On Cross Layer Congestion Control for CDMA-based Ad-hoc Networks

Hemant Kumar Rath and Abhay Karandikar
Information Networks Lab, Department of Electrical Engineering
Indian Institute of Technology Bombay, Mumbai 400 076, India
Email: {hemantr, karandi}@ee.iitb.ac.in

***Abstract*— We investigate a cross layer congestion control technique for TCP Reno-2 (NewReno) in CDMA-based wireless ad-hoc networks in which both TCP layer and PHY layer jointly control congestion. The PHY layer varies transmission power based on the channel condition, interference power received and congestion in the network and seeks to transmit at a minimum power level, whereas the TCP layer controls congestion using Reno-2 flow control. We show the convergence of our algorithm theoretically and through simulations. We demonstrate that our cross layer congestion control technique provides performance improvement in terms of throughput, window size variations, transmission power and converges to a stable value even with addition and deletion of flows to the network.**

I. Introduction

In a wired network, the links are assumed to be reliable and of fixed capacities. Congestion in such kind of networks is the major cause of packet loss and/or delay. Various techniques are employed in the Internet to counter the congestion problem, which are either proactive or reactive. These techniques are implemented in the Transmission Control Protocol (TCP) layer of Internet. In TCP, congestion is said to have occurred when the sender receives three duplicate acknowledgments (*dupacks*) or when a timeout (packet loss) occurs. The TCP congestion control techniques are based on an Additive Increase and Multiplicative Decrease (AIMD) technique, in which increase and decrease of congestion window (*cwnd*) is based on acknowledgements received and packet drops/*dupacks* respectively. TCP Tahoe [1], TCP Reno [2] and TCP Reno-2 or NewReno[3], [4] are some of the commonly deployed variants of congestion control techniques.

The congestion control techniques of TCP are demonstrated to be effective in the wired networks of the Internet. However this is not true in wireless networks, where packet loss and/or delay can result not only due to congestion, but also due to time varying nature of the wireless channel. The link capacity in wireless networks is not fixed but depends upon the Signal to Interference and Noise Ratio (SINR) at the receiver of the link. Congestion in a wireless link can be controlled either by controlling the transmission rate of the link or by increasing the capacity of that link. Capacity of a link can be increased either by increasing transmission power on that link or by decreasing transmission power on other interfering links. Since wireless nodes are battery powered, transmission power on a link should be minimum, such that the lifetime of the network can be increased. Also, transmissions at a higher power in a link leads to higher interference on some other link. Hence, instead of applying the congestion control techniques of wired networks directly to wireless networks a joint congestion and power control technique is desirable. An optimization based joint congestion and power control technique is proposed in [5] to address the congestion control problem of wireless networks for TCP Vegas. We have extended the technique of [5] to TCP Reno-2 (NewReno) in [6] in which the TCP layer performs the window based flow control and PHY layer varies the transmission power of wireless nodes depending on the congestion cost of the links. In this paper, we extend [6] by incorporating "network cost" which is the combination of congestion cost and energy cost of a link. The energy cost of a link is a function of the transmission power in that link whereas the congestion cost of a link is the packet dropping probability in that link. We seek to minimize the transmission power and maximize the transmission rate by considering the network cost of the links.

Hemant Kumar Rath is a Philips India Fellow and his research is supported by Philips India and that of Abhay Karandikar is supported by Department of Science and Technology (Project Code 05DS025).

A. Related Work

Modeling congestion control as an optimization based flow control problem has been addressed in [7], [8], [9], [10]. In these works, the authors have modeled a constrained optimization based flow control in which the sources adjust their transmission rates in response to the congestion in the network. The solution to the constrained optimization problem provides the solution to congestion control in the network.

Various authors have used optimization based flow control techniques in which instead of controlling the rate of transmission only, they control the transmission power of the links to control congestion in wireless networks. For example, in [5], the authors have employed power control along with congestion control in a wireless network and analyzed their scheme for TCP Vegas. In [11], the author has modeled the power control and utility maximization of a wireless network as a sum product problem. This approach is used to design a new Joint Optimal Congestion control and Power control (JOCP) algorithm for a wireless network. In [6], we have extended JOCP for TCP Reno-2 and have proposed a distributed congestion control and power control algorithm considering the packet loss due to buffer overflow in a node.

