

Congestion Control in Wireless Networks

Umesh D. Nimbhorkar (00D07006)

Mohit Garg (00D07015)

May 05, 2004

1 Problem Formulation

The most important goal in networking is to achieve reliable (probably delay-bounded) transmission of data over the network infrastructure. In wired networks, traffic congestion on network nodes is the dominant source of packet loss and delay. Thus, protocols like TCP use mechanisms like adaptive window control to tackle this problem of congestion.

In wireless networks, however, apart from the above problem of congestion, the time varying channel becomes another cause of unreliable data transmission and packet loss. The use of TCP in these cases (assuming all packet losses are due to congestion) can lead to very conservative designs if the congestion and fading time scales are different. A joint strategy to combat both congestion and fading must be used to increase throughput. This is the problem that we have worked on in this project.

2 Scope of the Problem

2.1 TCP Congestion Control

Transmission Control Protocol (TCP), used in wired networks uses *Adaptive Window Management* to control packets being offered to the network. This algorithm allows the sender to adapt its sending rate to the current traffic conditions in the path it is using for the particular flow.

The TCP protocol incorporates sender-receiver flow control through acknowledgements being sent, from the destination to the source node, for the packets received. Using these acknowledgement packets, the congestion control mechanism operates as follows: The source TCP uses a *Window* parameter which is the *maximum number of unacknowledged packets* that it can have at any given time during the connection. This restricts the maximum number of packets that the sender can offer to the network without having received the acknowledgement for any of them. Adaptive Window Control involves adaptively changing the size of the transmission window based on the acknowledgement packets that the source has received.

For example, in a typical TCP implementation, the window size W will be initially set at 1, and will be incremented by 1 for every acknowledgement received. This is based on the assumption that if the acknowledgement was received, then there is spare capacity in the network and hence it increases the number of packets it can offer to the network without being acknowledged.

Clearly, this cannot go on infinitely and hence there will be a time when a particular packet will be dropped and the sender will not receive an acknowledgement for it. Then, the window size W will be reduced and after some timeout delays, the packet will be retransmitted. Many variations of the above algorithm are available and used in practice. The differences are in the ways W is increased and packet losses are recovered from. *e.g.* Tahoe, Reno, NewReno etc.

2.2 Congestion Control in Wireless Networks

Adaptive Window based control as used by TCP was designed for use in wired networks. It becomes imperative to ask whether the same mechanism will also work in the case of wireless networks. In order to be able to answer this question, we need to understand the differences between wired and wireless networks.

The most dominant (and probably the only) difference between wired and wireless networks is the time-varying random fluctuations (also referred to as *fading*) in the wireless channel. This puts a severe doubt on the reliability ($\text{BER} \sim 10^{-6}$) of the bit-carrier infrastructure which is assumed to be extremely reliable ($\text{BER} \sim 10^{-9}$) in wired networks. Hence, it is not clear whether mechanisms being used in wired networks will perform comparably in wireless networks as well. Particularly, we wish to study the performance of TCP's adaptive window control approach w.r.t. wireless networks.

In TCP, the window size W is reduced whenever packets are lost – assuming that the loss is due to traffic congestion along the path of the flow under consideration. This assumption is not valid in wireless networks since the channel may have been in a poor state thereby causing the packet to be lost at the physical layer (OSI Model) itself. The statistics of this type of loss may be very different from a packet dropping at a router queue due to traffic congestion. Thus, we need to understand:

1. Whether there is a *need to distinguish* between traffic related losses and channel related losses in wireless networks.
2. If we agree to distinguish between the two, we need to *find efficient ways* of achieving this objective.
3. *Formulate strategies* to take different corrective actions for each of the above causes.

3 Related Work

TCP when proposed, about 15 years ago, was a heuristics based protocol. There was no analytical framework to characterize its performance. Theoretical research in this topic has started only recently and it is only now that we have some analytical models for understanding the behaviour of TCP and TCP-like protocols.

Kelly, in [8], has shown that congestion control and mechanisms can be viewed as distributed

algorithms solving the following network utility maximization problem,

$$\text{maximize } \sum_s U_s(x_s) \tag{1}$$

$$\text{subject to } \sum_{s \in L(l)} x_s \leq c_l, \forall l, \mathbf{x} \geq 0 \tag{2}$$

where source rates $\mathbf{x} = \{x_s\}$ are the optimization variables, c_l are the constant link capacities, $L(s)$ denotes the set of links traversed by the connection originating from source s , and the utility U_s for each source can be any increasing function with decreasing slope (concave).

