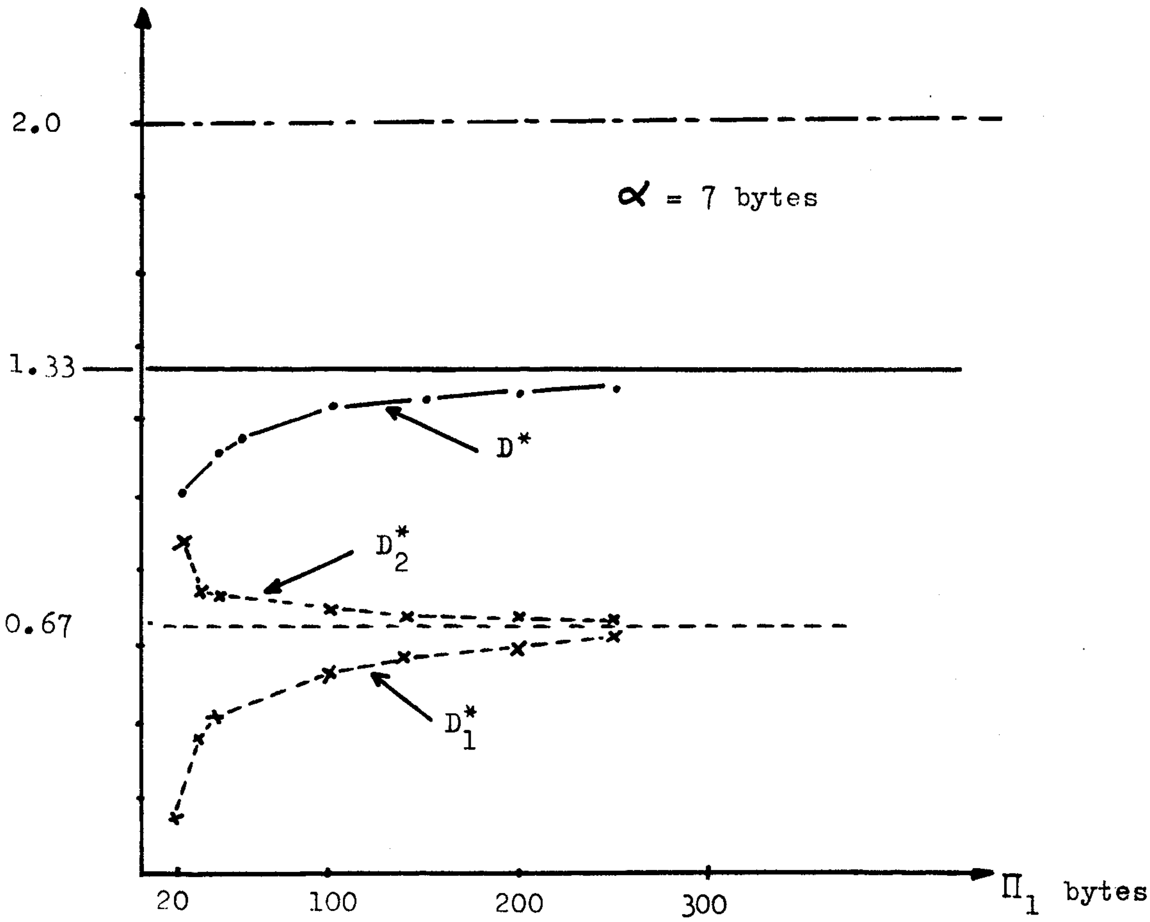


FIGURE 9

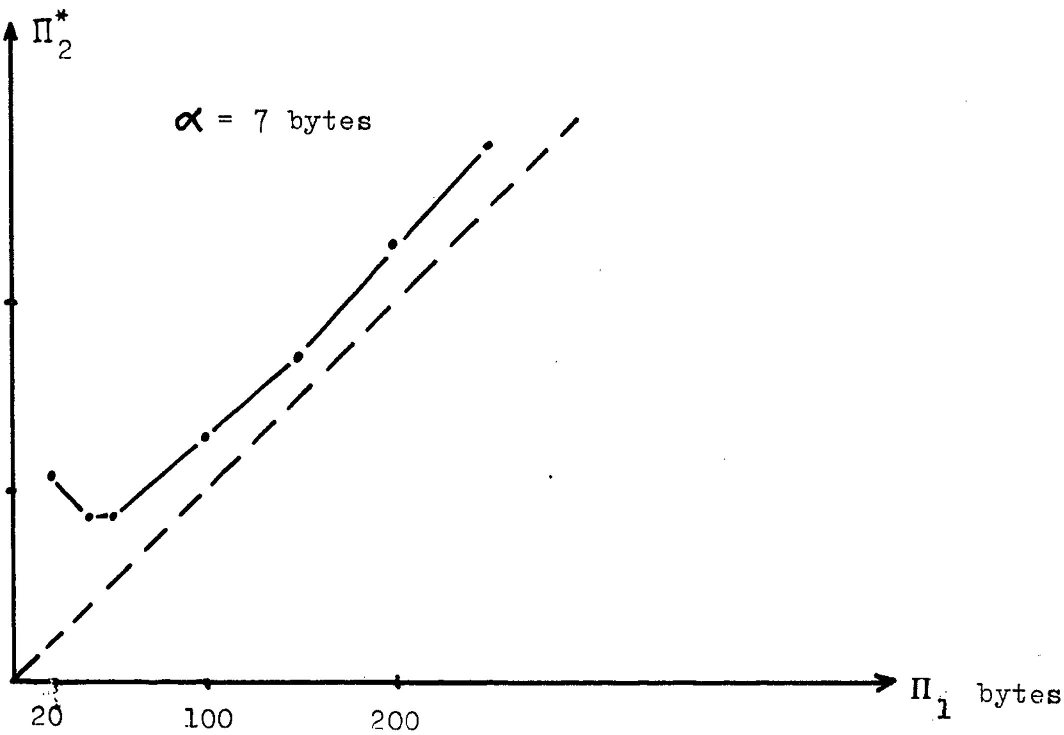


FIGURE 10

Suppose that $\pi \equiv \pi_1 = \pi_2$, and that packets transmitted in each direction are of constant length. If π is sufficiently large, it is reasonable to suppose that the time spent by the system in states PW or WP (see Figure 1) can be neglected. The state transition diagram is now given on Figure 11.

