# Fairness analysis of TCP/IP

E. Altman, C. Barakat, E. Laborde, P. Brown, D. Collange

INRIA Sophia-Antipolis
2004 route des Lucioles, BP 93
06902 Sophia-Antipolis Cedex

E-mail: altman@sophia.inria.fr

*DEC, 2000*

# OUTLINE

**I.** Introduction

**II.** The Model

**III.** Throughput Computation

**IV.** Numerical Results

**V.** Simulation

**VI.** Conclusions

# Previous Approaches to Compute Bandwidth Sharing

- 1. Use the square root formula for the throughput of connection $k$:

$$Thp_k = \frac{1}{T_k}\sqrt{\frac{c}{p_k}} \quad (1)$$

T. Ott, J. Kemperman, and M. Mathis, "The stationary behavior of the ideal TCP congestion avoidance".

The formulae is obtained through heuristic arguments.

In particular, it assumes that the losses are not affected by the connection $k$.

It does not does not say how to obtain $p_k$.

- 2. Use the synchronization assumption: when a congestion event (loss) occurs, all other connections also suffer losses.

P. Brown, "Resource sharing of TCP connections with different round trip times", *IEEE Infocom*, Mar 2000.

T.V. Lakshman and U. Madhow, "The performance of TCP/IP for networks with high bandwidth-delay products and random loss", *IEEE/ACM Transactions on Networking*, Jun 1997.

- Under this assumption it had been shown that $Thp_k$ is inversely proportional to $T_k^\alpha$ for some $\alpha \in (0,1)$.

- The synchronization assumption is valid only for connections with similar RTT, and for a drop tail buffer.

L. Zhang, S. Shenker, and D.D. Clark, "Observations on the Dynamics of a Congestion Control Algorithm: The Effects of Two-Way Traffic", *ACM SIGCOMM*, Sep 1991.

- In that case, when the buffer fills, it still takes one RTT till the loss is detected. During this time, all connections keep sending packets at a high rate so all connections suffer losses.

INRIA

## Objectives:

- We wish to study General RTT.
- We do not wish to make synchronization assumptions.
- We shall consider RED buffers: losses occur before the buffer fills. This will avoid synchronization even for connections with similar RTT (advantage of Red).

INRIA

# The model

- We assume a TCP version that recovers from losses without time-outs
- Long persistent TCP transfers
- Upon a loss, the window is halved. For each received ack the window increases by $1/W_k$.
- Assumption: queueing delay small w.r.t. propagation delay (due to the use of RED). This results in a linear window increase between losses.
- We use a fluid model and consider for simplicity 2 connections:

$$\frac{dW_k(t)}{dt} = \frac{dW_k(t)}{dack_k} \times \frac{dack_k}{dt} = \frac{1}{W_k(t)} \times \frac{W_k(t)}{T_k} = \frac{1}{T_k}.$$

- Congestion (loss) occurs when the sum of throughputs $X_1 + X_2$ reach the available bandwidth $\mu$:

$$X_1(t_n) + X_2(t_n) = W_1(t_n)/T_1 + W_2(t_n)/T_2 = \mu. \quad (2)$$

$t_n :=$ $n$th congestion event.

**Conclusion: we may reduce the problem to a single dimension**

- The probability that a source $k$ ($k = 1, 2$) reduces its window at $t_n$:

$$p_k = X_k(t_n)/\mu = W_k(t_n)/(\mu T_k).$$

# Throughput Computation

We show that we obtain a semi-Markov reward process.

- If connection 1 is hurt by congestion at instant $t_n$, the next congestion will appear after a time,

$$t_{n+1} - t_n = \frac{T_1 T_2^2}{T_1^2 + T_2^2} \times \frac{W_1(t_n)}{2},$$

and the window size of connection 1 prior to this next congestion event will be equal to,

$$W_1(t_{n+1}) = \frac{T_1^2 + 2T_2^2}{2(T_1^2 + T_2^2)} \times W_1(t_n). \qquad (3)$$

- If connection 2 is hurt by the congestion, connection 1 continues to increase its window without reduction until the next congestion event which occurs after a time,

$$t_{n+1} - t_n = \frac{T_1^2 T_2^2}{2(T_1^2 + T_2^2)} \times \left(\mu - \frac{W_1(t_n)}{T_1}\right).$$

In this case, the window of connection 1 prior to the next congestion event will be equal to,

$$W_1(t_{n+1}) = \frac{T_1 T_2^2}{2(T_1^2 + T_2^2)} \times \left(\mu - \frac{W_1(t_n)}{T_1}\right) + W_1(t_n). \quad (4)$$

$W_1(t_n)$ is thus an embedded Markov chain taking values in $\mathcal{I} \subset [1, W_1^{Max}]$.

