Utility-based Fair Bandwidth Sharing in Vehicular Networks

Jian Pu and Mounir Hamdi
Computer Science and Engineering, HKUST
Clear Water Bay, Kowloon, Hong Kong
{pujian, hamdi}@cse.ust.hk

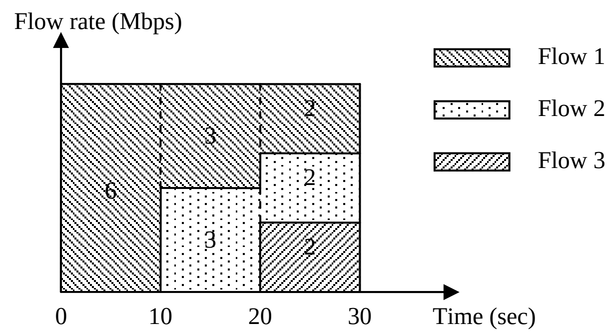
Abstract
The time-constraint data traffic is very common in vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) communications. Time-constraint flows are those who have fixed start and stop times and try to maximize the transferred data volume during the limited connection time. From the view of users, the total data transmitted by each flow should be proportional to each one's holding time if the network resources are allocated fairly. However, we find that the traditional fairness concept solely based on flow rates is not suitable for this scenario. Therefore, we redefine the fairness concept regarding the application utility for time-constraint flows. According to this utility-based fairness definition, we propose a new practical bandwidth sharing scheme using TCP parameter tuning for transferring data with fast-moving wireless nodes such as vehicles. Simulations through ns-2 show that the proposed algorithm can achieve better utility fairness than the standard TCP and it is also friendly to TCP in the long term.

Categories and Subject Descriptors
C.2.3 [**Computer-Communication Networks**]: Network Operations - *network management*.

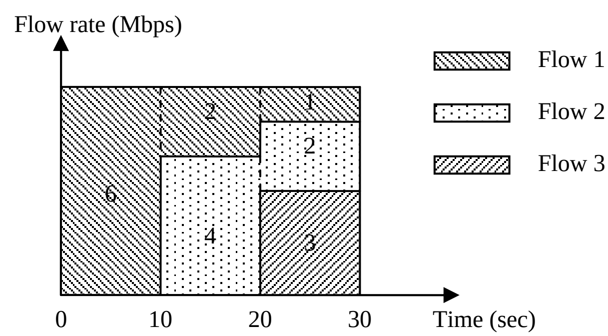
General Terms
Algorithms, Design, Experimentation.

Keywords
Flow control, fairness, utility.

1. INTRODUCTION
The time-constraint data traffic is very common in vehicular networks [1]. For vehicle-to-infrastructure (V2I) communications, a vehicle can download data from the server only when it drives into the communication range of a road-side base station, and it will get disconnected soon later when it drives out of the communication range of the current infrastructure network. For vehicle-to-vehicle (V2V) communications, the connection time between two vehicles is also limited by their relative velocity and the maximum communication range. Although two vehicles drive in the same direction may have the chance to maintain a long-time flow for data exchange, the connection time between two cars driving in opposite directions may be very limited if both of them are moving in high speeds. These kinds of time-constraint flows require more bandwidth to achieve utility-based fairness (explained below) when they are competing with other long-lived flows (*e.g.*, a car parking near the base station downloads movies from the server).



(a) Flow-rate-based fairness



(b) Application-utility-based fairness

Figure 1. Fair bandwidth sharing

Fairness is an important aspect of bandwidth allocation schemes. Basically, if n flows share one bottleneck link, each flow should get $1/n$ of the bottleneck link bandwidth. There are different definitions for bandwidth allocation fairness but most of them are defined solely on the flow rates, *e.g.*, max-min fairness, proportional fairness [6], Jain's fairness index [5]. Max-min fairness gives priority to users with small rates in bandwidth sharing. Max-min fairness is achieved when any one user's rate cannot be increased without decreasing the rate of another user who is already receiving a lower rate. A vector of flow rates x^* is said to be proportional fairness if it is feasible and for any other feasible rate vector x, the following holds:

$$\sum_i \frac{x_i - x_i^*}{x_i^*} \le 0 .$$

Jain's fairness index is defined on a set of flow rates as:

$$\frac{\left(\sum_i x_i\right)^2}{n\sum_i (x_i)^2},$$

where n is the flow number. This index value ranges from $1/n$ (the worst case) to 1 (the best case).

