Flow Control with Processing Constraint

Song Chong, *Member, IEEE*, Minsu Shin, *Student Member, IEEE*,
Jeonghoon Mo, *Member, IEEE*, and Hyang-Won Lee, *Student Member, IEEE*

***Abstract*— In this letter we re-examine flow control issues for a network environment where transmission links and CPUs on a data path can be jointly bottlenecked. We show that flow control without consideration of CPU congestion can significantly lose both fairness and efficiency in this environment. As a solution to this problem, we establish the notion of dual-resource proportional fairness, propose a distributed algorithm to achieve this objective and demonstrate its performance through simulations.**

***Index Terms*— Flow control, transmission link capacity, CPU capacity, fairness, efficiency, proportional fairness.**

I. Introduction

TRADITIONALLY congestion control research has focused on managing network bandwidth. But the fast pace increase in bandwidth does not make network bandwidth the only scarce resource. Because of advances in optical network technology, the rate at which physical bandwidth increases has far surpassed that of other resources such as CPU and memory bandwidth. Thus, bottlenecks shift from bandwidth to other resources. Furthermore, the rise of new applications that require in-network processing hastens this shift. For instance, a voice-over-IP call made from a cell phone to a PSTN phone must go through a media gateway that performs audio transcoding "on the fly" as the two end points often use different audio compression standards. Examples of in-network processing services are increasingly abundant from security, performance-enhancing proxies (PEP), to media translation. These services add additional loads to processing capacity in the network components. New router technologies such as extensible routers [1] also need to deal with scheduling of CPU usage per packet as well as bandwidth usage per packet.

In this letter, we re-examine flow control issues for an environment where both bandwidth and processing resources can be a bottleneck. The environment where a network component has to manage both resources simultaneously may not be a present concern of the Internet. However, we envision (also it is indeed happening now to some degree) that diverse network services would reside somewhere inside the Internet, most likely at the edge, processing, storing, and/or forwarding data packets on the fly. We are particularly focusing on examining whether the current flow control theory can be applied to this environment without modification, or if not, what solutions we can use to fix the problem.

Manuscript received April 28, 2005. The associate editor coordinating the review of this letter and approving it for publication was Dr. Nikos Nikolaou. This research was supported by the Ministry of Information and Communication, Korea, under the grant for BrOMA-ITRC program supervised by IITA.

Song Chong, Minsu Shin, and Hyang-Won Lee are with the Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea (e-mail: song@ee.kaist.ac.kr; {msshin, mslhw}@netsys.kaist.ac.kr).

Jeonghoon Mo is with Information and Communications University (ICU), Taejon, Korea (e-mail: jhmo@icu.ac.kr).

Digital Object Identifier 10.1109/LCOMM.2005.10029.

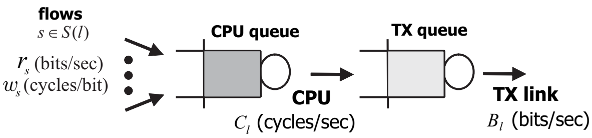


Fig. 1. Link model.

II. Flow Control in Dual-Resource Environment

A. Network Model

We consider a network that consists of a set of unidirectional links, $L = \{1, \cdots, L\}$, where each link l has a packet processing resource (i.e., CPU) with capacity C_l (cycles/sec) and a packet transmission resource (i.e., transmitter(TX) link) with capacity (or bandwidth) B_l (bits/sec) in cascade as in Fig. 1. The network resources are shared by a set of flows (or data sources), $S = \{1, \cdots, S\}$. Each flow s has its data rate r_s (bits/sec) and its route to destination, defined by a set of links, $L(s) \subset L$. Let $S(l) = \{s \in S | l \in L(s)\}$ be the set of flows that use link l. We assume there is a set of applications, a flow is an instance of an application, and flows have different processing demands depending on their applications. We represent this notion by *processing density* w_s of each flow s which is defined to be an average processing cycles required per bit. The processing demand of a flow s is then $w_s r_s$ (cycles/sec). Since there are limits on the processing and bandwidth capacities, the amount of processing and bandwidth usage by all flows sharing the link must be less than or equal to the capacities at anytime. We represent this notion by the following two constraints: for each link $l \in L$, $\sum_{s \in S(l)} w_s r_s \leq C_l$ (*processing constraint*) and $\sum_{s \in S(l)} r_s \leq B_l$ (*bandwidth constraint*). This will be called as *dual-resource environment* and a nonnegative rate vector $r = [r_1, \cdots, r_S]^T$ satisfying these dual constraints for all links $l \in L$ is said to be *feasible*.