- We discretize this process and compute of the transition probabilities $P = (p_{ij})_{i,j\in\mathcal{I}}$.

- Suppose that connection 1 is in state $i$ at time $t_n$. Define $g(i)$ as the state of this connection at time $t_{n+1}$ when it is hurt by the congestion at time $t_n$. $\hat{g}(i)$ denotes its state if connection 2 is hurt by the congestion at time $t_n$.

$$g(i) = \frac{T_1^2 + 2T_2^2}{2(T_1^2 + T_2^2)} \times i \tag{5}$$

$$\hat{g}(i) = \frac{T_1 T_2^2}{2(T_1^2 + T_2^2)} \times \left(\mu - \frac{i}{T_1}\right) + i \tag{6}$$

Matrix $P = (p_{ij})_{i,j\in\mathcal{I}}$ can then be written as,

$$p_{ij} = \begin{cases} p_1 = \frac{i}{\mu T_1} & \text{if } j = g(i) \\ p_2 = 1 - p_1 = 1 - \frac{i}{\mu T_1} & \text{if } j = \hat{g}(i) \\ 0 & \text{otherwise} \end{cases}$$

INRIA

# Definition of a semi-Markov process

- Define the process $A(t)$ as

$$A(t) = W_1(t_n) \qquad \text{for } t_n \leq t < t_{n+1}.$$

The transition time of this process depends on the current and the next state. $A(t)$ forms a semi-Markov process.

- Associated cost: Suppose that $A(t)$ visits state $i$ then jumps to state $j$ on the next congestion event. We define $f_{ij}$ as the integral of $W_1(t)$ between these two transitions. We denote the time between these transitions by $\tau_{ij}$.

$f_i$ denotes the cost function associated to state $i$ := the expected value of $f_{ij}$ over all the possible values of $j$.

$$\begin{aligned} f_i &= \sum_{j\in\mathcal{I}} f_{ij}p_{ij} = f_{ig(i)}p_{ig(i)} + f_{i\hat{g}(i)}p_{i\hat{g}(i)} \\ &= i/(T_1\mu)\int_0^{\tau_{ig(i)}} (t/T_1 + i/2)\,\mathrm{d}t \\ &\quad + (1 - i/(T_1\mu))\int_0^{\tau_{i\hat{g}(i)}} (t/T_1 + i)\,\mathrm{d}t \end{aligned}$$

- Denote by $\tau_i$ the average time $A(t)$ stays in state $i$,

$$\tau_i = \sum_{j\in\mathcal{I}} \tau_{ij}p_{ij} = \tau_{ig(i)}p_{ig(i)} + \tau_{i\hat{g}(i)}p_{i\hat{g}(i)}.$$

- The throughput of connection 1 is equal to the time average of its congestion window divided by $T_1$

$$\overline{X_1} = \lim_{t\to\infty} \frac{1}{t} \int_0^t \frac{W_i(\tau)}{T_1} \mathrm{d}\tau$$

Using the theory of Markov reward processes (or of delayed regenerative processes), this limit is obtained as:

$$\overline{X_1} = \frac{\Sigma_{i\in\mathcal{I}}\, \pi_i f_i}{\Sigma_{i\in\mathcal{I}}\, \pi_i \tau_i}, \qquad P - a.s.$$

- For connection 2, the relationship between $\{W_1(t_n)\}$ and $\{W_2(t_n)\}$ is used to avoid the repetition of all the work.

To every state of the Markov chain associated to connection 1 corresponds a state of the Markov chain associated to connection 2.

