CONGESTION CONTROL IN COMMUNICATION NETWORKS FOR COMPLEX SYSTEMS

Melha Bitam[1] and Hassane Alla

Laboratoire d'Automatique de Grenoble
Domaine Universitaire BP 46
38402 Saint Martin d'Hères, France

Abstract: Large scaled distributed systems require many connections between various elements which constitute them. Accordingly, they use communication networks. With the growth of communication networks, severe congestion problems appeared. To avoid dropping packets and then guarantee the needs of distributed systems in terms of quality of service, some algorithms were put into TCP. Some interesting results can be obtained by studying the loop time of data in transit in a transmission line. This paper presents a method to determine the buffer size from the data loop time. The whole model of a transmission line subjected to algorithms of control is hybrid. Thus, the chosen modeling tool is a hybrid Petri nets.

Keywords: Distributed systems, hybrid model, TCP/IP and congestion avoidance.

1. INTRODUCTION

Distributed systems employ communication networks to exchange the data which they need. However, transmission networks are very requested all over the world and this generates lines strongly charged and data losses. Some congestion control algorithms, essentially established by Jacobson (Jacobson, 1988) were put into TCP traffic. These algorithms are: slow start, congestion avoidance, fast retransmit and fast recovery algorithms (Jacobson, 1988; Stevens, 1997). This control acts on emitters by decreasing flows when congestion occurs. Congestion is detected either from timeouts or from three duplicate acknowledgements.

The transmission loop time is the time passed between sending the data and receiving its acknowledgement. This loop time depends closely on the charge of the transmission line and especially on the charge of buffers. In this paper, we propose a new method to control congestion using the data loop time in a communication line. The data in buffer is managed by fifo (first in first out) queuing. Thus, when a new packet enters the buffer, it waits until all packets present in the buffer leave it. This waiting time depends on the charge of the buffer. We use information "waiting time" to deduce the buffer charge and then decide the appropriate data quantity that we can send without causing the buffer overflows.

In the literature, we did not find a complete model of an Internet connection, namely the emitters, the canals, the routers and the receivers. This motivated us to develop a complete model which allows us to study the influence of networks in terms of delays and messages losses in order to evaluate the behavior and the performances of the transmission line (Bitam, 2004; Bitam and Alla, 2005).

In a transmission line, messages are sent with high speed and this can be compared to continuous flows. The system also contains discrete behaviors such as protocols decisions. Then, the communication network can be seen as a hybrid system with continuous flows. We choose hybrid

[1] Author for correspondence : melha.bitam@lag.ensieg.inpg.fr

PN because the continuous part provides a good approximation of certain discrete phenomena that can be compared to continuous flows, here flow of messages (Gershwin and Schick, 1980). Finally, because of the overlapping of several timers in a transmission, the final model is colored hybrid PN (Bitam and Alla, 2005, Bitam, 2004).

This paper is organized as follows: Section 2 presents the algorithms of Jacobson (Stevens, 1997; Jacobson, 1988): slow start, congestion avoidance and the timer algorithm. We have based our model on these algorithms because they are the most frequently used ones. Section 3 presents briefly the hybrid model of a transmission line under TCP/IP algorithms. Section 4 presents our contribution in congestion control. First, we will calculate the loop time without considering fifo queuing in routers. Then, we will compare this value with the real loop time (with considering fifo queuing). Finally, we will exploit this information to control the quantity of data sent which can not exceed the capacity of the line sources. The conclusion and a description of our future work are given in section 5.

2. TCP/IP CONNECTIONS AND CONTROL CONGESTION

Several algorithms established in TCP/TP defined the phases of congestion control. First, the capacity of the line is tested by increasing its sent data gradually. This is the role of slow start algorithm. The size of sent data, *congestion window*, is noted *cwnd*. It varies in time and evolves exponentially after each success transmission. An acknowledgment (*ack*) is sent by the receiver to the emitter to inform that data is well received. This acknowledgment allows the emitter to continue transmission. When the sent data reached a fixed threshold, named *slow start threshold* (*ssth*), we consider that the line is full and we move on to the next stage which is the congestion avoidance. During this phase, the evolution of *cwnd* is linear (figure 1). When congestion occurs, the size of sent data is reinitialized at the minimum size and the system restart with slow start algorithm (figure 1). When a packet is sent, the emitter waits the acknowledgement for a given time. This time is given by a timer named *retransmit timer* usually designed *Tempo*. At the beginning data sending, *Tempo* is triggered. If the acknowledgement is not arrived at the expiry of *Tempo*, the emitter deduced that the data is dropped and reinitialized the transmission (Jacobson, 1988).