B. Proportional fairness in dual-resource environment

Fairness and efficiency are two main objectives in flow control. The notion of fairness and efficiency has been extensively studied and well understood in traditional bandwidth allocation problems [2] [3]. However, it is unclear that this notion would be readily extended to the dual-resource environment where processing and bandwidth resources can be jointly limited and thus must be jointly managed. Consider a single link case (say, link l only) as an example. Suppose that the link operates in the bandwidth-limited region as traditional bandwidth allocation problems assume. Then, the fair and efficient rate allocation is obviously $r_s = \frac{B_l}{|S(l)|}$ for all $s \in S(l)$ where $|S(l)|$ denotes the cardinality of $S(l)$, which can be shown to be *proportionally fair sharing* of B_l [2]. Similarly, if we suppose that the link operates in the processing-limited region, the fair and efficient rate allocation must be $r_s = \frac{C_l}{w_s|S(l)|}$ for all $s \in S(l)$ which yields proportionally fair sharing of C_l. The interesting case is when processing and bandwidth resources are jointly limited. We need to answer when this case occurs, what should be the fair and efficient rate allocation in this case and how this notion of fairness and efficiency can be extended to multiple link cases.

Consider the following aggregate log utility maximization problem with dual constraints:

$$\max_{r} \sum_{s \in S} \log r_s \tag{1}$$

$$\text{subject to} \quad \textstyle\sum_{s\in S(l)} w_s r_s \le C_l, \quad \forall\, l \in L \tag{2}$$

$$\textstyle\sum_{s\in S(l)} r_s \le B_l, \quad \forall\, l \in L \tag{3}$$

$$r_s \ge 0, \quad \forall\, s \in S. \tag{4}$$

The solution r^* of this problem is unique since it is a strictly concave maximization problem over a convex set [4]. Furthermore, r^* is proportionally fair since $\sum_{s\in S} \frac{r_s - r_s^*}{r_s^*} \le 0$ holds for all feasible r by the optimality condition of the problem. Note, however, that this r^* can be different from the one found in Kelly's formulation [2] since the feasible set of r can be changed with the additional processing constraint.

From the duality theory [4], r^* satisfies that

$$r_s^* = \frac{1}{\sum_{l\in L(s)} (w_s\theta_l^* + \pi_l^*)}, \quad \forall\, s \in S, \tag{5}$$

where $\theta^*=[\theta_1^*, \cdots, \theta_L^*]^T$ and $\pi^*=[\pi_1^*, \cdots, \pi_L^*]^T$ are Lagrange multiplier vectors for (2) and (3), respectively, and θ_l^* and π_l^* can be interpreted as congestion prices of CPU and TX of link l, respectively. Eq. (5) reveals an interesting property that the optimal rate of each flow is inversely proportional to the aggregate link price of its route and each link price is given by $w_s\theta_l^* + \pi_l^*$, i.e., sum of CPU congestion price θ_l^*, multiplied by the processing density of the flow w_s, and TX congestion price π_l^*. The congestion price θ_l^* or π_l^* is positive only when the corresponding resource becomes a bottleneck, and is zero, otherwise. Therefore, for each link l, we can classify its status as follows: *processing-limited* (or *CPU-limited*) if $\theta_l^* > 0$ and $\pi_l^* = 0$, *bandwidth-limited* (or *BW-limited*) if $\theta_l^* = 0$ and $\pi_l^* > 0$, and *jointly-limited* if $\theta_l^* > 0$ and $\pi_l^* > 0$.

