Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

# Internet Congestion Control Subject to Node And Link Constraints ⋆

**Yongxiang Xia, David J. Hill**

*Department of Information Engineering*
*Research School of Information Sciences and Engineering*
*Australian National University*
*Canberra ACT 0200, Australia*
*(e-mail: {Yongxiang.Xia,David.Hill}@anu.edu.au)*

**Abstract:** This paper studies the mathematical modelling of Internet congestion control. Differently to previous models, which consider either the link capacity or the node processing capability as the constraints, here we take both of them into account, i.e., the aggregate flow rate on a link cannot exceed the link capacity and the aggregate flow rate at a node is limited by the node processing capability. A decentralized primal-dual algorithm is proposed to solve the congestion control problem and its convergence is proven. Using this algorithm we show the bottleneck of the network performance when these two constraints are unbalanced.

## 1. INTRODUCTION

With the development of the Internet, mathematical modelling of congestion control have received considerable attention [Low et at. (2002); Srikant (2003); Paganini et al. (2005)]. The basic idea of congestion control is simple. The sources control their transmission rates based on the network congestion level. If the flow rate is too large, then the source decreases its transmission rate to avoid congestion. Similarly, if the rate is too small, then the source increases the rate to achieve more benefits. In mathematical models, each flow is assigned a utility function and the system adjusts the flow rates to maximize the aggregate system utility subject to some resource constraints. In this way, congestion control can be interpreted into an optimization problem.

Several decentralized end-to-end network congestion control algorithms have been proposed in the past decade [Kelly et al. (1998); Liu et al. (2003); Low et al. (1999)]. Most of these algorithms consider the link capacity constraint, so that the aggregate flow rate on a link cannot access the link capacity. However, the link capacity is not the only constraint subjected to the network traffic. Actually, from a viewpoint of Graph Theory [Diestel (2005)], the Internet structurally includes two sets of subjects, nodes and links. The link capacity only describes the traffic constraint on the links. There should be a similar constraint to the nodes. That is, the processing capability of a node also limits the maximum aggregate transmission rates through it. Here the processing capability of a node is defined as the maximum number of bits that this node can process within unit time. In [Doyle et al. (2005)], the authors used the node capability rather than the link capacity as the constraint.

Since both the link capacity and the node processing capability are inevitable constraints to the Internet traffic, we consider these two constraints together in this paper. First we will propose a decentralized congestion control algorithm based on the two constraints. Then we will prove the convergence of the algorithm and show the convergence rate. After that, the effects of these two constraints will be investigated using simulations.

## 2. CONGESTION CONTROL ALGORITHM

### *2.1 Optimization problem*

Consider a communication network with $N$ nodes, $L$ links. $S$ flows of packets are transmitted from their sources to destinations within the network. Set $A_{li} = 1$ if flow $i$ goes through link $l$ and $A_{li} = 0$ otherwise. Then the matrix $A = (A_{li}, 1 \le l \le L, 1 \le i \le S)$ contains all the link routing information. If the rate of flow $i$ is $x_i$ and the capacity of link $l$ is $C_l$, then we have the following inequality

$$Ax \le C, \tag{1}$$

where $x = (x_i, 1 \le i \le S)$ and $C = (C_l, 1 \le l \le L)$ are the flow rate vector and capacity vector, respectively. Similarly, we can define the node routing matrix $B$, where $B_{ni} = 1$ if flow $i$ goes through node $n$ and $B_{ni} = 0$ otherwise. Then we have the inequality of node processing capability.

$$Bx \le D, \tag{2}$$

where $D$ is the node capability vector with the element $D_n$ to be the processing capability of node $n$. The above two inequalities (1) and (2) show the constraints to the communication network, i.e., the aggregate rate on each link cannot exceed the link bandwidth and the accumulated rate at each node cannot exceed the node processing capability.