Low, Paganini, and Doyle, in [3], applied the above formulation to different variants of TCP. They showed that the different variants of TCP, which differ in the feedback parameters and rate control mechanism, solve the same optimization problem as in (1)-(2) with different utility functions.

Chiang and Man, in [1], modified the above formulation for application in wireless networks. They used adaptive power control (hence link capacities in (2) are no longer constants) as a possible way to combat wireless channel fading. They formulated this as a joint power and rate optimization problem and proposed a distributed algorithm for the same. We are specifically looking at this formulation.

4 System Model

4.1 Control System formulation of Congestion Control Mechanisms

A *wired* network is modeled as set of N sources and L links with capacities c_l , $l \in L$. Source i uses a set $L_i \subseteq L$ of links. Thus we can write the route incidence matrix $[[R_{ij}]]$, as follows,

$$R_{li} = \begin{cases} 1, & \text{if } l \in L_i \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Each source i has an associated transmission rate $x_i(t)$, and the aggregate flow $y_l(t)$ at each link can be written as,

$$y_l(t) = \sum_i R_{li} x_i(t - \tau_{li}^f) \tag{4}$$

where τ_{li}^f are the forward transmission delays. A *congestion measure* $p_l(t)$ is associated with each link (this is referred to as *cost*). The sources are assumed to have access to aggregate cost of all links in their route, denoted by $q_i(t)$,

$$q_i(t) = \sum_l R_{li} p_l(t - \tau_{li}^b) \tag{5}$$

where τ_{li}^b are the backward delays of the feedback paths.

Thus, to specify the congestion control system, the following need to be specified.

- The way in which the sources adjust their rates ($x_i(t)$) based on their aggregate prices. (*e.g.* TCP window control mechanism)

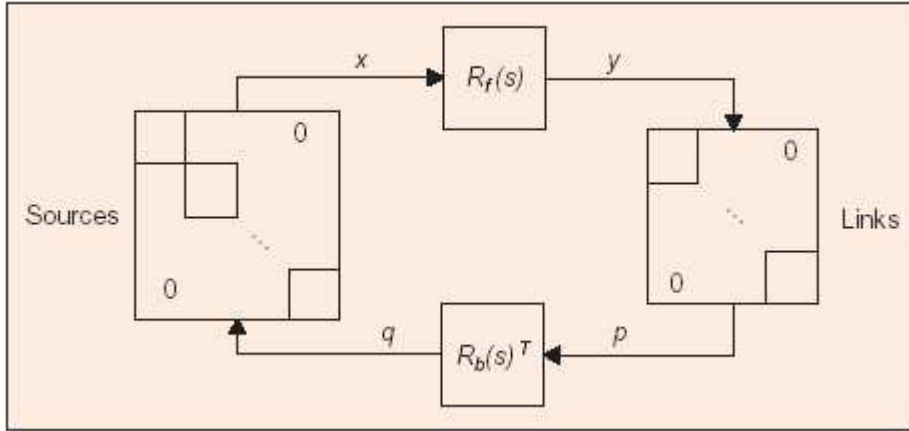


Figure 1: General Congestion Control Structure (Source [3])

- The way in which links adjust their prices ($p_l(t)$) based on their aggregate rates. (*e.g.* Active Queue Management (AQM) algorithm)

At source level, the following dynamic model can be postulated,

$$\dot{z}_i = F_i(z_i, q_i) \quad (6)$$

$$x_i = G_i(z_i, q_i) \quad (7)$$

where z_i is local state variable. Similarly, at link level, we can write the dynamic law

$$\dot{v}_l = H_l(y_l, v_l) \quad (8)$$

$$\dot{p}_l = G_l(y_l, v_l) \quad (9)$$

One important restriction in the above control laws is that they must be decentralized *i.e.* sources and links have access to only their local information. The overall congestion control system structure now takes the form in fig. 1.

Low, Paganini, and Doyle, in [3] showed that the equilibrium of the above dynamic system has an optimization interpretation *i.e.* for a suitable utility function $U_i(x_i)$, the equilibrium rates will solve

$$\max_{x_i} U_i(x_i) - x_i q_i^* \quad (10)$$

where q_i^* are equilibrium aggregate prices of the route corresponding to source i .