We now apply this classification to a single link case (say, link l only). There are three different cases as below. Let $\bar{w}_a = \frac{\sum_{s\in S(l)} w_s}{|S(l)|}$ and $\bar{w}_h = \left(\frac{\sum_{s\in S(l)} \frac{1}{w_s}}{|S(l)|}\right)^{-1}$ which are the arithmetic and harmonic means of the processing densities of flows sharing the link l, respectively.

1) *Processing-limited case* ($\theta_l^* > 0$ and $\pi_l^* = 0$): In this case, $r_s^* = \frac{1}{w_s\theta_l^*}$, $\forall s \in S(l)$, $\sum_{s\in S(l)} w_s r_s^* = C_l$ and $\sum_{s\in S(l)} r_s^* \le B_l$. By combining these, we know that this case occurs when $\frac{C_l}{B_l} \le \bar{w}_h$ and the rate allocation becomes $r_s^* = \frac{C_l}{w_s|S(l)|}$, $\forall s \in S(l)$, i.e., processing capacity is fully utilized and equally shared by flows.
2) *Bandwidth-limited case* ($\theta_l^* = 0$ and $\pi_l^* > 0$): In this case, $r_s^* = \frac{1}{\pi_l^*}$, $\forall s \in S(l)$, $\sum_{s\in S(l)} w_s r_s^* \le C_l$ and $\sum_{s\in S(l)} r_s^* = B_l$. By combining these, we know that this case occurs when $\frac{C_l}{B_l} \ge \bar{w}_a$ and the rate allocation becomes $r_s^* = \frac{B_l}{|S(l)|}$, $\forall s \in S(l)$, i.e., bandwidth is fully utilized and equally shared by flows.
3) *Jointly-limited case* ($\theta_l^* > 0$ and $\pi_l^* > 0$): This case occurs when $\bar{w}_h < \frac{C_l}{B_l} < \bar{w}_a$. By plugging $r_s^* = \frac{1}{w_s\theta_l^* + \pi_l^*}$, $\forall s \in S(l)$, into $\sum_{s\in S(l)} w_s r_s^* = C_l$ and $\sum_{s\in S(l)} r_s^* = B_l$, we can obtain θ_l^*, π_l^* and consequently r_s^*, $\forall s \in S(l)$.

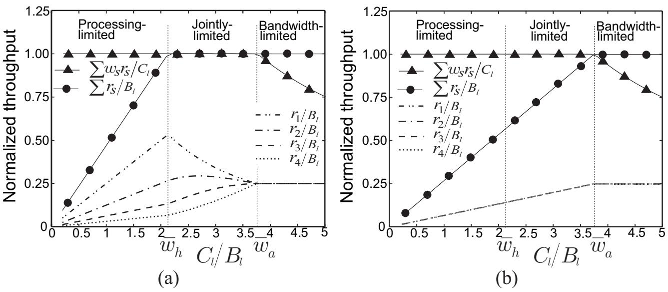
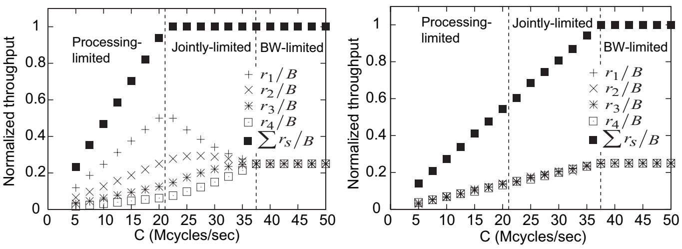


Fig. 2. Fairness and efficiency in dual-resource environment (single link case): (a) dual-resource flow control, (b) bandwidth-oriented flow control.