We find that the traditional fairness definitions based on flow rates are not suitable for the time-constraint flows because they are much different from the long-lived flows as implicitly assumed in the previous literature. In practice, a good rate allocation algorithm should reflect the application utilities of the users and tries to achieve application-utility-based fairness. We will demonstrate the difference between the two kinds of fairness in a simple example (Figure 1).

Suppose that there are three time-constraint flows sharing a single bottleneck link of 6 Mbps. Each flow's start and stop time points are fixed: 0-30 sec for Flow 1, 10-30 sec for Flow 2, and 20-30 sec for Flow 3. All the users try to maximize the data transferred during their limited connection time while the network tries to maintain fairness among users. From the view of the users, the total data transferred should be proportional to their connection time. Since the connection time of the three flows are 30:20:10 = 3:2:1, the total data transferred by the three flows are expected to be 3:2:1. However, the traditional flow-rate-based fairness is against this intuition. The network would try to allocate the bottleneck link bandwidth evenly among all living flows as shown in Figure 1(a). Therefore, the total data transferred by the three flows would be 110:50:20 = 5.5:2.5:1. Note that Flow 1 lasts three times longer than Flow 3 but transferred 5.5 times more data than Flow 3. User 3 may complain that this is unfair as it is not gained a fair share of the network resource and it should not be prejudiced due to its short connection time. But if we allocate the bottleneck link bandwidth in the way as shown in Figure 1(b), then the total data transferred by the three flows would be 90:60:30 = 3:2:1, which are proportional to each flow's holding time.

The drawback of the traditional concept of fairness is due to its "forgetfulness". It does not remember any history flow information but only takes the instantaneous flow rates into consideration. This idea may hold for real-time multimedia flows. For example, a user watching an online video can enjoy high-quality motion pictures if the instantaneous flow rate x_i is high regardless of any history information (such as how many data it has sent, how long it has hold the link, *etc*.). Hence, the corresponding utility function for user i could simply be defined in the form of $U_i(x_i)$. However, this kind of utility function is not applicable to time-constraint flows, which try to send as many data as possible in the limited connection time but care less about the instantaneous flow rates. As mentioned before, the basic requirement of fairness in this scenario is that the data transferred by flow i (denoted as S_i) should be proportional to its holding time T_i. In this case, the utility function should also remember S_i and T_i in addition to x_i.

It seems that for time-constraint flows, each flow only needs to remember its average flow rate $\overline{x_i}(t)=\frac{S_i(t)}{T_i(t)}$ to calculate its utility. However, we will show that in order to design an effective utility-based bandwidth sharing algorithm, we still need to remember S_i and T_i for each flow. For example, considering Figure 1(b) at time 20 sec, we have

$$\overline{x_1}(20)=80/20=4$$

and

$$\overline{x_2}(20)=40/10=4.$$

So Flow 1 and Flow 2 have the same average flow rate at this time. But obviously they should have different rate allocation in the following time period according to the utility-based fairness as shown in the figure.

The rest of the paper is organized as follows. Section II describes the system model as a utility-based optimization problem and then proposes a distributed flow control algorithm through TCP tuning. In Section III, We implement the proposed algorithm in ns-2 and perform simulations to evaluate the utility fairness among time-constraint flows as well as TCP friendliness. Finally, we conclude the paper in Section IV.

2. FAIR BANDWIDTH SHARING FOR TIME-CONSTRAINT FLOWS

2.1 System Model

Let I be the set of users and J be the set of links. The capacity of link $j \in J$ is c_j. A is the routing matrix where for any $A_{i,j} \in A$, $A_{i,j}=1$ if user i is using link j and $A_{i,j}=0$ otherwise. At time t, for user i, its holding time is $T_i(t)$ and its transferred data is $S_i(t)$. Suppose we want to assign a new flow rate $x_i(t+\Delta t)$ to user i in the next time period Δt. The utility function of user i is U_i which is a function of S_i, T_i and x_i. This time-variant system tries to solve the following optimization problem:

$$System: t \rightarrow t+\Delta t$$

$$\begin{aligned}
&\text{maximize} \sum_{i\in I} U_i\left(S_i(t+\Delta t), T_i(t+\Delta t), x_i(t+\Delta t)\right)\\
&\text{subject to } \forall i \in I, x_i(t) \geq 0\\
&\qquad \forall j \in J, \sum_{i\in I} A_{i,j} x_i(t) \leq c_j\\
&\qquad \forall i \in I, S_i(t+\Delta t) = S_i(t) + x_i(t+\Delta t)\Delta t\\
&\qquad \forall i \in I, T_i(t+\Delta t) = T_i(t) + \Delta t
\end{aligned}$$