This framework requires minimal assumptions about the congestion control protocol. Given a congestion control protocol (*e.g.* TCP), we can derive the utility function associated with the protocol. The role of prices here, is to coordinate the actions of the individual sources to achieve individual optimality along with global optimality, *i.e.* to ensure that the solutions of (10) also

TCP		Model
Reno	Source control	$\dot{x}_i = \frac{1-q_i(t)}{\tau_i^2} - \frac{1}{2}q_i(t)x_i^2(t)$
RED	Link control	$\dot{b}_i = \begin{cases} (y_i(t) - c_i) & \text{if } b_i(t) > 0 \\ [y_i(t) - c_i]^+ & \text{if } b_i(t) = 0 \end{cases}$ $\dot{r}_i = -\alpha_i c_i (r_i(t) - b_i(t))$ $p_i = m_i(r_i)$
	Utility	$U_i(x_i) = \frac{\sqrt{2}}{\tau_i} \tan^{-1} \left(\frac{\tau_i x_i}{\sqrt{2}} \right)$
Vegas	Source control	$\dot{x}_i = \begin{cases} \frac{1}{\tau_i^2}, & \text{if } x_i(t) < \bar{x}_i(t) \\ -\frac{1}{\tau_i^2}, & \text{if } x_i(t) > \bar{x}_i(t) \end{cases}$
FIFO	Link control	$\dot{p}_i = \begin{cases} \frac{1}{c_i}(y_i(t) - c_i), & \text{if } p_i(t) > 0 \\ \frac{1}{c_i}[y_i(t) - c_i]^+, & \text{if } p_i(t) = 0 \end{cases}$
	Utility	$U_i(x_i) = \alpha_i d_i \log x_i$

Figure 2: Models of TCP (Source [3])

solve the problem,

$$\max_{\mathbf{x} \geq \mathbf{0}} \sum_i U_i(x_i) \quad (11)$$

$$\text{subject to } R\mathbf{x} \leq \mathbf{c} \quad (12)$$

Note that these equations are same as (1)-(2)

In other words, maximize aggregate utility across all sources, subject to link capacity constraints. The link prices p_i now appear as Lagrange multipliers for the problem (11)-(12). The different variants of TCP can now be interpreted as the special cases of the optimization problem. All these differ in the corresponding utility functions, the source and link control dynamic laws, the feedback prices. The following table (2) shows the dynamic laws and the utility functions for some TCP variants.

4.2 Modification for Wireless Environment

In a wireless network, link capacities c_l are randomly varying functions of time. Thus, in order to distinguish between packet loss due to fading and that due to congestion, we should ideally track the channel capacities and take control decisions accordingly.

One possible approach can be to exploit the dependence of channel capacity on power. In general we can write the capacities as $c_l(P, t)$. The intuition is to adaptively vary the transmit power P such that the variations in capacity due to fading are minimized. Chiang and Man, in

[1], follow a similar approach.

Thus we have the following modified version of the optimization equations (11)-(12),

$$\max_{\mathbf{x} \geq \mathbf{0}} \sum_i U_i(x_i) \quad (13)$$

$$\text{subject to } R\mathbf{x} \leq \mathbf{c}(\mathbf{P}) \quad (14)$$

$$P_l \leq P_{l,max} \quad \forall l \quad (15)$$

$$\mathbf{P}, \mathbf{x} \geq \mathbf{0} \quad (16)$$

5 Implementation Framework

In a wireless network, link capacities randomly fluctuate with time due to channel conditions (fading). The algorithm in [1] adaptively varies the transmission power at each node to counteract the effect of channel fading. This power control algorithm works over and above the TCP implementation.

The paper shows that the joint optimization of the utility functions as in (13)-(16) is equivalent to solving the following two independent optimization problems,

$$\text{maximize}_{\mathbf{x} \geq \mathbf{0}} \sum_s U_s(x_s) - \sum_s \sum_{l \in L(s)} p_l x_s \quad (17)$$

$$\text{maximize}_{\mathbf{P}_{max} \geq \mathbf{P} \geq \mathbf{0}} \sum_l p_l c_l(\mathbf{P}) \quad (18)$$

where, p_l , the link prices, act as the Lagrange multipliers in the optimization equation. The first optimization is already implicitly solved by the congestion control mechanism and hence, we need to maximize the second equation while allocating the Lagrange multipliers as link prices.

A Shannon capacity model is assumed for the channels under consideration (which may not be a good assumption to make for the wireless channel). Thus,

$$c_l = \frac{1}{T} \log(1 + \mathcal{K} \text{SIR}_l) \quad (19)$$

where, T is the symbol duration, \mathcal{K} is a constant depending on the modulation scheme used and the required BER, and SIR_l is the signal-to-interference ratio for link l defined as $\text{SIR}_l = \frac{P_l G_{ll}}{\sum_{k \neq l} P_k G_{lk} + n_l}$ for a given set of transmit powers P_l , path losses G_{lk} (from the transmitter on link l to the receiver on link k), and noises n_l (for the receiver on link l).