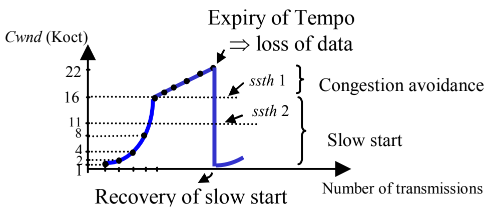


Fig. 1. Slow Start and Congestion Avoidance algorithms

3. HYBRID PN MODELLING

The advantage of a Petri nets (PN) model is to highlight all the elements of the network: flows in canals and routers and discrete orders of protocols. In the same time, PN models still simple and intuitive. Figure 2.a represents the line considered to transmit our data. It is composed of one emitter E connected to one receiver R via a router Rt and two transmission canals (the detailed model is given in Bitam, 2004; Bitam and Alla, 2005).

Continuous flow model: The most important part of line dynamics is its router dynamic. It behaves such a tank: it is fed at the speed V_1 and is emptied at the speed V_2 (figure 2.b). The maximal capacity of the buffer is C. This limitation is given by the complementary place initially marked C.

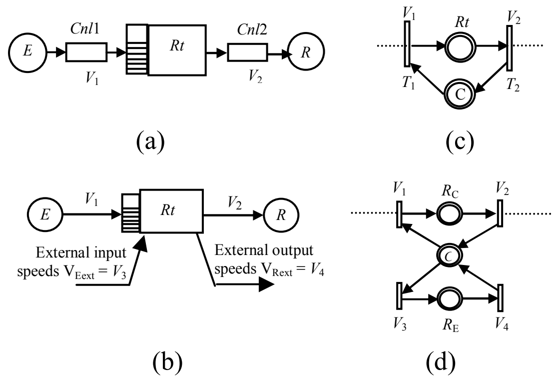


Fig. 2. a) An isolated line. b) The router shared by external emitters. Continuous PN model of c) a router. d) the router shared by external emitters.

We considered that an unknown number of emitters are connected to the line and send information to an unknown number of receivers via the router Rt which shares its capacity C. The sum of the input and output speeds of these external connections can be seen as only one input speed V_3 and one output speed V_4 (figures 2.c and 2.d).

Discrete protocol model: In PN model of figure 3.a, the place R represents the receiver and the weight m of the arc $R \rightarrow T_5$ represents the segment size. When the marking of place R reaches m, we generate an acknowledgment by putting a mark in place *ack*.

The exponential evolution in slow start and linear one in congestion avoidance are represented in figure 3.b and 3.c respectively.

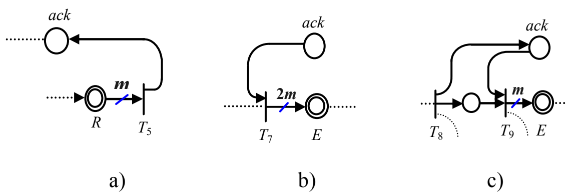


Fig. 3. a) Generate acknowledgements. PN model of *Cwnd* evolution during: b) slow start. c) congestion avoidance.

The whole model of a transmission line merged in Internet environment subjected to TCP/IP protocols is given in (Bitam, 2004; Bitam and Alla, 2005).

4. CONGESTION CONTROL

At emitter, we have not information about the buffer load in the gateway. If we have this information, we could control the flow sent into the line and thus avoid congestion.

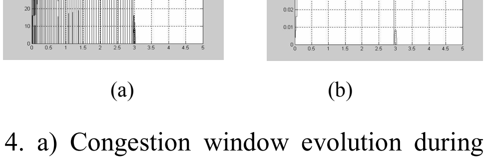
When we receive data acknowledgment, we calculate data the loop time. This loop time depends closely on the buffer charge. In fact, data leaves from router according to fifo queuing. So, the last packet which enters in the buffer must wait for the time required the other data present in the router to leave it. The time spent by a packet in the buffer is noted δ_t.