The definition of the utility function U_i is dependent on the specific application of user i. For time-constraint flows, the utility should be proportional to the total data transferred and inversely proportional to its holding time. So the utility function U_i would be:

$$U_i(t)=\frac{S_i(t)}{T_i(t)}.$$

Then if $U_i = U_j$ for any two flows i and j (*i.e.*, their average sending rates are the same), the total data sent by each flow would be proportional to each one's holding time, which is the expected result by users for time-constraint flows.

Therefore, we have

$$\begin{aligned} U_i(t+\Delta t) &= \frac{S_i(t+\Delta t)}{T_i(t+\Delta t)} \\ &= \frac{S_i(t)+x_i(t+\Delta t)\cdot\Delta t}{T_i(t)+\Delta t} \\ &= \frac{\Delta t}{T_i(t)+\Delta t}x_i(t+\Delta t)+\frac{S_i(t)}{T_i(t)+\Delta t}. \end{aligned}$$

Note that all the values of the variables in the above formula can be calculated at time t except x_i. In addition, we know that the feasible flow rate vector $\{x_i\}$ is restricted by the link capacity vector $\{c_i\}$. So the original problem is reduced to a bounded linear optimization problem.

The perfect utility-based fairness is achieved when $U_i = U^*$ for $\forall i \in I$. We call U^* as the desired utility. To simplify our analysis, let's consider a system with one bottleneck link of capacity c. If this bottleneck link is always saturated during the next time period Δt, the desired utility U^* can be solved as:

$$\begin{aligned} U^*(t+\Delta t) &= \frac{\sum_{i\in I} S_i(t+\Delta t)}{\sum_{i\in I} T_i(t+\Delta t)} \\ &= \frac{c\cdot\Delta t+\sum_{i\in I} S_i(t)}{|I|\cdot\Delta t+\sum_{i\in I} T_i(t)}. \end{aligned}$$

In addition, we can solve the desired flow rate x_i^* for each flow $i \in I$ as:

$$\begin{aligned} x_i^*(t+\Delta t) &= \frac{S_i(t+\Delta t)-S_i(t)}{\Delta t} \\ &= \frac{U^*(t+\Delta t)\cdot(T_i(t)+\Delta t)-S_i(t)}{\Delta t} \\ &= U^*(t+\Delta t)\cdot\left(\frac{T_i(t)}{\Delta t}+1\right)-\frac{S_i(t)}{\Delta t}. \end{aligned}$$

Take an example in Figure 1(b), at time t = 20, we have: $c=6$, $\Delta t=10$, $S_1(t)=80$, $T_1(t)=20$, $S_2(t)=40$, $T_2(t)=10$, $S_3(t)=0$, $T_3(t)=0$. So the desired utility at time $t+\Delta t$ can be calculated as:

$$U^*(t+\Delta t)=3.$$

And the desired flow rates for the next time period Δt are:

$$\begin{aligned} x_1(t+\Delta t)&=1, \\ x_2(t+\Delta t)&=2, \\ x_3(t+\Delta t)&=3. \end{aligned}$$

2.2 Distributed Flow Control Algorithm

The above approach to the optimization problem is tractable in mathematics but it requires a central controller with global knowledge. That is, a controller is needed to collect all history flow information, compute the desired utility, and allocate the desired flow rates to all users through some feedback communication channel. However, it is impractical to deploy such centralized control system on the Internet due to the scalability issue. We are more interested in distributed solutions. Especially, we know that TCP [3] is the de facto congestion control protocol in the Internet and there are parameters in its algorithm which can be used to control the steady-stage rate in bandwidth competing. We would like to design a distributed flow control algorithm through TCP parameter tuning to solve the bandwidth sharing problem for time-constraint flows.