The rest of the paper is organized as follows. In Section II, we discuss the system model and network cost for a multi-hop wireless ad-hoc network. In Section III, we discuss the individual and social optimization techniques and solve a joint congestion and power control technique for TCP Reno-2. We discuss our simulations, results and convergence analysis of the cross layer congestion control algorithm in Section IV. Finally, we provide the concluding remarks in Section V.

II. System Model

We assume a multi-hop CDMA-based wireless ad-hoc network, where the nodes are placed uniformly over a 2-dimensional area. The nodes are stationary and have identical system properties. Let $SINR_l$ be the SINR of a link[1] $l \in L$ expressed as $SINR_l = \frac{P_l G_{ll}}{\sum_{k \neq l} P_k G_{lk} + n_l}$, where P_l is the transmission power on the link l, G_{ll} is the path gain from the transmitter of link l to the receiver of the link l and G_{lk} is the path gain from the transmitter on link k to the receiver on link l. n_l is the thermal noise on the link l. Using Shannon's capacity theorem, we determine the maximum capacity of link l as: $c_l = \frac{1}{T}\log(1 + MI.SINR_l)$ packets/sec, where T is the symbol period and MI is the modulation index which depends on the modulation scheme used.

We consider a set I of source-sink pairs which share a network of L unidirectional links. The capacity of any individual link l is c_l, $l \in L$. Each source-sink pair $i \in I$ has a utility function $U_i(x_i)$ associated with its transmission rate x_i. Any node in our model can send its own traffic and also relay (forward) traffic of other nodes. All traffic are elastic in nature. The utility function $U_i(x_i)$ is assumed to be concave, increasing and double differentiable. The route matrix R consists of all possible links and $R_{li} = 1$, if the path of source-sink pair i traverses link l and $R_{li} = 0$, otherwise. We assume that the time scale of change of routing is higher than the time scale of our analysis, which is a multiple of Round Trip Time (RTT). The aggregate flow $y_l(t)$ (packets/sec) at each link l can be written as: $y_l(t) = \sum_i R_{li} x_i(t)$. Each link l is associated with two different costs, namely congestion cost and energy cost as explained below.

A. Congestion Cost

It is the cost of congestion in a link, e.g., packet dropping probability of a link in TCP Reno and TCP Reno-2 and queuing delay in TCP Vegas. We consider TCP Reno-2 in our analysis, in which the packet loss probability due to congestion at link l is modelled as buffer overflow of an $M/M/1/B$ queuing model. We express the congestion cost $\lambda_l(t)$ of link l (derived in [6]) for TCP Reno-2 as:

$$\lambda_l(t) = \begin{cases} \frac{\max\left(0, (y_l(t) - c_l(t))\right)}{y_l(t)} & \text{if } y_l > 0, \\ 0 & \text{if } y_l = 0. \end{cases} \tag{1}$$

B. Energy Cost

The energy cost is the cost of transmission power spent by a wireless node to transmit its own traffic or to rely (forward) others traffic. Energy cost is important when the nodes are battery powered and maximum node life time is desired. The energy cost $b_l(t)$ is a function of transmission power $p_l(t)$ (i.e., energy consumption) of the node in that link. We express energy cost of a link l as:

$$b_l(t) = \theta + k_1 f_1(k_2 p_l(t)), \tag{2}$$

where θ is the minimum cost to keep the transmitter on (equivalent to maintain minimum SINR) and k_1 and k_2 are constants which can be chosen for a pricing model by the network operator. The value of k_1 and k_2 also indicate whether energy cost or the congestion cost is dominant. The function f_1 is a piecewise monotonically increasing function.