If we don't take into consideration fifo queuing at the routers, packets will not be delayed by the other data in the buffer. So, things happened as if the spending time in the buffer is null. Obviously, this does not correspond to the reality, but a comparison between results obtained with fifo queuing and those obtained without fifo queuing allows us to deduce some interesting conclusions. This is the originality of this work. Let us consider first the case where the fifo queuing is not taken into account.

4.1. Without considering Queuing Management file

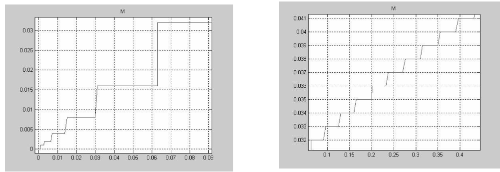
Simulations were done by using Matlab on the whole model of a transmission line merged in Internet. We consider first, a model which doesn't take into consideration fifo queuing. Some simulations results and a performance evaluation are given in (Bitam and Alla, 2005). Figures 4.a and 4.b respectively represent one case of the congestion window evolution and its corresponding loop time (*M*) obtained from these simulations.

We can easily remark in these two figures that the loop time depends closely on the size of the congestion window sent. The congestion window evolves firstly exponentially then linearly in time. Let us look in more details the form of the loop time *M*. Figures 5.a and 5.b represent *M* in exponential phase and in linear phase of the congestion window respectively. We can also remark that the loop time also evolves firstly exponentially then linearly.



(a) (b)

Fig. 4. a) Congestion window evolution during a whole transmission. b) Loop time evolution during a whole transmission.



a) b)

Fig. 5. Loop time evolution during a) Slow start evolution. b) Congestion avoidance evolution.

We noted M_i as successive values of M given by figure 5.a and 5.b. and we represent by t_i its corresponding occurrence time.

At exponential phase (figure 5.a), we have:

$$M_2 = 2*M_1,\quad M_3 = 4*M_1,\quad M_4 = 8*M_1,\ \ldots$$

$$M_i = 2^{i-1}M_1 \tag{1}$$

with M_1 the first value of M and M_i the ith one.

and

$$t_1 = M_1,$$
$$t_2 - t_1 = 2*t_1 \Rightarrow t_2 = 3*t_1,$$
$$t_3 - t_2 = 2*(t_2 - t_1) \Rightarrow t_3 = 7*t_1,$$
$$t_i = \left(2^{i-1} + 2^{i-2} + \ldots\ldots + 2 + 1\right)*t_1 \tag{2}$$

In linear phase (figure 5.a), we have:

$$M_{k+1} = M_k + M_1;$$
$$M_{k+2} = M_{k+1} + M_1 \Rightarrow M_{k+2} = 2*M_1 + M_k,\ \ldots$$
$$M_j = jM_1 + M_k \tag{3}$$

with M_k the latest value of M calculated during the slow start phase. So, M_{k+1} is the first value of M during congestion avoidance phase.

Assume that α_2 is the first time interval which separates two successive values of M during the congestion avoidance phase, then:

$$\alpha = t_k - t_{k-1} + t_1 = (2^{k-1}+1)t_1, \quad (\text{with } t_1 = M_1).$$

Thus,

$$t_{k+1} = \alpha + t_k,$$
$$(t_{k+2} - t_{k+1}) - (t_{k+1} - t_k) = t_1$$
$$\Rightarrow t_{k+2} = 2*\alpha + t_1 + t_k,$$
$$(t_{k+3} - t_{k+2}) - (t_{k+2} - t_{k+1}) = t_1$$
$$\Rightarrow t_{k+3} = 3*\alpha + 3*t_1 + t_k,$$
$$(t_{k+4} - t_{k+3}) - (t_{k+3} - t_{k+2}) = t_1$$
$$\Rightarrow t_{k+4} = 4*\alpha + 6*t_1 + t_k, \ldots$$

$$t_{k+j} = j*\alpha[j + (j-1) + \ldots + 2 + 1]*t_1 + t_k \tag{4}$$

Conclusion 1: Each of M_i values and t_i values depend only on the first value of M: M_1. Thus, if we have M_1, we can easily calculate all the other values of M when the transmission goes well.

As we mentioned above, these calculations don't take into account the fifo queuing management in the router. Let us consider now what happened if this file management is taken into account.