A TCP sender uses the congestion window (cwnd) to control its flow rate. The Additive Increase Multiplicative Decrease (AIMD) algorithm embedded in TCP congestion control has two parameters: α and β. If there is no loss, the sender will increase its congestion window by α packets per round-trip time (RTT). And if it experiences a packet loss, the sender will decrease the congestion window by the portion of β. In TCP, the default value of α is 1 and β is 0.5. The flow rate can be approximated as [7]:

$$x=\frac{MSS}{RTT}\sqrt{\frac{\alpha(2-\beta)}{2\beta p}}$$

where *MSS* is the maximum segment size in bytes, *RTT* is the round-trip time, and p is the packet loss probability. Therefore, by increasing α and/or decreasing β for user i, we can increase the flow rate x_i and thus increase the utility of user i. This provides a way to solve the underutilization problem of TCP for flows of short connection time. The basic idea is that we increase the flow rate x_i more aggressively than TCP when S_i is low and then generally slow down the growth rate to the same as TCP when S_i is large.

$$\alpha=\begin{cases} \alpha_{\min}, & S(t)\le S_{\min} \\ \alpha_{\min}, & S(t)\ge S_{\max} \\ \alpha_{\max}+\dfrac{\alpha_{\min}-\alpha_{\max}}{S_{\max}-S_{\min}}(S(t)-S_{\min}), & otherwise \end{cases}$$

$$\beta=\begin{cases} \beta_{\min}, & S(t)\le S_{\min} \\ \beta_{\max}, & S(t)\ge S_{\max} \\ \beta_{\min}+\dfrac{\beta_{\max}-\beta_{\min}}{S_{\max}-S_{\min}}(S(t)-S_{\min}), & otherwise \end{cases}$$

Here $\alpha_{\min}$ and $\beta_{\max}$ are actually the TCP default values of α and β, $\alpha_{\max}$ is the maximum value of α, $\beta_{\min}$ is the minimum value of β, $S_{\min}$ is the lower threshold of flow size, $S_{\max}$ is the upper threshold of flow size. When a flow has transmitted more data than S_{max}, it will be considered as a long-lived flow instead of a time-constraint flow and it would behave just the same as a normal TCP flow. We call this modified version of TCP as MTCP. MTCP gives more bandwidth to short flows compared to the standard TCP in order to achieve the utility fairness among short and long flows.

3. PERFORMANCE EVALUATION

We have implemented the proposed flow control protocol MTCP in the network simulator ns-2 [2]. Table 1 gives a list of the parameter settings in our simulations. The network topology is shown in Figure 2. There is one road where vehicles can drive in two directions. A base station is set up at the roadside and it is connected to other servers through wired links. Users in cars can download files from the servers when they drive into the service area of the base station.

Table 1. Simulation parameters

α_{max}	2.0
α_{min}	1.0
β_{max}	0.5
β_{min}	0.2
S_{min} (bytes)	10000
S_{max} (bytes)	1000000
MSS (bytes)	1000
Wired link bandwidth (Mbps)	5
Wired link delay (ms)	5
Wireless radio range (m)	600

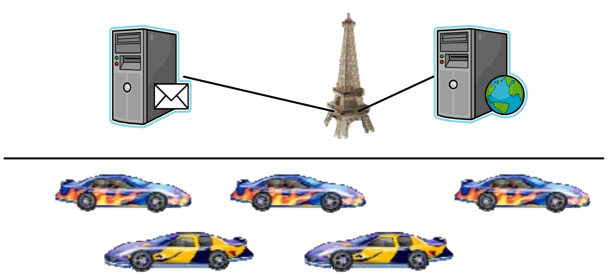
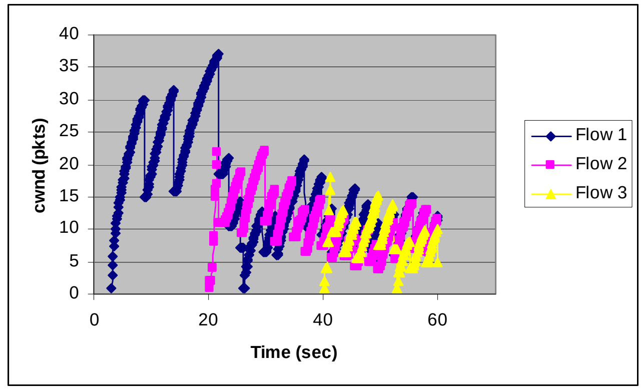