The aim of transmitting a flow of packets from its source to the destination is to get some benefit from the information

⋆ This work was supported by the Australian Research Council's Discovery Projects Scheme (Project number FF0455875).

transmission (e.g., downloading a file, reading news, or making an online booking, etc.). It is natural to set a utility function $U_i$ for flow $i$, and assume that $U_i$ is related to its rate $x_i$. Thus, we can denote it as $U_i(x_i)$. Finally, the congestion control can be achieved by solving the following optimization problem

$$\begin{aligned} \max_x \ & \sum_i U_i(x_i) \\ \text{subject to} \ & Ax \le C \\ & Bx \le D \\ \text{over} \ & x \ge 0 \end{aligned} \tag{3}$$

In order to solve the above optimization problem, define the Lagrangian

$$\begin{aligned} L(x,p,q) &= \sum_i U_i(x_i) - \sum_l p_l \left( \sum_{i \in F(l)} x_i - C_l \right) \\ &\quad - \sum_n q_n \left( \sum_{i \in F(n)} x_i - D_n \right) \\ &= \sum_i \left[ U_i(x_i) - x_i \left( \sum_{l \in L(i)} p_l + \sum_{n \in N(i)} q_n \right) \right] \\ &\quad + \sum_l p_l C_l + \sum_n q_n D_n \end{aligned} \tag{4}$$

where $F(l)$ is the set of flows which go through link $l$, $F(n)$ is the set of flows which go through node $n$, and $L(i)$ and $N(i)$ are sets of links and nodes on flow $i$'s route, respectively.

The optimal flow rate $x_i$ satisfies

$$\frac{\partial L(x,p,q)}{\partial x_i} = 0 \tag{5}$$

Substituting equation (5) into (4), we get

$$U_i'(x_i) = \sum_{l \in L(i)} p_l + \sum_{n \in N(i)} q_n \tag{6}$$

To solve $x_i$ from (6), we assume the following assumption holds:

*Assumption 1:* on the interval $I_i = [m_i, M_i]$, the utility functions $U_i(x_i)$ are increasing, strictly concave, and twice continuously differentiable.

Based on equation (6) and *Assumption 1* we have

$$x_i = \left[ U_i'^{-1} \left( \sum_{l \in L(i)} p_l + \sum_{n \in N(i)} q_n \right) \right]_{m_i}^{M_i} \tag{7}$$

where $[z]_a^b = \min\{\max\{z,a\}, b\}$ restricts the resulted $x_i$ within a reasonable range $[m_i, M_i]$, and $U_i'^{-1}$ is inverse of $U_i'$.

To get optimal values of $p_l$ and $q_n$, we can define the dual function as

$$H(p,q) = \max_x L(x,p,q) \tag{8}$$

and the dual problem as

$$\min_{p \ge 0, q \ge 0} H(p,q) \tag{9}$$

This dual problem can be solved using the gradient projection method [Bertsekas et al. (1997), p. 212]

$$\begin{aligned} p_l(t+1) &= \left[ p_l(t) - \alpha \frac{\partial H}{\partial p_l}(p(t), q(t)) \right]^+ \\ &= \left[ p_l(t) + \alpha \left( \sum_{i \in F(l)} x_i - C_l \right) \right]^+ \end{aligned} \tag{10}$$

$$\begin{aligned} q_n(t+1) &= \left[ q_n(t) - \beta \frac{\partial H}{\partial q_n}(p(t), q(t)) \right]^+ \\ &= \left[ q_n(t) + \beta \left( \sum_{i \in F(n)} x_i - D_n \right) \right]^+ \end{aligned} \tag{11}$$

where $\alpha$ and $\beta$ are stepsizes and $[z]^+ = \max\{z, 0\}$.

### 2.2 Algorithm

Based on the above derivation, we have the following decentralized primal-dual algorithm:

- *Initial conditions*

$$\begin{aligned} \text{Link } l &: p_l(0) \ge 0; \\ \text{Node } n &: q_n(0) \ge 0; \\ \text{Flow } i &: x_i(0) \in [m_i, M_i]. \end{aligned}$$