If each link l is associated with $\lambda_l(t)$ and $b_l(t)$ as congestion cost and energy cost respectively, then the end-to-end aggregate cost $q_i(t)$ for source-sink pair i is expressed as:

$$\begin{aligned} q_i(t) &= \sum_l R_{li}(\lambda_l(t) + b_l(t)) \\ &= \sum_l R_{li}\mu_l(t), \end{aligned} \tag{3}$$

where μ_l is the "network cost" of link l; sum of congestion cost and energy cost of the link l. End-to-end aggregate cost is feedback to the source node which computes the rate of transmission. In Fig. 1, we illustrate a simple 2-link, one source-sink pair scenario with the direction of data and cost.

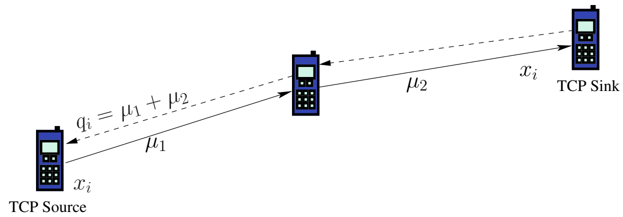

Fig. 1. Source-Sink Pair in an Ad-hoc Network

III. Optimization based Congestion Control

We seek to maximize rate of transmission and minimize transmission power in order to deal with the congestion control and power control problem. The system consisting of source-sink pairs and wireless links attempts to reach the equilibrium point (x^*, q^*). Each source-sink pair i chooses its data rate $x_i(t)$ such that its utility is maximized for the end-to-end cost $q_i(t)$ and the aggregate flow in a link l is less than or equal to the capacity of the link, i.e., $y_l(t) \leq c_l(t)$. We express these in the following individual optimization problem:

$$\max_{x_i} \Big[U_i(x_i) - q_i x_i \Big]; \quad \text{s.t.} \quad \sum_i R_{li} x_i \leq c_l. \tag{4}$$

A. Social Optimization

Since in (4), each user attempts to maximize its individual profit by transmitting at the optimum x_i^*, social optimality or system optimality as in [12] can be achieved for an optimum transmission rate $X = \{x_i\}$. We express the social optimality equation as follows:

$$\begin{aligned} &\max_{X \geq 0} \sum_i U_i(x_i); \quad \text{s.t.} \quad RX \leq C(P); \\ &\text{where,} \quad X = \{x_i\}, \text{and} \quad C(P) = \{c_l(P)\}. \end{aligned} \tag{5}$$

Since the link capacity in a wireless network is a function of transmission power, we use $c_l(P)$ instead of c_l in the above equation. We solve (5) using Karush-Kuhn-Tucker (KKT) [13] optimality conditions by solving the complementary slackness conditions at equilibrium. For this, we associate a Lagrangian Multiplier μ_l for the constraint to (5). Then we determine the stationary points ($X^* = \{x_i^*\}$, $\mu^* = \{\mu_l^*\}$) using:

[1] SINR is measured at the receiver of link l.

$$\phi_{system}(X,P,\mu) = \sum_i U_i(x_i) - \sum_l \mu_l \Bigg(\sum_i R_{li}x_i - c_l(P)\Bigg) \quad (6)$$

Maximization of ϕ_{system} is decomposed[2] as in [5]:

$$\begin{aligned} \max \quad I(X,\mu) &= \sum_i U_i(x_i) - \sum_l \mu_l \sum_i R_{li}x_i, \\ \max \quad I(P,\mu) &= \sum_l \mu_l c_l(P), \\ \text{s.t.,} \quad X \geq 0; &\quad 0 \leq P_l \leq P_l^{Max} \end{aligned} \quad (7)$$

The above decomposition is possible as the common variable μ_l is linked between the two sub-objective functions $I(X,\mu)$ and $I(P,\mu)$. Since the utility function of TCP Reno-2 $\left(U_i(x_i) = \frac{1}{RTT_i}\log\left[\frac{x_iRTT_i}{2x_iRTT_i+3}\right]\right)$ [6] is concave and twice differentiable for $x_i, RTT_i \geq 0$, $I(X,\mu)$ will converge to a global maximum. We investigate the concave nature of $I(P,\mu)$ in the following section.