Figure 2. Network topology

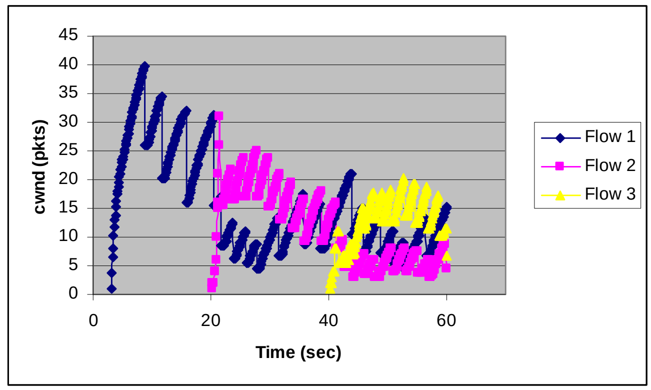
3.1 Time-Constraint Flows

In this scenario, there are three cars drive at different speeds: 10 m/s, 15 m/s, and 30 m/s. Therefore, their connection times are inversely proportional to their speeds: 60 sec, 40 sec, and 20 sec.

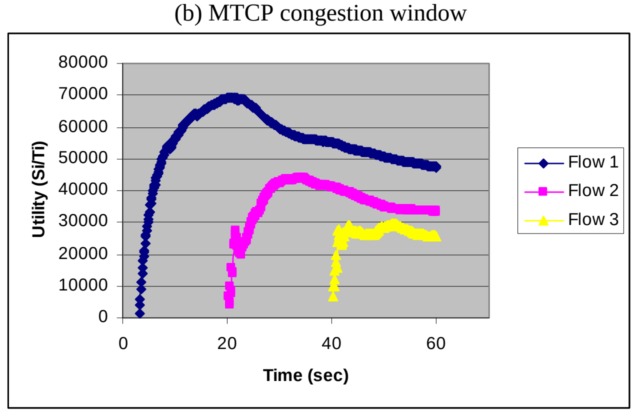
Each user starts downloading data from the server once the connection is up and stops downloading when the connection is down. Assume that the unit utility gain is equal for all the three users, *i.e.*, $\mu_1 = \mu_2 = \mu_3 = 1$. Thus, the utility of a user is the same as its average data rate: $U_i(t) = S_i(t)/T_i(t)$, for $i = 1,2,3$. In Figure 3, (a) and (c) show the results of TCP while (b) and (d) show the results of MTCP.