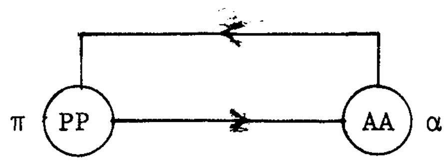


FIGURE 11

In this case

$$D_1 = D_2 = \frac{\pi}{\alpha + \pi} \quad , \quad D = D_1 + D_2 = \frac{2\pi}{\alpha + \pi}$$

For instance, for $\pi = 282$ we have $D_1 = D_2 = 18\,735$ bits/sec. (if channel capacity is 19 200 bits/sec.). A measurement experiment yielded $D_1' = D_2' = 17\,200$ bits/sec. The difference between the theoretical prediction and the measurement result can be explained by the existence of software wait times or of propagation delays. This phenomenon can be represented by a W state in the transition PP to AA and a W' state in the transition AA to PP ; let ω and ω' be the times spent in W and W', respectively. We shall provide an estimate of $\omega + \omega'$ by comparing the theoretical and experimental results. We now must have

$$D_1' \cong \frac{\pi}{\alpha + \pi + (\omega + \omega')} \times 19.200 \text{ bits/sec.}$$

so that $(\omega + \omega') = 25.8$ bytes. That is, $(\omega + \omega')$ is the time equivalent to the transmission time of 25.8 bytes along the line. In time units we have $(\omega + \omega') = 25.8 \times 8/19.200 = 10.75$ milliseconds. This time would include all software times in the nodes as well as propagation delays.

6. Conclusions

In this paper we have given a model of the stop and wait protocol (TMM in CIGALE [2]) and have validated it by using measurements on the actually implemented system.

We then show that the maximum throughput of this protocol cannot exceed 2/3 of total two-way throughput for exponentially distributed packet lengths. We also obtain the optimum value of relative packet sizes.

The case of long constant packets of equal length has been presented, and we have provided an estimate of software wait times.

In the Appendix we have presented a general model of the protocol, in which the exponential assumptions concerning packet lengths are not made.

This analysis shows clearly that the send and wait protocol is unable to exploit fully the capacity of the communication links if packets are not of fixed and equal length in both directions, and long enough to render the randomizing effect of software wait times and propagation delays negligible. Our study also shows the effectiveness of simple mathematical models for the study of network protocols.

APPENDIX

We shall treat the case where the packet lengths are distributed according to some general distribution. Let $F_1(t)$, $F_2(t)$ be the probability distribution function for packets transmitted from 1 to 2, or 2 to 1, respectively. Then the probability, in the state transition diagram of Figure 1, of the transition "PP to PW" is :

$$p_1 \equiv \int_0^\infty dF_2(t)\ [1 - F_1(t)]$$

while the transition probability "PP to WP" is :

$$p_2 \equiv \int_0^\infty dF_1(t)\ [1 - F_2(t)]$$

The average time spent in state PP is :

$$\tau \equiv \int_0^\infty t[dF_2(t)\ (1 - F_1(t)) + dF_1(t)\ (1 - F_2(t))]$$

The average time spent in state PW is :

$$\tau_1 \equiv \int_0^\infty dx[1 - \int_0^\infty dF_2(t)\ (F_1(t+x) - F_1(t))]$$

while the average time spent in stape WP is :

$$\tau_2 \equiv \int_0^\infty dx[1 - \int_0^\infty dF_1(t)\ (F_2(t+x) - F_2(t))]$$

Therefore

$$D_1 = \frac{\tau + p_1\tau_1}{\tau + p_1\tau_1 + p_2\tau_2 + \alpha} \quad , \quad D_2 = \frac{\tau + p_2\tau_2}{\tau + p_1\tau_1 + p_2\tau_2 + \alpha}$$

The results of Section 3 are obtained if we assume that

$$F_1(t) = 1 - e^{-t/\pi_1}, \quad F_2(t) = 1 - e^{-t/\pi_2}$$

REFERENCES

[1] J.L. GRANGE - "CIGALE the packet switching subnetwork of CYCLADES" - Journées Informatique Médicale - Toulouse - Mars 1976 - Réseau CYCLADES, MIT 617.

[2] L. POUZIN - "Presentation and major design acpects of the CYCLADES computer network" - 3rd Data Comm. Symp. TAMPA - Nov. 1973 - 80-85 - Réseau CYCLADES SCH 511.

[3] N. NAFFAH - "Spécification d'interface TMM avec le réseau CIGALE" - Déc. 1975 - Réseau CYCLADES TRA 521.

[4] S. ROSS - "Applied probability models with optimization applications" - Holden-Day - San Francisco - 1970.

[5] L. POUZIN - "Efficiency of full-duplex synchronous data burk procedures" - IFIP WG6.1 - INW 35 - June 1973 - Réseau CYCLADES IRA 510.

[6] G. FAYOLLE, E. GELENBE - G. PUJOLLE - "An analytic evaluation of the performance of the stop-and-wait prptocol" - Rapport de Recherche n° 207 - IRIA-LABORIA - Dec. 1976.