B. Nature of $I(P,\mu)$

We assume that the symbol period T and the modulation index MI as unity, and for high $SINR$ we re-write the Shannon capacity $c_l = \log(SINR_l)$. Using this we express $I(P,\mu) = \sum_l (\lambda_l + b_l)\log(SINR_l(P))$ for our investigation. Instead of investigating the concave nature of $I(P,\mu)$, we transform the power vector P to a logarithmic power vector $\tilde{P}$ and then investigate the concave nature of the transformed $I(\tilde{P},\mu)$. Let $\tilde{P}_l = \log P_l, \forall l$. We write $I(\tilde{P},\mu)$ as follows:

$$\begin{aligned} I(\tilde{P},\mu) &= \sum_l \mu_l \log\left[\frac{G_{ll}e^{\tilde{P}_l}}{\sum_{k\neq l} G_{lk}e^{\tilde{P}_k} + n_l}\right] \\ &= \underbrace{\sum_l \lambda_l \left[\log(G_{ll}e^{\tilde{p_l}}) - \log\Bigg(\sum_{k\neq l} e^{\tilde{P}_l + \log G_{lk}} + n_l\Bigg)\right]}_{I_1(\tilde{P},\mu)} \\ &\quad + \underbrace{\sum_l b_l \left[\log(G_{ll}e^{\tilde{p_l}}) - \log\Bigg(\sum_{k\neq l} e^{\tilde{P}_l + \log G_{lk}} + n_l\Bigg)\right]}_{I_2(\tilde{P},\mu)} \\ &= I_1(\tilde{P},\mu) + I_2(\tilde{P},\mu) \end{aligned} \quad (8)$$

From (8) it is evident that $I_1(\tilde{P},\mu)$ is concave in $\tilde{P}$, as $\log(G_{ll}e^{\tilde{P}_l})$ is linear in $\tilde{P}$ and $\log\Bigg(\sum_k e^{\tilde{P}_l + \log G_{lk}} + n_l\Bigg)$ is convex in $\tilde{P}$.

$$\begin{aligned} I_2(\tilde{P},\mu) \quad = \quad & \sum_l \theta\left[\log(G_{ll}e^{\tilde{p_l}}) - \log\Bigg(\sum_{k\neq l} e^{\tilde{P}_l + \log G_{lk}} + n_l\Bigg)\right] \\ + \quad & \sum_l k_1 f_1(k_2 P_l)\Bigg[\log(G_{ll}e^{\tilde{p_l}}) \\ - \quad & \log\Bigg(\sum_{k\neq l} e^{\tilde{P}_l + \log G_{lk}} + n_l\Bigg)\Bigg] \end{aligned} \quad (9)$$

[2]Distributed solution is possible as along as there is an interaction between the two decomposed equations through some information passing (message passing in our case). This is known as sum product algorithm [11].

Similarly, from (9) it is evident that $I_2(\tilde{P},\mu)$ is concave as $f_1(k_2P_l)$ is linear and monotonic in P_l. Hence, $I(\tilde{P},\mu)$ is concave in $\tilde{P}$ and in P. Since $I(X,\mu)$ and $I(P,\mu)$ are concave and double differentiable, (6) will converge to a global maxima resulting in optimum $X^* = \{x_i^*\}$, $\mu^* = \{\mu_l^*\}$. The solution methodologies to find the global maxima are explained below.

Each source-sink pair i solves the first maximization equation involving $I(X,\mu)$ as it knows its own utility function $U_i(x_i)$ and the end-to-end cost (which is fed back by the system) by using TCP Reno-2 congestion control in every RTT. We solve the second maximization equation involving $I(P,\mu)$ by choosing appropriate transmission powers of the wireless nodes as discussed below. Differentiating $I(P,\mu)$ with respect to P_l, we evaluate the l^{th} component of the gradient $\nabla I(P,\mu)$. We use the Steepest Descent method as in [6] to solve the maximization problem with a small step size δ to obtain the transmission power as:

$$P_l(t+1) = P_l(t) + \delta\Bigg(\nabla I(P,\mu)\Bigg) \quad (10)$$

Each node computes its required transmission power P_l and network cost μ_l of each outgoing link associated with it and passes on these information to its neighbors, such that end-to-end cost of a path can be computed by each source-sink pair i. Each source-sink pair i computes its transmission rate using TCP Reno-2 and transmission power using (10), which is equivalent to solving a joint congestion control and power control for an ad-hoc network. We suggest an algorithm to implement this joint congestion and power control and provide the pseudo code in Algorithm 1. The parameter Δ in Algorithm 1 is a small tunable system parameter.