Fig. 2 (a) illustrates this property using an example of four flows with different processing densities (w_1, w_2, w_3, w_4) = (1, 2, 4, 8) where $\bar{w}_h$=2.13 and $\bar{w}_a$=3.75. In the jointly-limited case, 100% efficiency is always maintained in both bandwidth and CPU usages irrespective of $\frac{C_l}{B_l}$, whereas there occurs a good compromise between bandwidth fairness and CPU fairness as $\frac{C_l}{B_l}$ varies. That is, as bandwidth becomes more scarce than CPU ($\frac{C_l}{B_l} \uparrow \bar{w}_a$), bandwidth fairness becomes more emphasized and hence improves while CPU fairness degrades, and vice versa. For comparison, we consider a *hypothetical* flow control scheme which neglects CPU congestion such that the rate allocation solely depends on TX congestion by $r_s = \frac{1}{\sum_{l\in L(s)} \pi_l}$, $\forall s \in S(l)$ (we call this as *bandwidth-oriented flow control*). Note that this has the same form as Kelly's rate allocation [2] but the difference is that it can incur packet drops at CPU queues in the dual-resource environment because the processing constraint (2) can be infeasible. For the single link case, the bandwidth-oriented flow control with dual constraints gives two possibilities at equilibrium: if $\frac{C_l}{B_l} \ge \bar{w}_a$, $r_s = \frac{B_l}{|S(l)|}$, otherwise, $r_s = \frac{C_l}{\bar{w}_a|S(l)|}$. This is depicted in Fig. 2 (b) and compared with Fig. 2 (a). As expected, both fairness and efficiency significantly degrade in the processing-limited case as well as the jointly-limited case, which clearly demonstrates the importance of considering CPU congestion in controlling flows in the dual-resource environment.

C. Asynchronous and distributed algorithm

By appealing to the duality theory and using the gradient projection method as in [3], the congestion prices can be solved by the following distributed iterations: for link $l \in L$, $\theta_l(t+1) = [\theta_l(t) - \gamma(C_l - \sum_{s\in S(l)} w_s r_s(t))]^+$



(a) (b)

Fig. 3. Simulation results in dumbell topology (single link case): (a) dual-resource flow control, (b) bandwidth-oriented flow control.

and $\pi_l(t+1) = [\pi_l(t) - \gamma(B_l - \sum_{s\in S(l)} r_s(t))]^+$ where γ is a positive step size. On the other hand, the rates can be solved by the following distributed equations: for source $s\in S$, $r_s(t) = [\sum_{l\in L(s)}(w_s\theta_l(t) + \pi_l(t))]^{-1}$ where the path price $\sum_{l\in L(s)}(w_s\theta_l(t) + \pi_l(t))$ is known to source s by feedbacks. The following theorem establishes the global convergence of this distributed algorithm in an asynchronous environment. Define w=$\max\{\max_{s\in S} w_s, 1\}$, $\alpha = \max\{\max_{l\in L} B_l, \max_{l\in L, s\in S(l)} \frac{C_l}{w_s}\}$, A_1=$\max_{l\in L}|S(l)|$, A_2=$\max_{s\in S}|L(s)|$ and $A_3 = \max_{l\in L}\{\sum_{s\in S(l)}|L(s)|\}$.

Theorem 2.1: Suppose that communication delays between sources and links are bounded by D respectively, and price and rate updates are asynchronous with intervals bounded by D respectively. Then, if $0 < \gamma < [w^2\alpha^2(A_1A_2 + 4A_3D)]^{-1}$, starting from any nonnegative $(r(0), \theta(0), \pi(0))$, the sequence $\{(r(t), \theta(t), \pi(t))\}$ generated by the asynchronous and distributed algorithm converges to the unique primal-dual optimal solution of the problem (1)-(4).

Proof: The proof is straightforward following a standard technique [3], and is omitted due to the limited space. ■

III. SIMULATION RESULTS

Using ns-2, we simulate both single link and multiple link scenarios. For the single link scenario, we use a dumbell topology with fixed bandwidth B=40Mbps and varying CPU capacity C in [5M, 50M] (cycles/sec). The resources are shared by four application groups with different processing densities (0.25, 0.5, 1.0, 2.0) and each group consists of 10 flows. All the flows experience 40msec round-trip propagation delay identically. Fig. 3 shows the result where r_s, $s = 1, 2, 3, 4$, represents the aggregate throughput of 10 type s application flows. This result confirms that the proposed asynchronous and distributed control indeed achieves the optimality presented in Fig. 2 (a) and significantly outperforms the bandwidth-oriented counterpart neglecting $\theta(t)$. For the multiple link simulations, we use a parking lot topology where we consider two scenarios. In the first scenario shown in Fig. 4, we vary the bandwidth of link L1, B_1, in [20, 100] (Mbps) while fixing other capacities (B_2=B_3=40Mbps, C_1=C_2=C_3=30Mcycles/sec) to construct four different operating regions (I, II, III and IV) such that {L1, L2, L3} is {BW, CPU, CPU}-limited in I, {BW, CPU, jointly}-limited in II, {jointly, CPU, jointly}-limited in III and {CPU, CPU, jointly}-limited in IV. Note that L1 status changes from bandwidth-limited to processing-limited via jointly-limited as B_1 increases. The second scenario is symmetric to the first