The optimization is carried out using the *gradient descent* algorithm with constant step size k . The following equation is presented as the distributed power update equation,

$$P_l(t+1) - P_l(t) + k \frac{p_l(t)}{P_l(t)} - k \sum_{j \neq l} G_{lj} m_j(t) \quad (20)$$

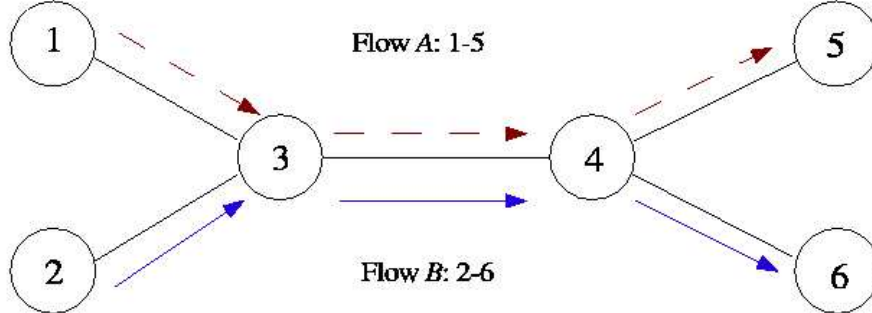


Figure 3: Simulation Topology

where m_j are the messages passed from link j to the transmitter on link l ,

$$m_j(t) = \frac{p_l(t) \text{SIR}_j(t)}{P_j(t) G_{jj}(t)} \quad (21)$$

The algorithm assumes complete knowledge of the system state at all nodes and each link performs the updates independent of the other links. All transmit powers are upper-limited by $P_{l,max}$ during the iterations. The values of p_j , SIR_j and P_j can be directly measured by node j locally, and the path loss G_{jj} can be estimated through periodic training sequences.

6 Simulation Results

We simulated the above model for TCP Reno. TCP Reno is the *fast-retransmit/fast-recovery* variant of the TCP congestion control algorithm. This protocol aims at increasing the throughput of TCP by inferring loss via duplicate acknowledgements and uses better window control mechanism to recover from loss. The prices (Lagrange multipliers in (17)) can now be interpreted as link level loss probabilities [3]. Precisely, with $c_l(t)$ as link capacities and $x_l(t)$ as aggregate flow over that link, we can write prices as [9],

$$p_l(t) = \left[1 - \frac{c_l(t)}{x_l(t)} \right]^+ \quad (22)$$

We simulated the scenario using Matlab [10]. The topology used is shown in fig. 3. Each node is a TCP Reno agent, whose transmission power is updated in each cycle depending on the loss probabilities calculated. Each link is basically a wireless channel between two nodes whose capacity depends on the the SIR at the receiver due to all transmissions in surroundings. The channel gains were randomly chosen with log-normal shadowing ($\sigma = 8$ dB). A link also has a finite transmission buffer associated with it. Now the packet loss will be either due to

Start Tx Power	With Power Control	Without Power Control
1	298	220
2	295	228
3	296	221
4	308	233
5	297	162

Table 1: Number of Packets successfully transmitted Vs. Start Transmit Power

buffer overflow (congestion) or due to channel fading (BER). Two competing TCP flows over the wireless link were initiated as shown. In order to understand the effect of power control the channel gains, once initialised, remained constant during the course of the simulation. The queueing delays and propagation delays were also simulated for each flow.

We observed the end to end throughput for the two flows with and without the above power control mechanism. We found that the power control mechanism consistently gave higher throughput than the constant power TCP, (Table 1) thus indicating that power control indeed helps in improving TCP performance. It can also be seen that, since the power control algorithm is adaptive, the throughput is independent of the start transmission power. The transmit powers for each node converge after sometime, thus indicating the convergence of the power control algorithm. As the channel gains are chosen randomly at start, it may happen that one route gets a bad channel. In this case, without power control, the system is not fair as can be seen from the throughputs of the two. The power control algorithm, on the other hand, adjusts the transmit powers adaptively and still achieves fairness.