IV. Experimental Evaluation

We have considered a CDMA-based ad-hoc network with 6 wireless nodes and two pairs of TCP transmitters and receivers (1-5) and (2-6) as shown in Fig. 2. All nodes in our simulation are capable of transmitting and receiving "network cost". Nodes 1, 2, 5, 6 are TCP Reno-2 agents. Depending upon the network costs, we update the transmission power of the participating nodes. We set the TCP retransmission timeout to be $4 \times RTT$. We update RTT by using the exponential averaging technique as: $RTT = \alpha RTT_{estimated} + (1-\alpha)RTT_{measured}$. We assume that $\alpha = 0.85$ for our simulations. We assume that the time required for transmission in each of the segments 1-3, 2-3, 3-4, 4-5 and 4-6 are same. The forward and reverse channel characteristics are same. The channel gains are assumed to be log-normally distributed with variance $\sigma = 8\ dB$. The path loss factor γ is assumed to be 4. We use Matlab [14] for our simulations. We have implemented TCP Reno-2 in Matlab. We set $w_{initial} = 3$, $P_l^{Min} = 3\ units$ and $P_l^{Max} = 15\ units$ in our simulations. The frequency of updating of $SINR_l$ is a configurable parameter (usually this is a multiple of RTTs). We calculate the data rate x_i by using the relation $x_i(t) = \frac{w_i(t)}{RTT_i}$, whereas $w_i(t)$ and RTT_i are updated using TCP Reno-2 congestion control principle.

A. Results

We simulate TCP Reno-2 congestion control mechanism with and without power control techniques. We have performed 10 independent runs and averaged out the results. In

Algorithm 1 :Cross Layer Congestion Control for CDMA-based Ad-hoc Network

$x_i(0) \leftarrow 1 \; \forall i$
$w_i(0) \leftarrow 3 \; \forall i$
$P_l(0) \leftarrow P_l \; \forall l$
$c_l(0) \leftarrow c_l^{Max} \; \forall l$
$b_l(0) \leftarrow \theta \; \forall l$
$\lambda_l(0) \leftarrow 0 \; \forall l$
$\mu_l(0) \leftarrow \theta \; \forall l$
$t \leftarrow 1$
while TRUE **do**
  Update G_{ll} and G_{lk} periodically
  $SINR_l(t) \leftarrow \frac{P_l(t-1)G_{ll}}{\sum_{k\neq l} P_k(t-1)G_{lk}+n_l} \; \forall l$
  $P_l(t) \leftarrow P_l(t-1) + \delta(\nabla I(P,\mu)) \; \forall l$
  if $|P_l(t) - P_l(t-1)| \leq \Delta$ **then**
    $P_l(t) \leftarrow P_l(t-1) \; \forall l$
  else
    $P_l(t) \leftarrow \min(P_l(t), P_l^{Max}) \; \forall l$
  end if
  Update $w_i(t), RTT_i$ from TCP Module $\forall i$
  $x_i(t) \leftarrow \frac{w_i(t)}{RTT_i} \; \forall i$
  $y_l(t) \leftarrow \sum_i R_{li}x_i \; \forall l$
  $b_l(t) \leftarrow \theta + k_1 f_1(k_2 P_l(t)) \; \forall l$
  $c_l(t) \leftarrow \frac{1}{T}\log(1 + MI \times SINR_l(t) \; \forall l$
  $\lambda_l(t) \leftarrow \frac{\max(0, y_l(t)-c_l(t))}{y_l(t)} \; \forall l$
  $\mu_l(t) \leftarrow \lambda_l(t) + b_l(t) \; \forall l$
  $q_i(t) \leftarrow \sum_l R_{li}\mu_l \; \forall i$
  $t \leftarrow t+1$
end while

