Journal of University of Science and Technology Beijing
Volume 10, Number 3, June 2003, Page 66

A rate based congestion control algorithm in networks with coexisting unicast and multicast sessions

Qian Dong and Jianying Xie

School of Electronics & Information Technology, Shanghai Jiaotong University, Shanghai 200030, China

(Received 2002-07-14)

Abstract: The optimal rate control problem in networks with unicast and multirate multicast sessions is investigated. A penalty function approach is used to solve a convex program formulation of this problem, and then a heuristic rate control algorithm is derived. The algorithm is distributed, and suitable both for source-driven unicast sessions and receiver-driven multicast sessions. To obtain practical viability, the computational burden on core routers as well as end-hosts is kept very low, also is the overhead of network congestion feedback. Simulation results show that the algorithm guarantees TCP (Transmission Control Protocol)-based unicast sessions coexisting with multirate multicast sessions in a fair and friendly manner. It is also shown that various fairness criteria of resource allocation could be achieved by choosing appropriate utility functions, and resource-utilizing efficiencies would be likewise different.

Key words: multicast; unicast; congestion control; rate control; fairness

1 Introduction

In general, different receivers within the same multicast group always have diverse characteristics, *e.g.*, usable link capacity, data processing capability and QOS (Quality of Service) requirement. Unirate multicasting is likely to overwhelm the slow receivers, meanwhile, starve the fast ones. Multirate multicasting, in which different receivers could receive data at different rates, can be used to accommodate the heterogeneous requirements and capabilities of receivers. It provides more flexibility to users and allows more efficient usage of network resources, but an effective rate control mechanism must be established as a guarantee. The rate control strategy should ensure that the traffic injected to networks remain within the limits that networks could bear, and ensure that networks resources are shared by competing flows in a certain fair manner. About fairness, there already exist many acceptable definitions, and some well-known ones are max-min fairness and proportional fairness [1]. Kelly proposed a nonlinear program formulation of the aggregate utility maximization problem under link capacity constraints [1]. Massoulie demonstrated that various fairness objectives can be realized within a unified framework by choosing different utility functions [2]. Recently, considerable research has been devoted to the problem of fair resource allocation in multicast networks, but most of the work is focused on achieving max-min fairness [3-5]. Although there has been a lot of research on the utility maximization problem in unicast networks, the case with both unicast and multicast traffic didn't attract significant attention. Kar used duality theory to solve the resource allocation problem, and developed an optimization-based rate control algorithm for multirate multicasting [6]. This algorithm takes into account the heterogeneous requirements of multicast receivers, but it has high overhead of computation and communication, and requires network links to maintain per-flow state information. Deb proposed another algorithm based on subgradient approach [7,8], however, the marking scheme, which is implemented at core routers by using a virtual queue [9], is too complicated. Moreover, the marking level $\widetilde{C}_l$ on each link is difficult to determine. In this paper, a penalty function approach is used to solve a convex program formulation, and a distributed rate control algorithm that maximizes the total user utility in networks with coexisting unicast and multicast flows is also derived. A new marking function is obtained through combining RED (Random Early Detection) with one-bit ECN (Explicit Congestion Notification).

2 Multirate multicasting principle

First the principle of multirate multicasting is illustrated as the basis. **Figure 1** shows a multicast tree of a certain multicast group. It is constructed corre-

---

Corresponding author: Qian Dong, **E-mail:** Dong_Qian@hotmail.com

sponding with the shortest path tree. The root S of the tree denotes the multicast traffic source. The forking nodes R_1, R_2 and R_3 are junction nodes. Leaves R_{x1}, R_{x2}, R_{x3} and R_{x4} are multicast receivers. The rest nodes, whose in-degree and out-degree are both 1, are non-junction nodes. The branches denote links with various bandwidths. Following the definitions of graph theory, R_2 is the parent of R_{x1} and R_{x2}, R_1 is the parent of R_2 and R_3, and S is the parent of R_1. Conversely, R_1 is a child of S, R_2 and R_3 are children of R_1, and so on. Due to the heterogeneity of multicast receivers, the rate control algorithm for multirate multicasting is receiver-driven. Each receiver decides its rate based on its utility function and the congestion signals received, and then sends request to its parent. A junction node gathers all such requests from its children, picks out the highest, and requests that rate from its parent. Requests go up the tree through the junction nodes in this upward consolidation way, until finally reach the source. The source issues traffic to its children at their requested rates; these nodes then forward traffic to their own children, and so on; and the traffic finally reaches each receiver at its requested rate. In networks with only unicast traffic, the total traffic rate through a link is the sum of the rates of all sessions. However, if there are multicast sessions passing through that link, from the principle mentioned above, it is known that the total traffic rate of a multicast session is the highest rate of all downstream receivers of that multicast session. Thus each multicast session is characterized by a number of virtual sessions [10], and each receiver of a multicast session could be regarded as a virtual session. As a result, unless a virtual session has the highest rate among all the virtual sessions of the same multicast session passing through a certain link, it does not directly contribute to the congestion on the link. Therefore, any congestion control scheme that treats each of the virtual sessions like separate unicast sessions is not appropriate for this case.