7 Conclusion & Future Work

This adaptive power control algorithm over TCP certainly improves the performance in terms of throughput as shown in for TCP Vegas in [1] and verified by us for TCP Reno. Thus, using ‘pure’ TCP over wireless networks is sub-optimal. The algorithm converges rapidly and can be a candidate for jointly combating fading and congestion in wireless networks. The ability to dynamically control the transmit powers provides another control over the link prices and hence influences the feedback for the TCP control system. This in turn improves the performance.

We feel that the following issues need to be studied w.r.t the above protocol

- It is not clear whether the proposed algorithm actually distinguishes between fading and congestion related losses. In other words, the improved performance maybe due to the change in link capacities (transmission rates) according to the amount of packets in the link buffer so as to minimise the probability of loss. This needs to be understood thoroughly using more detailed simulations.
- Another possible drawback of this algorithm is the total transmission energy for each node. In our simulations, the TCP source nodes typically ended up transmitting at much higher power levels. Thus, the performance of this algorithm in energy constrained networks also

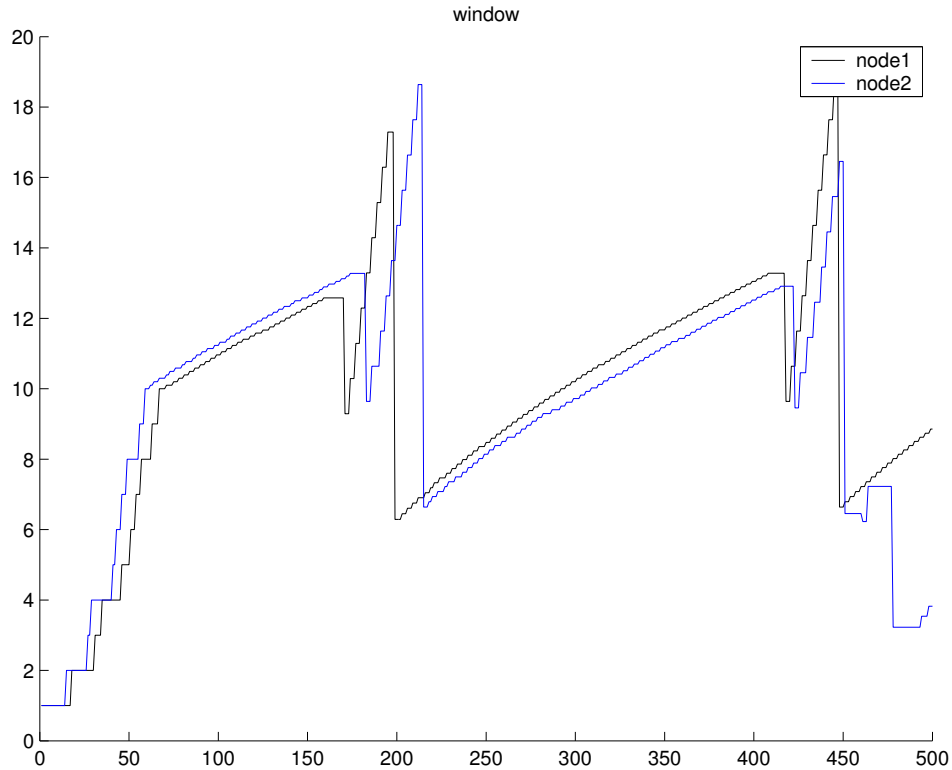


Figure 4: TCP Window Vs Time (with power control)

needs to be investigated. Modifications to this algorithm to incorporate fairer power levels may also be looked into.

- The above algorithm assumes full information about the system state at each node. This is clearly impractical for real life networks. This shortfall needs to be studied thoroughly.

Acknowledgements

We sincerely thank Prof. Abhay Karandikar for giving us an opportunity to work in this area. We would also express our gratitude to Ashutosh Gore for the many discussions that we had with him.

References

- [1] M. Chiang, and R. Man, *Jointly Optimal Congestion Control and Power Control in Wireless Multihop Networks*, GlobeComm 2003.
- [2] K. B. Kim, A. Tang, and S. H. Low, *Design on AQM in Supporting TCP Based on the Well-Known AIMD Model*, GlobeComm 2003.