4.2. With Queuing Management file:

If the router is not managed by fifo queuing, the loop time M is calculated as explained previously. Now, if fifo queuing manages the router data, these data spend a time δ_t to leave the router. The real loop time can't be calculated as above because the external speeds are not known in advance. So, the buffer charge also is unknown. The real loop time, noted M_f, can just be observed. Thus, the delay δ_t can also be observed and if we assume that there is not any other treatment in the router which can delayed the data anymore, δ_t will indicates the router charge. If δ_t is small, the charge is low, else the charge is high. The general forms of M and M_f are given in figure 6. The goal of this section is to compare these two values and to get the adequate conclusions.

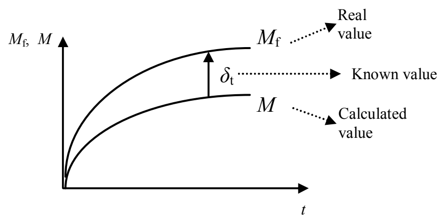


Fig. 6. Real and calculated loop time.

Case of an isolated line: In order to present ideas progressively, we first consider, the case of an isolated line, i.e., a line which is not merged in Internet environment. The considered line is given in figure 7.

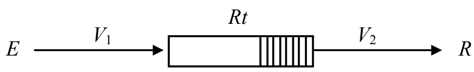


Fig. 7. An isolated connection.

We named *seg*, the last segment which arrived at the router (figure 8).

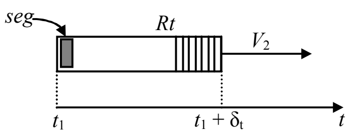


Fig. 8. Messages arrival in the router *Rt*.

With t_1: entry moment of *seg* in router and δ_t: time taken by *seg* to pass through the router.
It seems obvious that, at the moment t_1, δ_t depends on the output speed V_2 and on the data quantity in the router, named m_{Rt}.

$$\delta_t = \frac{m_{Rt}(t_1)}{V_2}$$

$$\Rightarrow m_{Rt}(t_1) = \delta_t V_2 \qquad (5)$$

$\Rightarrow \delta_t$ and V_2 are known, therefore the buffer charge, m_{Rt} is known too. Thus, this information became available for the emitter.

Conclusion 1: In the case of an isolated line, we obtained the precise value of the router charge at each moment t_1.

Note 1: A data is sent at moment t_1. When its acknowledgment is received, after a time M_f, we calculate m_{Rt}. Thus, this calculation can be carried out with each reception of an acknowledgment.

Case of a line merged in Internet environment: Messages coming from our transmission line E and other messages coming from the external sources E_{ext} are mixed in the router *Rt* (figure 9).

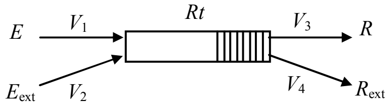


Fig. 9. A connection subjected to external environment.

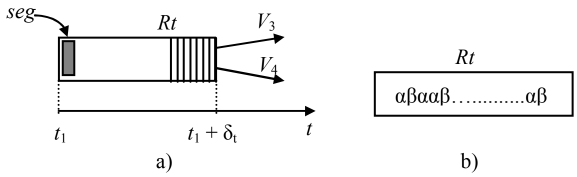


Fig. 10. a) Messages arrival in the router *Rt*. b) Example of router contents

We suppose in figure 11 that α and β are the messages coming from E and E_{ext} respectively.
The messages α go out the router at speed V_3 while the messages β go out the router at the speed V_4 (figure 10).

- *If output speeds V_3 and V_4 are taken constant*:

$$\delta_t = \frac{\alpha}{V_3} + \frac{\beta}{V_4} + \frac{\alpha}{V_3} + \frac{\alpha}{V_3} + \frac{\beta}{V_4} + \dots\dots\dots + \frac{\beta}{V_4}$$

$$\delta_t = \underbrace{\frac{m_{Rt\,line}(t_1)}{V_3}}_{\delta_{t\ line}} + \underbrace{\frac{m_{Rt\,ext}(t_1)}{V_4}}_{\delta_{t\ ext}} \qquad (6)$$

With: $m_{Rt\ line}$ and $m_{Rt\ ext}$ the router markings coming from E and E_{ext} respectively.