Only the cost functions for connection 2 need to be recalculated.

The throughput of connection 2 is again calculated by dividing the time average of $W_2(t)$ by $T_2$.

INRIA

# Numerical Results

- We plot the throughputs of the two connections as a function of the ratio of their RTT, and compare to the model with synchronization.
- The bottleneck bandwidth is 1.5 Mbps.
- TCP packets are of size 576 bytes.
- The RTT of the slow connection is fixed to 0.5 s. The RTT of the fast connection is varied.
- The X-axis represents the ratio of the small RTT and the long one.

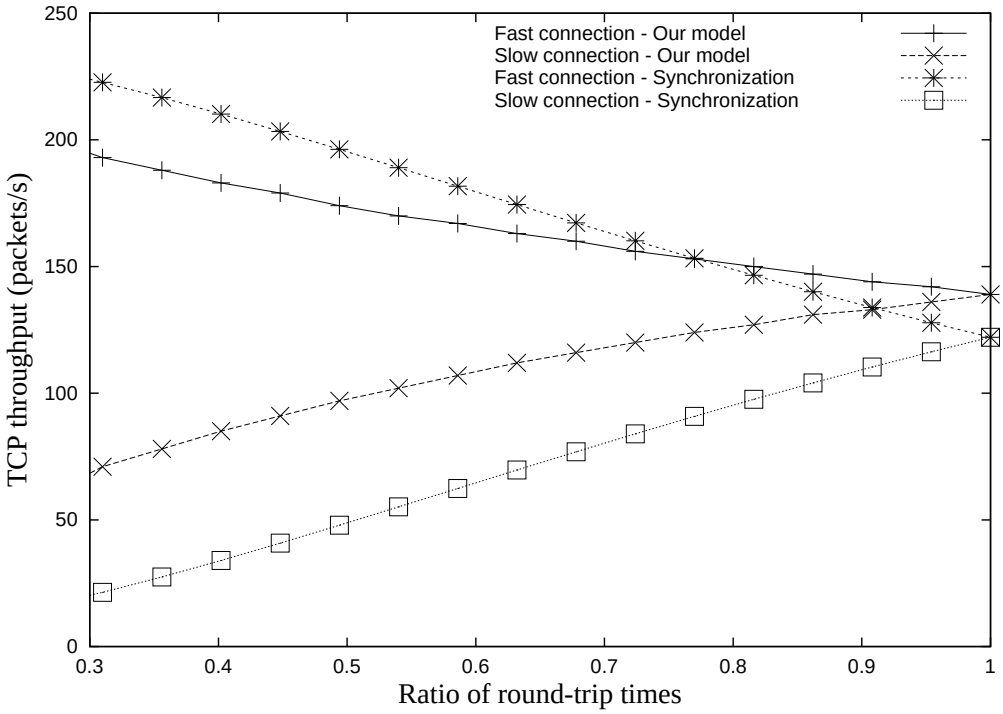


Figure 1: Comparison of throughputs

- The inner curves correspond to our model.

**The throughput achieved by the slow connection is better when there is no synchronization**

Given that the slow connection has a small throughput, there is a small probability that it reduces its window upon congestion. This gives it better performance.

However, when the two connections are synchronized, the slow connection is obliged to reduce its window with the fast one.

INRIA

# Utilization

- The increase in the performance of the slow connection in our case is accompanied by a decrease in the performance of the fast one.
- However, the deterioration in the performance of the fast connection is not as important as the improvement in the performance of the slow one. This means that our model predicts better utilization of the bottleneck bandwidth than the model in [LM97].
- Indeed, when a congestion occurs, the sum of the rates is equal to $\mu$. In our case, one of the two connections reduces its window and then the reduction in the total rate is less than half $\mu$.
- However in the synchronization case, the two connections divide their windows simultaneously and the reduction in the total rate is equal to $\mu/2$.