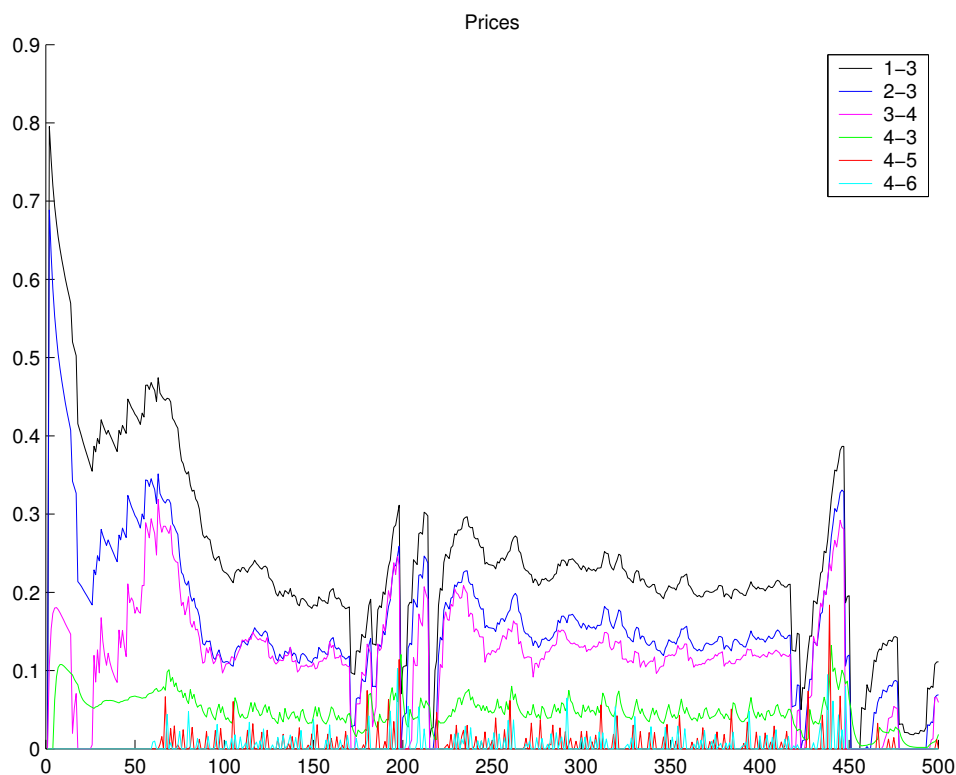


Figure 5: Link prices Vs Time (with power control)

- [3] S. H. Low, F. Paganini, and J. C. Doyle *Internet Congestion Control*, IEEE Control Systems Magazine, Feb. 2002.
- [4] S. Ryu, *et al*, *Advances In Internet Congestion Control*, IEEE Communications Surveys & Tutorials, Third Quarter 2003, Vol. 5, No. 1.
- [5] H. Balakrishna, V. N. Padmanabhan, S. Srinivasan, and R. H. Katz, *A Comparison of Mechanisms for Improving TCP Performance over Wireless Links*, ACM SIGCOMM, August 1996.
- [6] A. Kumar, D. Manjunath, and J. Kuri, *Communication Networking: An Analytical Approach*, Manuscript.
- [7] O. Westin, Homepage of an M.Sc Thesis, <http://www.sics.se/~owe/exjobb/>
- [8] F. P. Kelly, *Charging and rate control for elastic traffic*, Eur. Trans. Telecommunications, Vol. 8, pp. 33-37, 1997. Available: <http://www.statslab.cam.ac.uk/~frank/elastic.html>
- [9] *Equilibrium and Dynamics of TCP*, Available: www.cnse.caltech.edu/Industry/Conferences/2001/7thAIC
- [10] MATLAB: <http://www.mathworks.com>