Thus, we obtained the following equation system:

$$\begin{cases} a\, m_{Rt\,line}(t_1) + b\, m_{Rt\,ext}(t_1) = \delta_t \\ m_{Rt\,line}(t_1) + m_{Rt\,ext}(t_1) \le C \end{cases} \qquad (7)$$

With: $a = \frac{1}{V_3}$, $b = \frac{1}{V_4}$ and C is the maximal capacity of the router.

Conclusion 3: If the obtained equation system 7 has a solution, the router is not charged yet and we can continue to send data. Otherwise, the router is full.

Conclusion 4: As said in note 1, this calculation is made with each acknowledgment reception. Thus, we can know, after the reception of only one acknowledgment, that the router if full. It is a good improvement of congestion detection, because usually it is necessary to wait the arrival of 3 duplicates acknowledgments (Fall and Floyd, 1996), or worst to wait the expiry of a timer (Jacobson, 1988) to conclude a loss.

- *If the output speeds V_3 and V_4 vary in time*: Let us first assume that V_3 and V_4 change only once in time interval t_1 and $t_1 + \delta_t$:

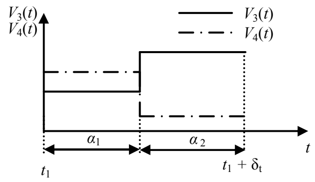


Fig. 12. Single speed change between t_1 and $(t_1 + \delta_t)$.

Note 2: Two types of events control the dynamics of global system: the first one is when marking of a continuous place becomes null and the second one is when the external speed changes its values in any known time interval. The algorithm of global system supervises the arrival of one of these two events to recalculate the system dynamics (Bitam and Alla, 2005). Thus, as long as none of these two events occurs, all the system speeds remain constant. The moments of dynamics changes are known, therefore the speeds values are known at any moment.

⇒ Since t_1 and δ_t are known, thus:

$$\left.\begin{array}{l} V_3(t_1), V_4(t_1) \\ V_3(t_1+\alpha_1), V_4(t_1+\alpha_1) \\ \alpha_1, \alpha_2 \end{array}\right\} \text{ are also known.}$$

At $t \in [\, t_1, \quad t_1 + \alpha_1]$: (figure 10)

$$\delta_{t1} = \frac{m_{Rt\,line}(t_1)}{V_3(t_1)} + \frac{m_{Rt\,ext}(t_1)}{V_4(t_1)}$$

At $t \in [\, t_1 + \alpha_1, \quad t_1 + \delta_t]$: (figure 13)

$$\delta_{t2} = \frac{m_{Rt\,line}(t_1+\alpha_1)}{V_3(t_1+\alpha_1)} + \frac{m_{Rt\,ext}(t_1+\alpha_1)}{V_4(t_1+\alpha_1)}$$

with:

$$m_{Rt\,line}(t_1+\alpha_1) = m_{Rt\,line}(t_1) - \alpha_1\, V_3(t_1),$$
$$m_{Rt\,ext}(t_1+\alpha_1) = m_{Rt\,ext}(t_1) - \alpha_1\, V_4(t_1).$$

Since, $\delta_t = \delta_{t1} + \delta_{t2}$, then:

$$\delta_t = \left.\frac{m_{Rt\,line}(t_1)}{V_3(t_1)} + \frac{m_{Rt\,line}(t_1+\alpha_1)}{V_3(t_1+\alpha_1)}\right\} \ldots\ldots \delta_{t\,line}$$
$$\left. + \frac{m_{Rt\,ext}(t_1)}{V_4(t_1)} + \frac{m_{Rt\,ext}(t_1+\alpha_1)}{V_4(t_1+\alpha_1)}\right\} \ldots\ldots \delta_{t\,ext}$$

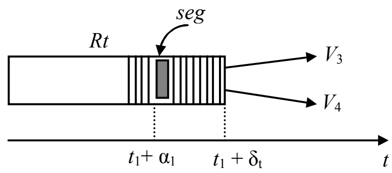


Fig. 13. Messages arrival in the router *Rt*.