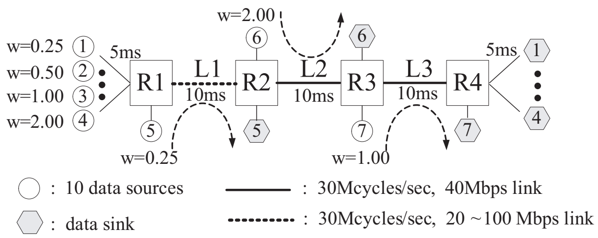
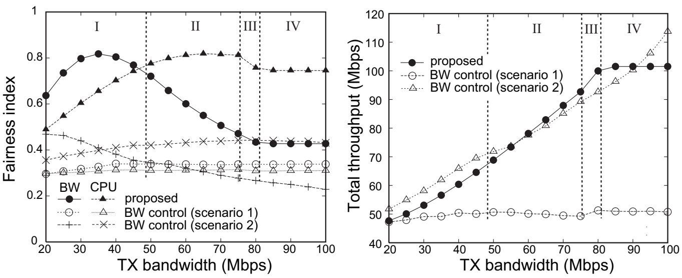


Fig. 4. Multiple link simulation scenario in parking lot topology.



(a) (b)

Fig. 5. Simulation results in parking lot topology (multiple link case): (a) bandwidth and CPU fairness indices, (b) total throughput.

scenario in that we switch the locations of group 5 and group 7 flows and vary B_3 in [20, 100] (Mbps) while fixing other capacities (B_1=B_2=40Mbps, C_1=C_2=C_3=30Mcycles/sec). The proposed dual-resource control yields identical resource allocations for these two symmetric scenarios but the bandwidth-oriented flow control does not since packet drops due to the infeasibility of (2) occur differently in these two scenarios. As shown in Fig. 5 (a), the proposed dual-resource control yields much better fairness than the bandwidth-oriented control in both bandwidth and CPU usages, in all four regions, in both scenarios where we use $\frac{(\sum_{s\in S} r_s)^2}{|S|(\sum_{s\in S} r_s^2)}$ and $\frac{(\sum_{s\in S} w_s r_s)^2}{|S|(\sum_{s\in S} w_s^2 r_s^2)}$ as bandwidth and CPU fairness measures, respectively. More importantly, the proposed control explicitly trades bandwidth fairness for CPU fairness as the network becomes more CPU-limited with increasing B_1 (or B_3), whereas the bandwidth-oriented control does not. Moreover, this advantage in fairness comes without losing efficiency; as shown in Fig. 5 (b), the total throughput in the proposed control case is much larger than (scenario 1) or comparable to (scenario 2) that in the bandwidth-oriented control case.

IV. FUTURE WORK

An interesting future work is how to implement the developed concept and theory in the current Internet architecture. A scalable and viable approach might be a TCP/AQM modification to take CPU congestion into account.

REFERENCES

[1] Y. Gottlieb and L. Peterson, "A comparative study of extensible routers," in *Proc. IEEE OpenArch*, June 2002, pp. 51–62.

[2] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *J. of the Operational Research Society*, vol. 49, pp. 237–252, Apr. 1998.

[3] S. H. Low and D. E. Lapsley, "Optimization flow control I: Basic algorithm and convergence," *IEEE/ACM Trans. Networking*, vol. 7, pp. 861–875, Dec. 1999.

[4] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA: Athena Scientific, 1999.