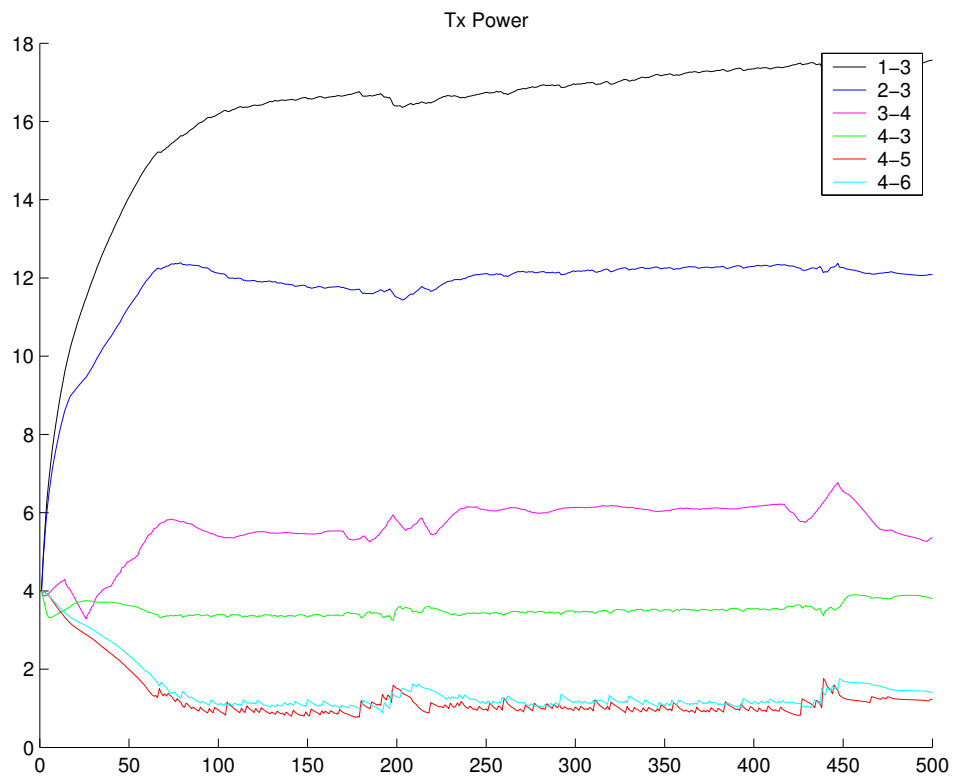


Figure 6: Channel Transmission Power Vs Time (with power control)



Figure 7: Channel data rates vs Time (with power control)

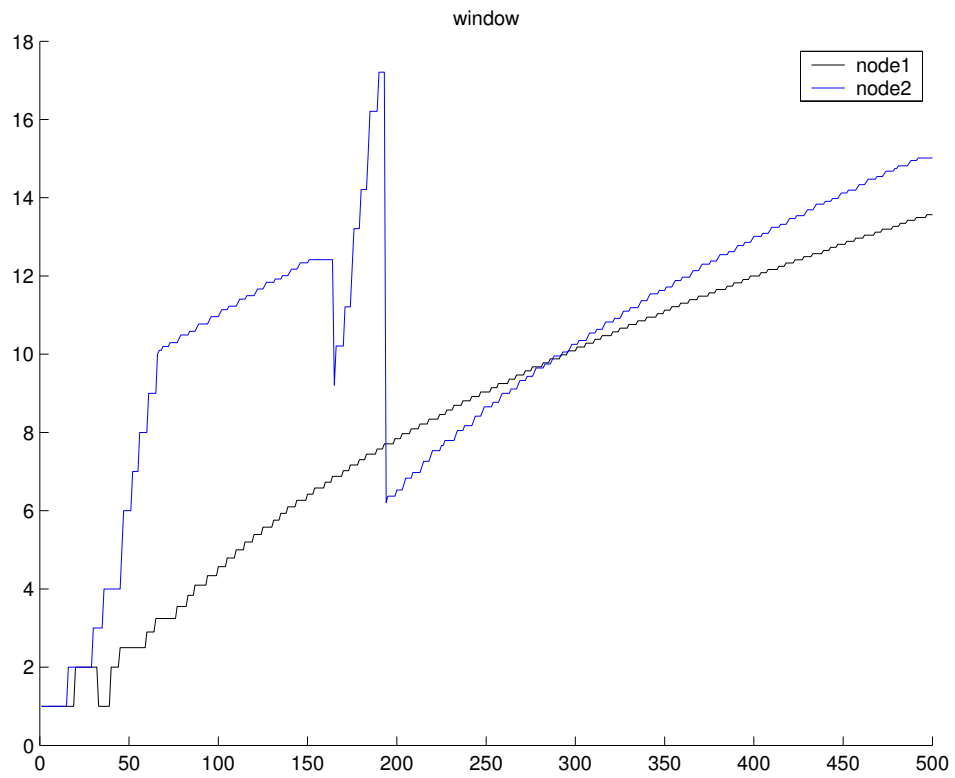


Figure 8: TCP Window Vs Time (without power control)