After calculations:

$$\delta_{t\,ext} = \left[\frac{1}{V_4(t_1)} + \frac{1}{V_4(t_1+\alpha_1)}\right] * m_{Rt\,ext}(t_1) - \alpha_1 * \frac{V_4(t_1)}{V_4(t_1+\alpha_1)}$$
$$\delta_{t\,ext} = \left[\frac{1}{V_4(t_1)} + \frac{1}{V_4(t_1+\alpha_1)}\right] * m_{Rt\,ext}(t_1) - \alpha_1 * \frac{V_4(t_1)}{V_4(t_1+\alpha_1)}$$

$\delta_{t\,line}$ has the form: $a\, m_{Rt\,line}(t_1) + b$ and $\delta_{t\,ext}$ has the form: $a'\, m_{Rt\,ext}(t_1) + b'$. $\delta_t = \delta_{t\,line} + \delta_{t\,ext}$, thus:

$\delta_t = a\, m_{Rt\,line}(t_1) + a'\, m_{Rt\,ext}(t_1) + b + b'$.

with: a, a', b, b' known constants.

We obtained the following equations system:

$$\begin{cases} a\, m_{Rt\,line}(t_1) + a'\, m_{Rt\,ext}(t_1) + c = \delta_t \\ m_{Rt\,line}(t_1) + m_{Rt\,ext}(t_1) \le C \end{cases} \tag{8}$$

with $c = b + b'$.

As mentioned in the previous case (equation system 7), the solutions of this system represent the buffer charge. Thus, we obtain identical results to those explained in conclusions 3 and 4.

Let us assume now that there are several changes of output speeds V_3 and V_4 between t_1 and $t_1 + \delta_t$. After calculations which we will not detail, we obtain:

$$\delta_{t\,line} = m_{Rt}(t_1)_{line} * \sum_{i=0}^{n-1} \frac{1}{V_3(t_1 + \sum_{j=0}^{i} \alpha_j)}$$
$$- \sum_{i=1}^{n-1} \left[\frac{1}{V_3(t_1 + \sum_{j=0}^{i} \alpha_j)} * \sum_{j=1}^{i} \alpha_j * V_3(t_1 + \sum_{l=0}^{i-1} \alpha_l) \right]$$
$$\delta_{t\,ext} = m_{Rt}(t_1)_{ext} * \sum_{i=0}^{n-1} \frac{1}{V_4(t_1 + \sum_{j=0}^{i} \alpha_j)}$$
$$- \sum_{i=1}^{n-1} \left[\frac{1}{V_4(t_1 + \sum_{j=0}^{i} \alpha_j)} * \sum_{j=1}^{i} \alpha_j * V_4(t_1 + \sum_{l=0}^{i-1} \alpha_l) \right]$$

With n an integer which represents the number of speeds changes.

$\delta_{t\,line}$ has the form: $a\, m_{Rt\,line}(t_1) + b$ and $\delta_{t\,ext}$ has the form: $a'\, m_{Rt\,ext}(t_1) + b'$ with: a, a', b, b' known constants.

Thus: $\delta_t = a\, m_{Rt\,line}(t_1) + a'\, m_{Rt\,ext}(t_1) + b + b'$, and:

$$\begin{cases} a\, m_{Rt\,line}(t_1) + a'\, m_{Rt\,ext}(t_1) + c = \delta_t \\ m_{Rt\,line}(t_1) + m_{Rt\,ext}(t_1) \le C \end{cases} \tag{9}$$

with $c = b + b'$.

For this equations system, the same conclusions can be made as for the equations 7 and 8.
Figure 14.a and 14.b define the area of possible values for $m_{Rt\text{line}}$ and $m_{Rt\text{ext}}$ according to the lines $m_{Rt\ line}(t_1) + m_{Rt\ ext}(t_1) = C$ and $a\ m_{Rt\ line}(t_1) + b\ m_{Rt\ ext}(t_1) = \delta_t$ respectively. The final solutions are given by the intersections of these two areas. Figure 14.c, 14.d, 14.e and 14.f represent the fourth possible cases for intersection of the figures 14.a and 14.b areas. In the first case (figure 14.c) there are surely no losses, in the last case (figure 14.f) there are surely losses, the 2 other cases present possibilities of losses in the router.