**Thus, in our case, the utilization is kept at higher levels**

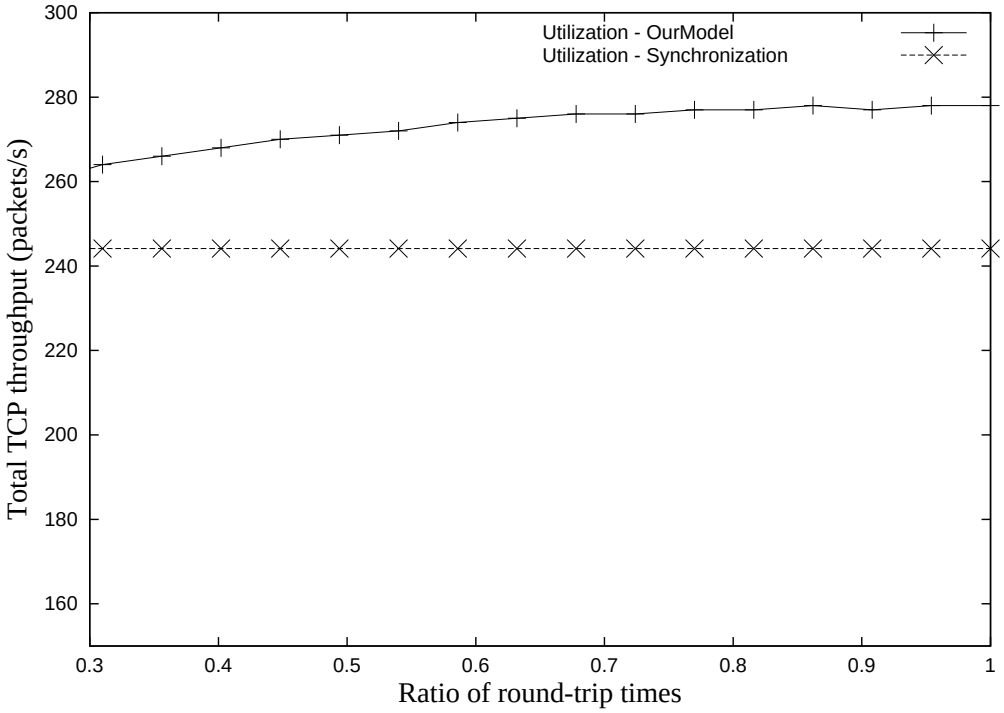


Figure 2: Comparison of utilizations

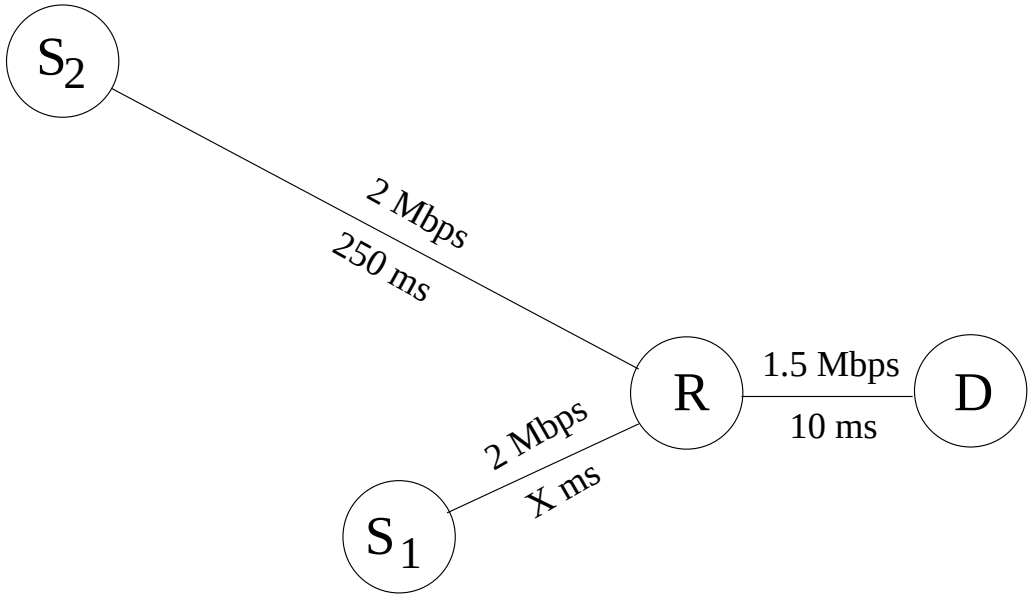


Figure 3: The simulation scenario

## Simulation

- We simulate two long TCP transfers over a bottleneck node using the `ns-2` simulator. The version TCP-SACK is used.
- We vary the propagation delay of the link between $S_1$ and $R$ between 40 ms and 200 ms.
- Simulations are run for 500s each.
- The receiver windows are set large enough so that the window is only limited by network parameters.