- *Link $l$'s Algorithm:* At time $t = 1, 2, \cdots$, link $l$:
  - · gets rates of flows which go through link $l$;
  - · computes its price using equation (10)
    $$p_l(t+1) = [p_l(t) + \alpha(x^l(t) - C_l)]^+$$
    where $x^l(t) = \sum_{i \in F(l)} x_i(t)$ is the aggregate flow rate on link $l$;
  - · communicates the new price $p_l(t+1)$ to all flows which go through $l$.
- *Node $n$'s Algorithm:* At time $t = 1, 2, \cdots$, node $n$:
  - · gets rates of flows which go through node $n$;
  - · computes its price using equation (11)
    $$q_n(t+1) = [q_n(t) + \beta(x^n(t) - D_n)]^+$$
    where $x^n(t) = \sum_{i \in F(n)} x_i(t)$ is the aggregate flow rate at node $n$;
  - · communicates the new price $q_n(t+1)$ to all flows which go through node $n$.
- *Flow $i$'s Algorithm:* At time $t = 1, 2, \cdots$, flow $s$:
  - · receives prices of links and nodes which flow $i$ goes through;
  - · chooses a new transmission rate $x_i(t+1)$ using equation (7)
    $$x_i(t+1) = [U_i'^{-1}(p^i(t) + q^i(t))]_{m_i}^{M_i}$$
    where $p^i(t) = \sum_{l \in L(i)} p_l(t)$ is the aggregate price of all the links on flow $i$'s route and $q^i(t) = \sum_{n \in N(i)} q_n(t)$ is the aggregate price of all the nodes on flow $i$'s route;
  - · communicates new rate $x_i(t+1)$ to links and nodes on its route.

## 3. CONVERGENCE OF THE ALGORITHM

In this section, we will prove the convergence of the algorithm. But before that, we assume that the utility function satisfies the following assumption.

*Assumption 2:* the curvatures of $U_i$ are bounded away from zero by $-U_i''(x_i) \geq 1/\alpha_i > 0$.

Because of the similarity in the definitions of $p$ and $q$, we can combine them together by $r = \begin{bmatrix} p \\ q \end{bmatrix}$ to simplify our proof. Then the dual function can be rewritten by $H(r) = H(p, q)$.

Next, we prove that using the algorithm, the dual problem converges to the optimal point.

*Theorem 1.* Assume *Assumption 1* and *2* hold, and the stepsizes $\alpha, \beta > 0$ are small enough. The limit point $r^*$ of the sequence $r(t)$ generated from the algorithm is optimal.

**Proof.** From *Assumption 1* we know that $H(r)$ is convex, lower bounded, and continuously differentiable.

Next, we will prove that $H(r)$ satisfies the Lipschitz condition, i.e., there exists a constant $K$ such that $\|\nabla H(r) - \nabla H(g)\|_2 \leq K\|r - g\|_2$, $\forall r, g \geq 0$.

According to the Mean Value Theorem [Bertsekas et al. (1997), p. 639], we can find $w = tr + (1-t)g \geq 0$, $t \in [0, 1]$ such that $\nabla H(r) - \nabla H(g) = \nabla^2 H(w)(r - g)$. So we have

$$\begin{aligned} \|\nabla H(r) - \nabla H(g)\|_2 &= \|\nabla^2 H(w)(r - g)\|_2 \\ &\leq \|\nabla^2 H(w)\|_2 \|r - g\|_2 \end{aligned} \quad (12)$$

From the definition of $H(w)$, we have

$$\nabla H(w) = E - Fx(w) \quad (13)$$

where $E = \begin{bmatrix} C \\ D \end{bmatrix}$ and $F = \begin{bmatrix} A \\ B \end{bmatrix}$. Then

$$\nabla^2 H(w) = -F \cdot \left[\frac{\partial x(w)}{\partial w}\right] \quad (14)$$

The element of the matrix $\left[\frac{\partial x(w)}{\partial w}\right]$ is given by

$$\frac{\partial x_i(w)}{\partial w_l} = \frac{F_{li}}{U_i''(x_i(w))} \quad (15)$$

So we have

$$\left[\frac{\partial x(w)}{\partial w}\right] = UF^T \quad (16)$$

where $U = \text{diag}(1/U_i''(x_i(w)))$ and $F^T$ is the transpose of $F$. Substituting (16) into (14), we get

$$\nabla^2 H(w) = -FUF^T \quad (17)$$

The above equation shows that $\nabla^2 H(w)$ is a symmetric matrix. Hence $\|\nabla^2 H(w)\|_1 = \|\nabla^2 H(w)\|_\infty$.