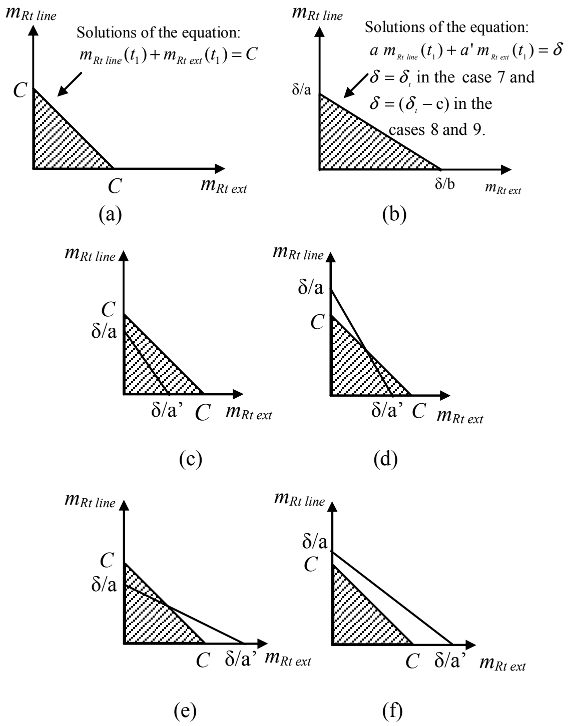


Fig. 14. Solutions of the equations a) $m_{Rt\ line}(t_1) + m_{Rt\ ext}(t_1) = C$,
b) $a\ m_{Rt\ line}(t_1) + a'\ m_{Rt\ ext}(t_1) = \delta$. Global solutions when c) $\delta / a' \prec C \ and\ \delta / a \prec C$
d) $\delta / a' \prec C \ and\ \delta / a \succ C$,e) $\delta / a' \succ C \ and\ \delta / a \prec C$,
f) $\delta / a' \succ C \ and\ \delta / a \succ C$.

A more detailed study of conclusions to be drawn from the obtained equations systems is in hand. The next objective of this work is to locate the state of router load. The final results will be the subject of the next article.

5. CONCLIDING REMARKS

In this paper, we treated the problem of congestion control in TCP/IP transmission.
Firstly, hybrid PN model of transmission line merged in Internet environment and subjected to TCP/IP protocols allows us to carry out some simulations (Bitam and Alla, 2005; Bitam, 2004) and extract some performances (Bitam and Alla, 2005). The same model allows us, as explained in this paper, to observe the evolution of loop time *M* when messages can leave the router immediately, limited only by the output capacity of the router. We have observed that when the congestion window evolves exponentially, *M* also evolves exponentially and when the congestion window evolves linearly, *M* also evolves linearly. Finally, the evolution of *M* in time was expressed by a set of equations. However, in real systems, messages spent a time in router since they are managed in a fifo discipline. We showed that the comparison between the first value of the loop time without fifo queuing and the real value considering fifo queuing is very interesting. Indeed, the difference between these two values expressed exactly the buffer charge.
Secondly, we considered the case of one isolated connection and we have easily expressed the buffer charge according to the delay δ_t and the output router speed rate. Then, we considered a connection merged in Internet environment with variable output speed rates. We obtained an equations system whose solutions represent the whole possible values of the router load. If this equation system has not any solution, it means that the router overflows. Moreover, this congestion detection is made after the reception of only one acknowledgment, while other detection methods usually wait the arrival of 3 duplicate *ack* to conclude a loss (Fall and Floyd, 1996), or worst wait the expiry of *Tempo* (Jacobson, 1988).
The futur objective is to evaluate the state of the router load from the emitter and thus control transmissions without waiting for a dropped packet to react to the congestion. In our future work, we will focus on the equation system which we obtained to deduce the router charge and establish the appropriate control to avoid congestion.

REFERENCES

M. Bitam, "Modélisation et analyse du comportement d'une ligne de communication TCP/IP", *Conférence Internationale Francophone d'Automatique CIFA2004*, Douz, Tunisia, November 2004.

M. Bitam and H. Alla, "Performance Evaluation of a TCP/IP Transmission Using Hybrid Petri Nets" *IEEE International Conference on Computer Systems and Applications (ICCSA)*, Cairo, Egypt, July 2005.

K. Fall and S. Floyd, "Simulation-based Comparisons of Tahoe Reno and Sack TCP", *Computer Communication Review*, July 1996, **vol. 26 No 3**, pp. 5-21.

Gershwin and Schick, (1980). Continuous Model of an Unreliable Two-Stage Material Flow System With a Finite Interstage Buffer, *technical report MIT LIDS-R-1032*.

V. Jacobson, "Congestion Avoidance and Control". *Proceedings of SIGCOMM '88*, August 1988, pp. 314-329.

W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery Algorithms", *RFC 2001*, January, 1997.