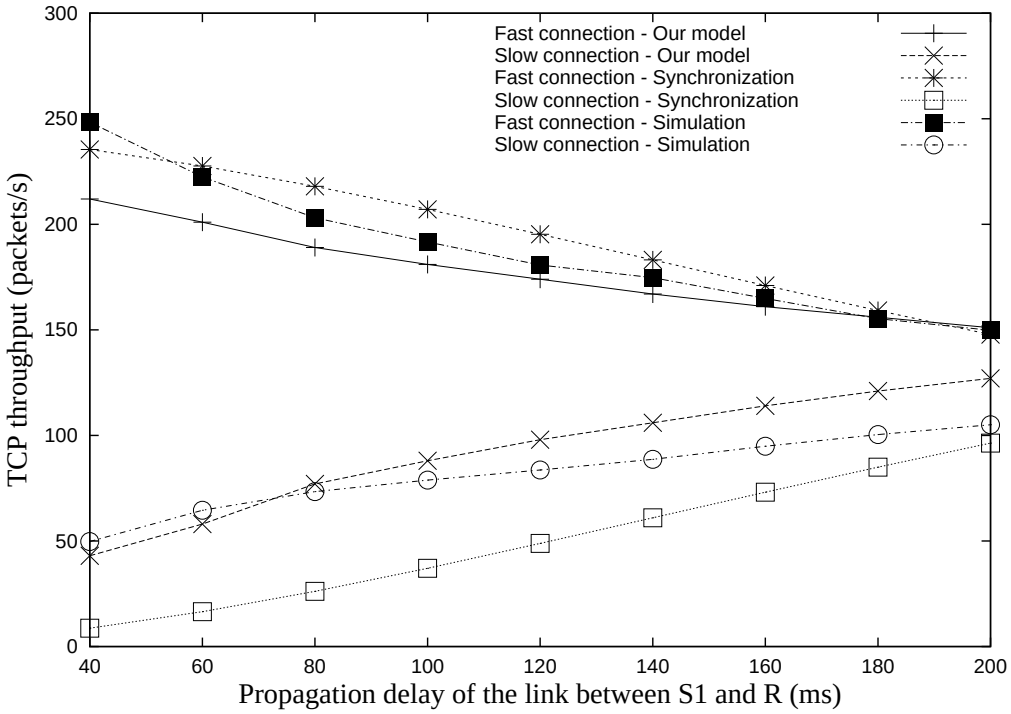


Figure 4: Throughputs in case of RED buffer

## RED Buffer

- $R$ is a RED buffer of a total size of 20 packets, a minimum threshold of 5 packets and a maximum threshold of 10 packets. The maximum drop probability is taken equal to 0.1 and the weight used in the calculation of the average queue size is taken equal to 0.002. The reason for taking small thresholds is to minimize the queueing time.

- The inner curves correspond to our model.
- The outer curves correspond to the synchronization model of Lakshman and Madhow
- The curves in between correspond to simulations.
- The results are closer to those of our model especially for the slow connection. This connection gets more bandwidth than what is predicted in the synchronization case.
- The RED buffer with its probabilistic drop alleviates the problem of synchronization and improves the fairness of TCP.

INRIA

# Conclusions

- We proposed a Markovian fluid model to study the fairness of TCP when connections are not synchronized.
- The absence of synchronization is claimed to be one of the main results of active queue management techniques such as RED.
- We showed that the fairness of TCP improves in a non-synchronized environment.
- We showed that the absence of synchronization improves the utilization of network resources.
- We validated these results with simulations.
- The burst absorption capacity of active buffers improves the accuracy of fluid models for TCP. Drop Tail buffers are biased against bursty traffic and fluid models don't work well especially in case of small buffers.