We know the norms of matrix satisfy the inequality [Bertsekas et al. (1997), P635]

$$\|\nabla^2 H(w)\|_2^2 \leq \|\nabla^2 H(w)\|_1 \cdot \|\nabla^2 H(w)\|_\infty \quad (18)$$

Then we have

$$\begin{aligned} \|\nabla^2 H(w)\|_2 &\leq \|\nabla^2 H(w)\|_1 \\ &= \max_l \sum_k [\nabla^2 H(w)]_{kl} \\ &= \max_l \sum_k \sum_i F_{ki} \frac{-1}{U_i''(x_i(w))} F_{li} \\ &= \max_l \sum_i \left[\frac{-1}{U_i''(x_i(w))} F_{li} \sum_k F_{ki}\right] \end{aligned} \quad (19)$$

Here $\sum_k F_{ki}$ is the number of nodes and links along the route of flow $i$. We set $\overline{L} = \max_i \sum_k F_{ki}$. $\sum_i F_{li}$ is the number of flows sharing a node or link. We set $\overline{F} = \max_l \sum_i F_{li}$. From *Assumption 2*, we know that $\frac{-1}{U_i''(x_i(w))} \leq \alpha_i$. We set $\overline{\alpha} = \max_i \alpha_i$. Consequently, (19) becomes

$$\begin{aligned} \|\nabla^2 H(w)\|_2 &\leq \max_l \sum_i \left[\frac{-1}{U_i''(x_i(w))} F_{li} \sum_k F_{ki}\right] \\ &\leq \overline{\alpha}\overline{F}\overline{L} \end{aligned} \quad (20)$$

Define $K = \overline{\alpha}\overline{F}\overline{L}$. Then equation (12) becomes

$$\|\nabla H(r) - \nabla H(g)\|_2 \leq K\|r - g\|_2 \quad (21)$$

Thus the Lipschitz condition holds.

Based on the above conditions, if we set $0 < \alpha, \beta < 2/K$, then the limit point $r^*$ is optimal [Bertsekas et al. (1997), P214]. ■

*Corollary 2.* Assume *Assumptions 1* and *2* hold, and the stepsizes $\alpha$ and $\beta$ satisfy $0 < \alpha, \beta < 2/K$. Starting from any initial conditions $m \leq x(0) \leq M$ and $r(0) \geq 0$, the algorithm converges to the primal-dual optimal point.

**Proof.** According to Theorem 1, we know that $r^* = \lim_{t\to\infty} r(t)$ is dual optimal.

According to *Assumption 1*, $U_i$ is increasing, strictly concave, and twice continuously differentiable. Hence $U_i'$ is continuous and one-to-one. Moreover, $U_i'$ is defined on a compact set $[m_i, M_i]$. So $x(r) = U_i'^{-1}(r)$ is continuous. According to (7), the optimal flow rate is

$$x^* = \lim_{t\to\infty} x(t) = \lim_{t\to\infty} x(r(t)) = x(r^*) \quad (22)$$

So $(x^*, r^*)$ is primal-dual optimal [Bertsekas et al. (1997), p. 664]. ■

## 4. SIMULATIONS

### *4.1 Network structure*

In order to testify our algorithm, we consider a random network [Erdős et al. (1960)] with $N = 100$ nodes and $L = 200$ links. To construct this network, first we generate 100 isolated nodes. Then 200 pairs of nodes are randomly selected and links are added between these pairs. (Assume there is no duplicate connection or self-connection). Finally we randomly choose 400 pairs of nodes as sources and destinations and generate $F = 400$ flows. Once a source-destination pair is determined, a route for this flow is determined from its source to the

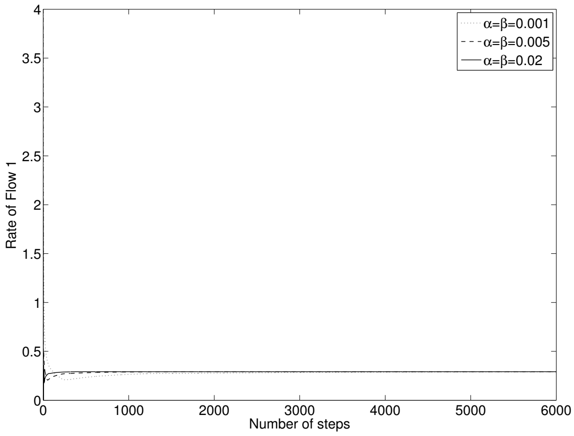


Fig. 1. Convergence rate of the iteration of flow rate. In each simulation we keep $\sum_l C_l = 1000$ and $\sum_n D_n = 1000$.