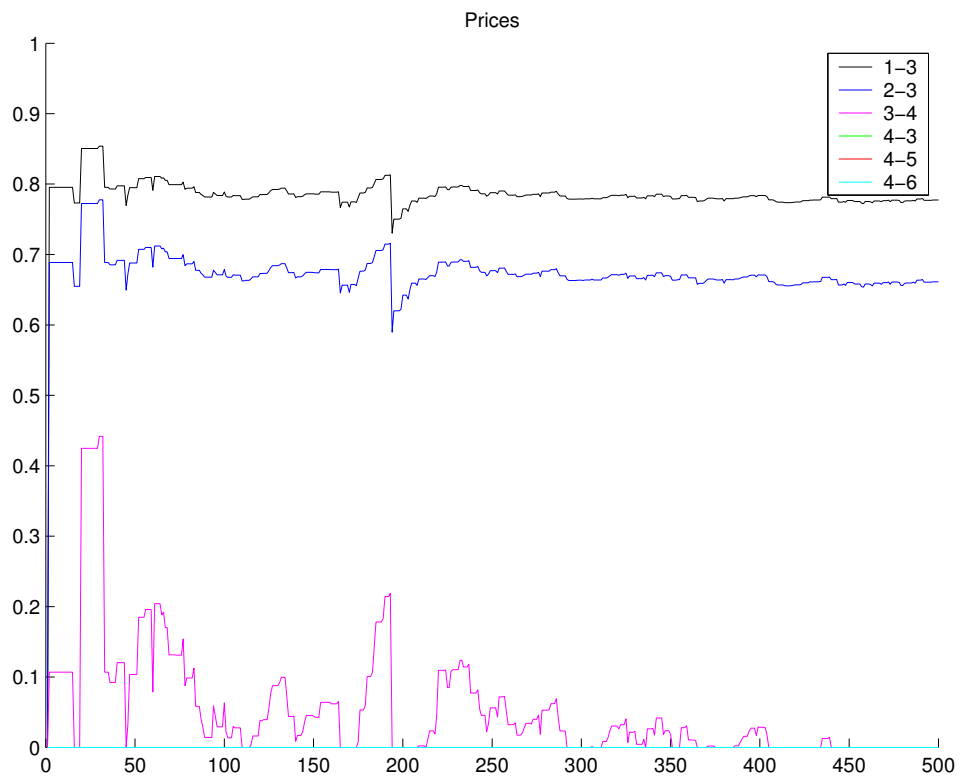


Figure 9: Link prices Vs Time (without power control)

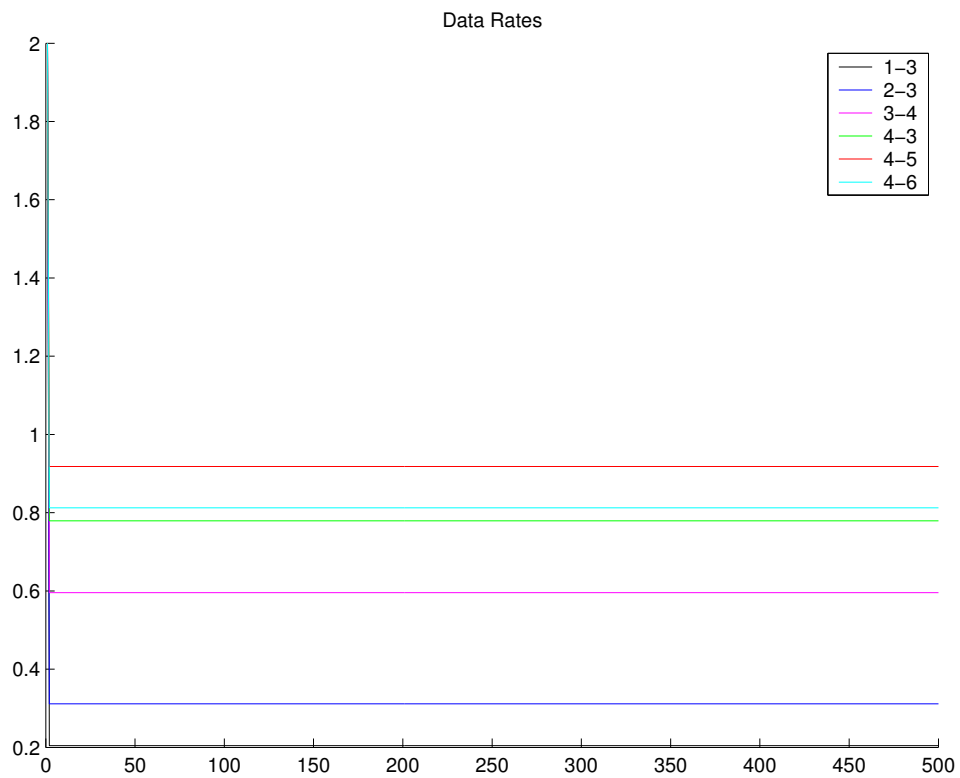


Figure 10: Channel data rates vs Time (without power control)