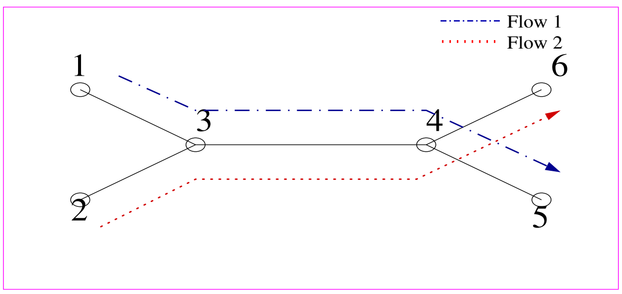

Fig. 2. System Model/Topology

fixed power scheme, the nodes transmit at a maximum power ($P_l^{Mmax} = 15 \; units$), whereas in our cross layer scheme, nodes transmit in an average of $P_l^{Avg} = 8.296 \; units$. We observe that the power consumption of nodes in our cross layer scheme is lower as compared to that of fixed power scheme. Fig. 3 shows the $cwnd$ variation of joint power and congestion control mechanism involving congestion cost only. Fig. 4 shows the $cwnd$ variation of joint power and congestion control mechanism involving network cost and Fig. 5 shows $cwnd$ variation without any power control. From these figures, we observe that the fluctuations in $cwnd$ with power control mechanism (in both congestion cost and network cost) is lower than the $cwnd$ fluctuations without power control mechanism. Also, the average window size of joint power and congestion control scheme is higher than that of congestion control without power control. Hence, power control provides stabilized and better throughput.

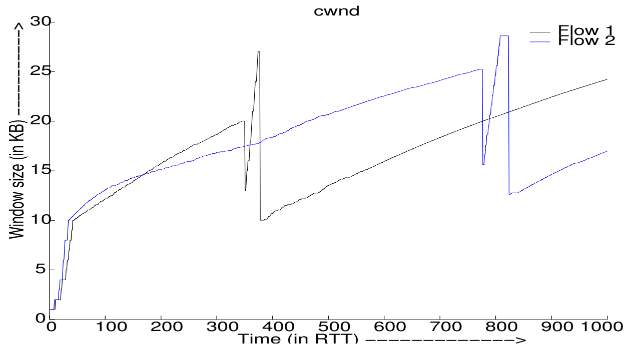

Fig. 3. Variation of $cwnd$ Size with Power Control (Congestion Cost only)

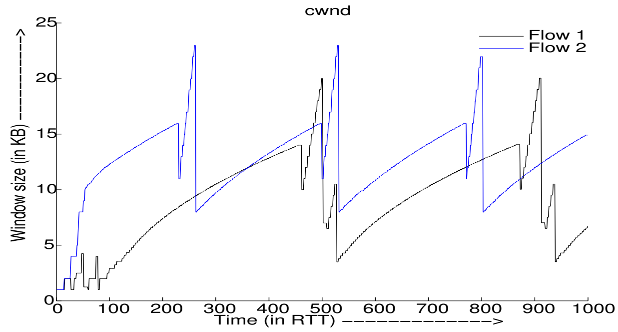

Fig. 4. Variation of $cwnd$ Size with Power Control (with Network Cost)

B. Convergence Analysis

Since $\phi_{system}(X, P, \mu)$ in (6) is separable and $I(X, \mu)$ as well as $I(P, \mu)$ are concave and twice differentiable, a global maximum is guaranteed. But it does not explain about the rate of convergence. The rate of convergence depends on the value of step size δ used in (10). To investigate the rate of convergence, we perform simulations with three different values of δ (0.1, 0.2 and 0.5) for the topology shown in Fig. 2 and observe the number of iterations to converge to an optimum transmission power level. For $\delta = 0.1$, it takes about 150 iterations to converge. For higher values of δ though it takes fewer number of iterations (100 and 80 iterations for $\delta = 0.2$ and 0.5 respectively) to converge to some rate, but does not converge to a stable transmission power level.