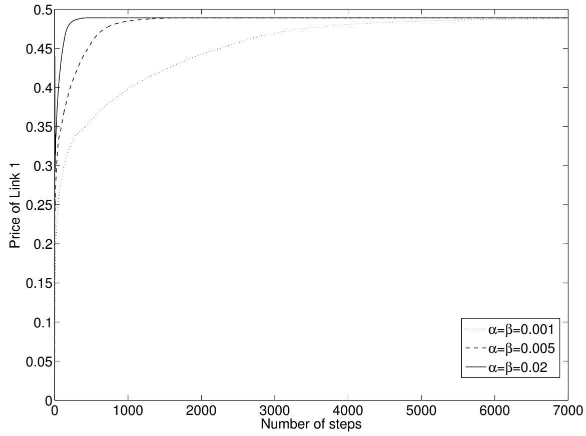


Fig. 2. Convergence rate of the iteration of link price. In each simulation we keep $\sum_l C_l = 1000$ and $\sum_n D_n = 1000$.

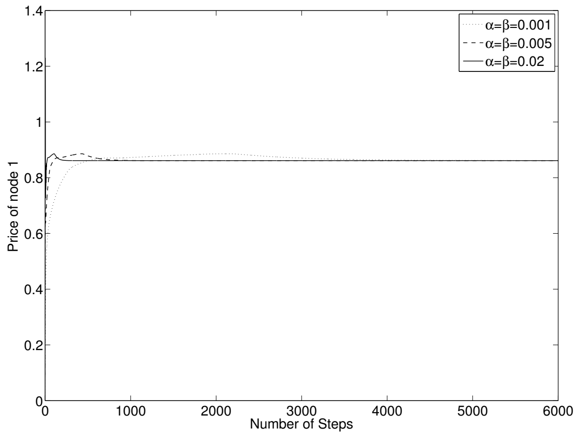


Fig. 3. Convergence rate of the iteration of node price. In each simulation we keep $\sum_l C_l = 1000$ and $\sum_n D_n = 1000$.

destination based on a shortest path rule. If there are more than one shortest path between the source-destination pair, then we randomly choose one.

In the network, the demand for processing capability is different from node to node. Generally speaking, hubs need more capability since more flows go through them. In our network setting, the flows are randomly generated using the shortest path rule. So we can set the processing capability of each node to be proportional to the number of shortest paths between any pair of nodes which go through this node. The latter is called *betweenness* in graph theory [Newman (2003)]. In our simulation we set

$$D_n = \gamma X_n \tag{23}$$

where $\gamma$ is a constant and $X_n$ is the betweenness of node $n$.

Similarly, links connected to hubs need larger capacities since more traffic load goes through them. Here we set

$$C_{mn} = \phi(X_m + X_n), \tag{24}$$

where $C_{mn}$ is the capacity of the link between nodes $m$ and $n$, and $\phi$ is a constant.

### *4.2 Performance indicators*

The traffic performance can be measured in two aspects: the aggregate system utility and the utilization ratio of network resources. Since the task of the optimization problem is to adjust the flow rates to find the maximum utility, we can use the aggregate utility, $\sum_i U_i(x_i)$, as our first performance indicator. In the following simulations, we use

$$U_i(x_i) = d_i \log(x_i + 1) \tag{25}$$

as the utility function, where $d_i$ is the number of links along the route of flow $i$. This utility function is actually a modified version of the utility function used in TCP Vegas [Low et at. (2002)].

When the total utility reaches the maximum, there are still some idle resources. That is to say, a proportion of the resources is wasted. From an economical point of view, we want this kind of waste to be as little as possible, or the utilization ratios of system resources to be as high as possible. For this reason, we use the utilization ratios as our second performance indicator. The utilization ratio of the link capacity is defined as

$$u_l = \frac{\sum_l \sum_i A_{li} x_i}{\sum_l C_l}, \tag{26}$$

where $\sum_l \sum_i A_{li} x_i$ is the total occupied bandwidth and $\sum_l C_l$ is the total provided capacity. Similarly, the utilization ratio of the node processing capability is defined as

$$u_n = \frac{\sum_n \sum_i B_{ni} x_i}{\sum_n D_n}, \tag{27}$$

where $\sum_n \sum_i B_{ni} x_i$ is the total occupied node capability and $\sum_n D_n$ is the total provided node capability.

### *4.3 Convergence rate*

In figures 1-3 we plot the convergence of the iteration of the flow rate, the link price and the node price, respectively.