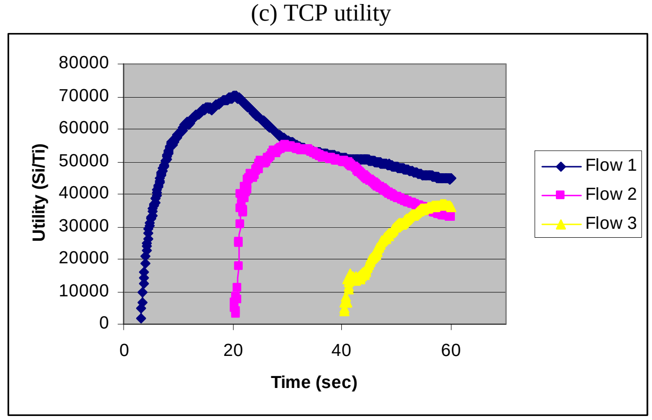
(a) TCP congestion window



(b) MTCP congestion window



(c) TCP utility



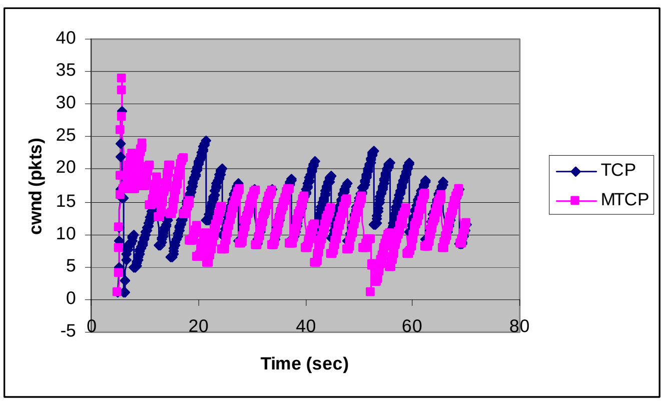
(d) MTCP utility

Figure 3. Simulations on time-constraint flows

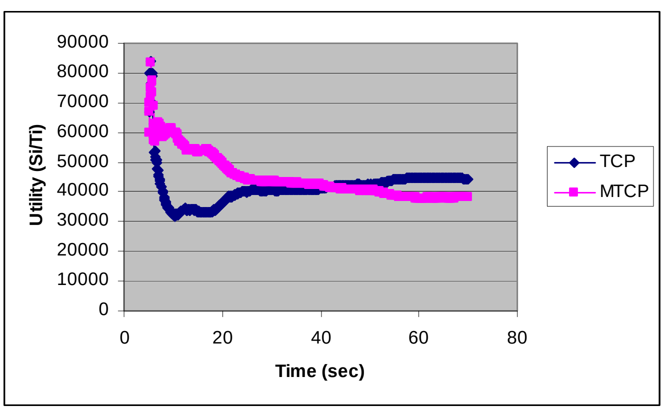
Let's look at the congestion windows first. When a new flow joins the network, TCP tries to maintain the same window size for all flows so that they can share the bottleneck link bandwidth evenly. If all flows are long-lived flows, then the utility of these TCP flows will eventually lead to be equal. But since they are time-constraint flows here, Figure 3(c) show that they have never got a chance to reach utility fairness during their living time. At the end of the simulation, the total bytes transferred by the three TCP flows are 2845000:1337000:514000 = 5.54:2.6:1, which is far from the utility-fair-share allocation 3:2:1. While for MTCP, it tries to allocate more bandwidth to new coming flows by increasing window aggressively and decreasing window conservatively. As shown in Figure 3(d), MTCP flows gradually converge to the utility-fair-share point during the short connection time. At the end of the simulation, the total bytes transferred by the three MTCP flows are 2681000:1318000:720000 = 3.72:1.83:1, which is near to the utility-fair-share allocation 3:2:1.

3.2 TCP Friendliness

TCP friendliness [4] is an important issue when designing a new flow control protocol because TCP has already been widely deployed on the Internet. In this scenario, we simulate one MTCP flow competing with another TCP flow. The simulation time is 70 sec and the results are shown in Figure 4.



(a) Congestion window



(b) Utility

Figure 4. Simulations on TCP friendliness

At the beginning, the MTCP flow grows faster than the TCP flow and gains more bandwidth. But the unfairness between the two flows (measured as U_{MTCP}/U_{TCP}) is upper-bounded by 2 (which is decided by the parameters $\alpha_{\max}$ and $\beta_{\min}$). So there is a way to adjust this short-term unfriendliness to TCP. Moreover, after the MTCP flow transfers data more than $S_{\max}$, it converts to a normal TCP flow and competes for the bandwidth in the same way as the other TCP flows. Thus, a MTCP flow is TCP-friendly in the long term.

4. CONCLUSION

We studied the common fairness concepts in link bandwidth allocation and pointed out that they are not suitable for time-constraint flows. We defined the utility-based fairness for these flows from the view of users. Then we modified TCP congestion control algorithm to reflect this new concept of fairness. We implemented the proposed algorithm in ns-2 and conducted simulations for performance evaluation. The results show that our protocol can achieve better utility fairness than the standard TCP and its TCP unfriendliness is bounded by both time and scale.

Although we study the fairness issue mainly in the context of vehicular networks, we believe that the basic idea should also be applicable to other time-constraint applications in the Internet. In the future, we will try to develop intelligent algorithms that can adapt the threshold parameters dynamically based on the changing network conditions. For example, the server may collect useful statistics on users' behavior and decide which kinds of flows should be treated as time-constraint flows and set the optimal parameters accordingly on the fly.

5. REFERENCES

[1] IEEE Draft Standard for Information Technology - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 7: Wireless Access in Vehicular Environments (WAVE), IEEE P802.11p/D7.0, May 2009.

[2] The ns-2 Network Simulator, http://www.isi.edu/nsnam/ns.

[3] Allman, M., Paxson, V. and Stevens, W. TCP Congestion Control, RFC2581,, 1999.

[4] Feng, J. and Xu, L., TCP-Friendly CBR-Like Rate Control. in *IEEE International Conference on Network Protocols (ICNP)*, (Orlando, FL, 2008), 177-186.

[5] Jain, R. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. John Wiley & Sons, 1991.

[6] Kelly, F.P., Maulloo, A.K. and Tan, D.K.H. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, *49*. 237-252.

[7] Xu, G. and Huang, Y., Transitional behaviors of general AIMD rate control. in *the 11th International Conference on Computer Communications and Networks*, (2002), 132-137.