The convergence is guaranteed as long as neither a new user enters nor an old user leaves the network. For any addition and/or deletion of nodes/flows, Algorithm 1 will take more iterations to converge. To investigate the convergence of addition/deletion of flows we have used a different topology as shown in Fig. 6 and Fig. 7. We have considered two cases, involving 4 flows and 5 links. In Case-I, we have 4 flows as shown, and run the simulations to find out the equilibrium rates of each flow, number of iterations it takes to converge, transmission power in each link and the total aggregate traffic in each link. We observe that it takes around 150-180 iterations to converge to the equilibrium values (shown in Fig. 8). Link wise transmission powers and aggregate rates are shown in Table I and flow wise transmissions are shown in Table II. We observe that in Case-I, L-3 and L-4 are most congested links as they accommodate 3 flows each. Hence as expected, the transmission powers in those links are higher as compared to other links. Similarly, we observe that L-1 is the least congested link and hence the transmission power is the least in

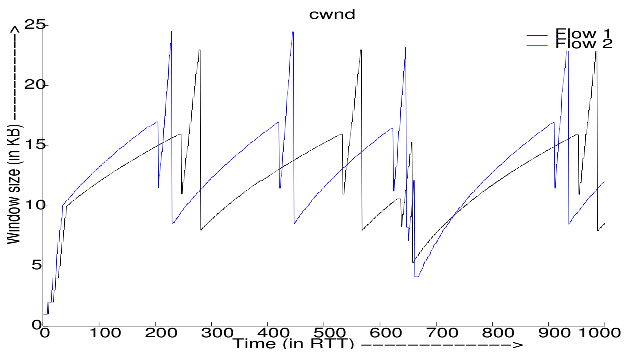

Fig. 5. Variation of *cwnd* Size without Power Control

L-1. After reaching convergence we delete Flow-4 involving L-5 and L-6 and create a new flow involving L-1 and L-2 as shown in Case-II. We observe that it takes another 180-200 iterations (Fig. 8) to converge, and hence we claim that our algorithm converges to addition and/or deletion of flows in a realistic time frame. In Case-II, L-2 and L-3 are most congested as they accommodate 3 flows each and hence the transmission power in those links are high.