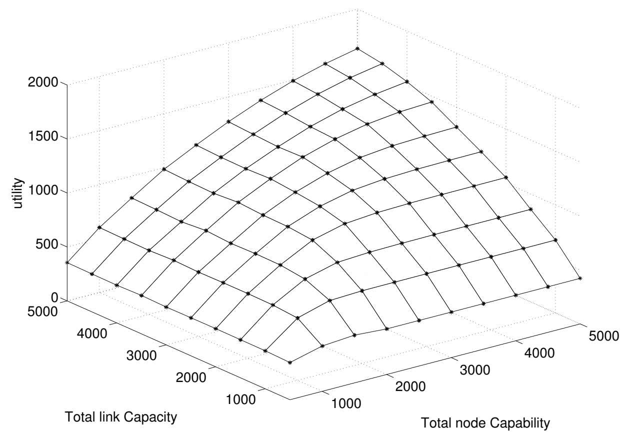


Fig. 4. Aggregate system utility as a function of total link capacity and total node processing capability. The result is averaged over 10 simulations.

Considering the random nature of the network construction we can choose any flow, link, or node, and investigate its convergence, without loss of generality. Here we choose the first flow, the first link, and the first node. Actually, we also investigate the convergence properties of other flows, links and nodes, and find that the results are quite similar.

The figures show that the convergence rate of the algorithm depends on the stepsizes $\alpha$ and $\beta$. The algorithm converges to the optimal point faster if we increase the stepsize. Of course, in order to guarantee the convergence, the values of $\alpha$ and $\beta$ cannot be too large.

### *4.4 Effect of two constraints*

In our algorithm, we consider two constraints on two kinds of system resources — the link capacity and the node processing capability. Next we will study how these constraints affect the system performance.

Figure 4 plots the aggregate system utility as a function of total provided link capacity and total provided node processing capability. The figure show that the aggregate utility is a nondecreasing function of two kinds of system resources. This result agrees with our commonsense that providing more bandwidth or more powerful router benefits the network traffic. However, since we have two kinds of resources, solely enhancing one of them may have little profit because the other one will become the bottleneck of the traffic and prevent the system from achieving higher utility. In figure 4, for example, if the total provided node processing capability is fixed to 500, then the resulting utility is almost unchanged no matter how much the total link capacity is increased.

Figures 5 and 6 plot the utilization ratios of node capability and link capacity, respectively. The bottleneck effect is shown more clearly in these two figures. In figure 5, for example, when the total link capacity is fixed, the larger the node capability is assigned to the network, the higher percentage of it is wasted. The utilization ratio of the node processing capability is high only when the total node capability is equal to or less than the total link capacity. As the result, the surface looks like a waterfall dropped

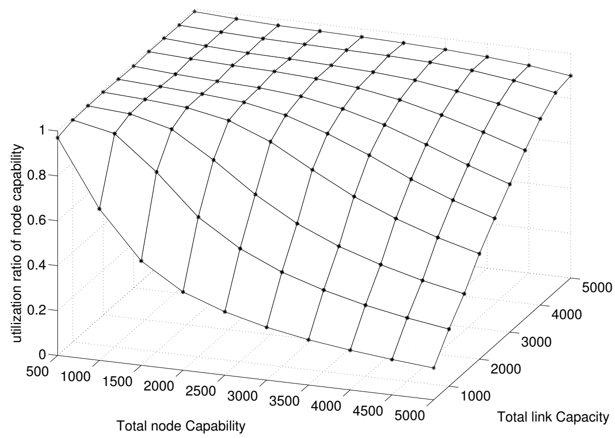


Fig. 5. utilization ratio of node capability as a function of total link capacity and total node processing capability. The result is averaged over 10 simulations.

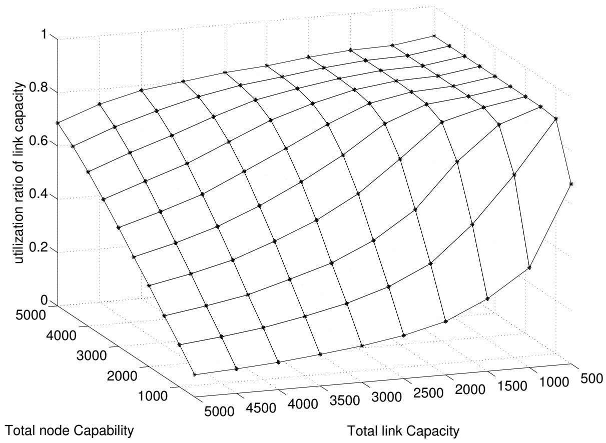


Fig. 6. Utilization ratio of link capacity as a function of total link capacity and total node processing capability. The result is averaged over 10 simulations.

roughly at the diagonal. In figure 6 we also find the utilization ratio of the link capacity displays the waterfall shape.

As a special case, if we keep the total node processing capacity to be infinity, then the link capacity becomes the only constraint. In this case, our algorithm degenerates to the algorithm proposed in [Low et al. (1999), Algorithm A1]. Figure 7 compares the result of our algorithm with that one. Because Algorithm A1 in [Low et al. (1999)] does not consider the node processing capability constraint, their optimal utility is a horizon line in the figure. On the contrary, the total node processing capability is one of the main reasons to limit the aggregate utility in our algorithm, especially when it is small. As it increases, the aggregate utility also increases. However, when it is large enough, it does not constrain the aggregate utility anymore. Then continuing increasing it has no help to the aggregate utility because the link capacity becomes the bottleneck. In this case the effect of our algorithm is just as same as Algorithm A1 in [Low et al. (1999)].

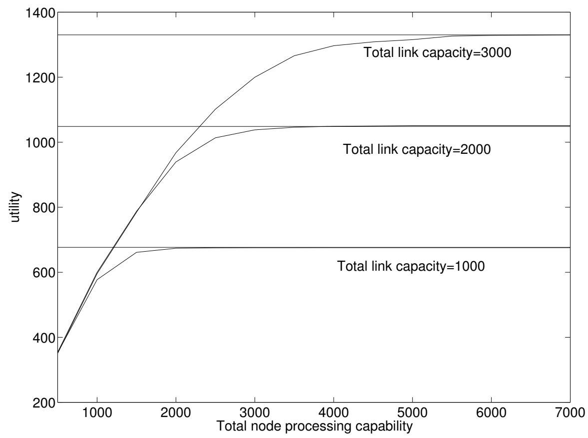


Fig. 7. Aggregate system utility changes with the total node processing capability. Here we plot the cases as the total link capacity equals to 1000, 2000, and 3000. Horizon lines are the results using Algorithm A1 in [Low et al. (1999)], with the corresponding total link capacity. The result is averaged over 10 simulations.

## 5. CONCLUSIONS

In the Internet, congestion may happen due to limited link capacity or node processing capability. In this paper we propose a decentralized primal-dual algorithm to solve the congestion control problem subject to these constraints. We also prove the convergence of the algorithm.

Using this algorithm, we can investigate the traffic performance. Since there are two kinds of system resources related to two constraints, solely increasing one of them cannot benefit the system performance too much as the other one will become the bottleneck of the traffic.

## REFERENCES

D.P. Bertsekas, and J.N. Tsitsiklis. *Parallel and distributed computation: Numerical methods.* Athena Scientific, Belmont, MA, 1997.

R. Diestel. *Graph theory, 3rd ed.* Springer, Berlin, c2005.

J. C. Doyle, D. L. Alderson, L. Li, S. Low, M. Roughan, S. Shalunov, R. Tanaka, and W. Willinger. The "robust yet fragile" nature of the Internet. *Proceedings of the National Academy of Sciences*, 102: 14497-14502, 2005.

P. Erdős and A. Rényi. On the evolution of random graphs. *Publ. Mah. Inst. Hung. Acad. Sci.*, 5: 17-60, 1960.

F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49: 237-252, 1998.

S. Liu, T. Basar, and R. Srikant. Controlling the Internet: a survey and some new results. *Proceedings of the 42nd IEEE Conference on Decision and Control*, Hawaii USA, page 3048-3057, 2003.

S. H. Low and D. E. Lapsley. Optimization flow control I: basic algorithm and convergence. *IEEE/ACM Trans. on Networking*, 7: 861-874, 1999.

S. H. Low, F. Paganini, and J. C. Doyle. Internet congestion control. *IEEE Control System Magzine*, 20: 2-43, 2002.

M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45: 167–256, 2003.

F. Paganini, Z. Wang, J. C. Doyle, and S. H. Low. Congestion control for high performance, stability, and fairness in general networks. *IEEE/ACM Trans. on Networking*, 13: 43-56, 2005.

R. Srikant. *The Mathematics of Internet congestion control.* Birkhauser, Boston, c2004.