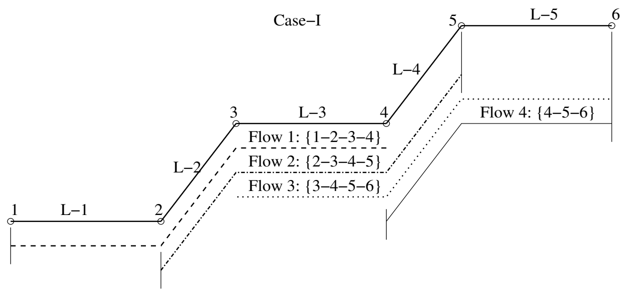

Fig. 6. Topology for Convergence Analysis

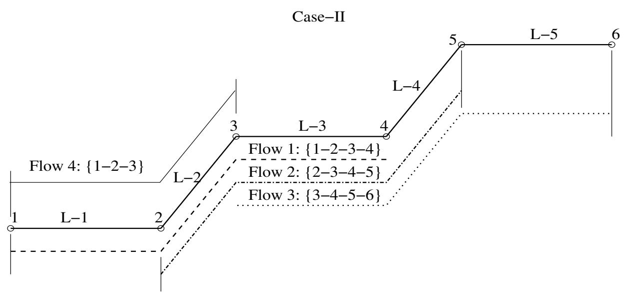

Fig. 7. Topology for Convergence Analysis with Flow Alteration

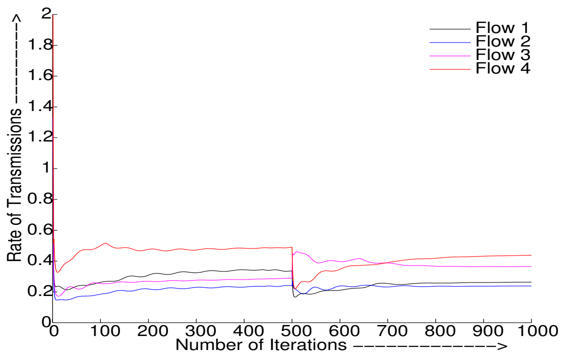

Fig. 8. Convergence after Addition/Deletion of Flows

TABLE I

POWER TRANSMISSION AND LINK USAGE

(a) Case - I

Links	P_l	y_l
L-1	5.2589	0.3147
L-2	8.479	0.5397
L-3	9.9616	0.8231
L-4	10.2413	0.9823
L-5	5.6831	0.7891

(b) Case - II

Links	P_l	y_l
L-1	6.9911	0.7128
L-2	13.6162	0.9489
L-3	12.6060	0.8493
L-4	10.0886	0.5816
L-5	7.6860	0.3453

TABLE II

FLOW RATE

(a) Case - I

Flows	x_i
Flow 1	0.3147
Flow 2	0.2250
Flow 3	0.2835
Flow 4	0.5056

(b) Case - II

Flows	x_i
Flow 1	0.2677
Flow 2	0.2361
Flow 3	0.3455
Flow 4	0.4452

V. CONCLUSIONS

The joint congestion and power control scheme based on "network cost" is a cross layer approach involving PHY and TCP layer. The proposed algorithm converges very fast for small values of step sizes and for addition and/or deletion of flows into the network. As expected, the cross layer congestion control technique provides stabilized throughput and low transmission power for reasonably good channel conditions. However, if the channel conditions are very bad, then there would be more losses due to bad channel resulting in a significant increase in λ_l, which in turn results in an increase in transmission power and b_l. In that case, the convergence of the cross layer congestion control scheme needs to be investigated and is considered as a future work of this paper.

REFERENCES

[1] V. Jacobson, "Congestion Avoidance and Control," in *ACM SIGCOMM*, pp. 314–329, August 1988.

[2] R. Braden, V. Jacobson, and L. Zhang, "TCP Extensions for Highspeed Paths," vol. RFC-1185, October 1990.

[3] S. Floyd and T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm," April 1999. RFC-2582.

[4] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP Selective Acknowledgment Options," April 1996. RFC-2018.

[5] M. Chiang and R. Man, "Jointly Optimal Congestion Control and Power Control in Wireless Multi-hop Networks," in *Proc. of GLOBECOM*, December 2003.

[6] H. K. Rath, A. Sahoo, and A. Karandikar, "Cross Layer Congestion Control Algorithm in Wireless Networks for TCP Reno-2," in *Proc. of NCC, IIT-Delhi*, January 2006.

[7] F. P. Kelly, "Charging and Rate Control for Elastic Traffic," *European Transactions on Telecommunications*, vol. 8, pp. 33–37, January 1997. (Corrected version of the original paper is available at http://www.statslab.cam.ac.uk/∼frank/elastic.ps).

[8] R. J. Gibbens and F. P. Kelly, "Resource Pricing and Evolution of Congestion Control," *Automatica*, vol. 35, December 1999.

[9] S. H. Low and D. E. Lapsley, "Optimization Flow Control-I: Basic Algorithm and Convergence," *IEEE/ACM Transactions on Networking*, vol. 7, pp. 861–874, November 1999.

[10] S. Ryu, C. Rump, and C. Qiao, "Advances in Internet Congestion Control," *IEEE Communications Surveys and Tutorials*, vol. 3, pp. 28–39, 2003.

[11] M. Chiang, "Distributed Network Control through Sum Product Algorithm on Graphs," in *Proc. of GLOBECOM*, November 2002.

[12] C. Courcoubetis and R. Weber, *Pricing Communication Networks: Economics, Technology and Modeling*. John Wiley and Sons, 2003.

[13] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 2nd ed., 1999.

[14] "Matlab version-6." http://www.mathworks.com.