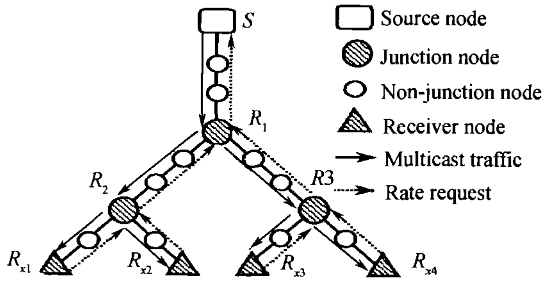

Figure 1 Simplified multirate multicasting networks

3 Optimization model and algorithm

3.1 Model and algorithm of unicast networks

Consider a network consisting of a set L of links, and let C_l be the capacity of link $l \in L$. The network is used by a set R of users. Associate each user $r \in R$ with a route, which is also denoted by r. Route r is a non-empty subset of L. Let $x_r > 0$ denotes the rate allocated to user r, and $U_r(x_r)$ is the utility function of the user. Assume that $U_r(x_r)$ is an increasing, continuously differentiable and strictly concave function of x_r over the range $x_r > 0$, with $U_r'(x_r) \to \infty$ as $x_r \to 0$ and $U_r'(x_r) \to 0$ as $x_r \to \infty$. To analyze the stability and fairness of the rate control algorithm in unicast networks, Kelly developed a mathematical model [11] as the following:

$$\max \sum_{r \in R} \omega_r U_r(x_r) - \beta \sum_{l \in L} P_l\left(\sum_{j: l \in j} x_j\right) \tag{1}$$

Here ω_r is the weight of user r. Parameter β is a non-zero positive constant. $P_l(\cdot)$ is a kind of penalty function. When the total traffic rate through link l is y, $P_l(y)$ could be thought of as a cost. Assume $P_l(y)$ is differentiable, with

$$\frac{\mathrm{d}}{\mathrm{d}y} P_l(y) = p_l(y) \tag{2}$$

Here the function $p_l(y)$ is a positive, continuous, strictly increasing function of y. Thus $P_l(y)$ is a strictly convex function. The nonlinear program problem (1), whose objective is to maximize the aggregate user utility and minimize the network cost, then becomes a convex program, and therefore has a unique optimal solution. Based on this model, Kelly derived the following rate control algorithm:

$$\frac{\mathrm{d}}{\mathrm{d}t} x_r(t) = \kappa_r\left(\omega_r - \frac{\beta}{U_r'(x_r)} \sum_{l \in r} p_l\left(\sum_{j: l \in j} x_j(t)\right)\right) \tag{3}$$

Here κ_r is a non-zero positive constant. Applying Lyapunov stability theory can prove that the rates x_r's, $\forall r \in R$, being adjusted according to equation (3), would converge to a unique stable point which is also the unique optimal solution to the maximization problem (1).

Through using the approximate relation between window and rate: $x_r(t) \approx W_r(t) \div d_r$, where d_r is Round Trip Time and $W_r(t)$ is window size, by Kunniyur obtained a result [12]: window based congestion control algorithm in the congestion avoidance phase can be expressed in terms of the following optimization objective, which all TCP senders adjust their data transmission rate or window size to maximize.

$$\max \sum_{r \in R} -\frac{1}{d_r^2 x_r} - \beta \sum_{l \in L} \int_0^{\sum_{j: l \in j} x_j} \frac{(x - C_l)^+}{x} \mathrm{d}x \tag{4}$$

Here the constant β is determined to be ln2 from the TCP characteristic that every received congestion signal halves the current window size. Equation (4) shows that: by ignoring slow start, time-out, *etc.*, and choosing the utility functions to be $-\frac{1}{x}$, TCP congestion control algorithm can be expressed as an optimization formulation similar to equation (1).

3.2 Model and algorithm of networks with coexisting unicast and multicast sessions

To design a unified algorithm for unicast and multicast sessions, it is necessary to integrate two kinds of sessions into a common model. Let S be the set of unicast and multicast sessions and R_s be the set of all receivers in session $s \in S$. For unicast sessions, R_s is a singleton set. (s, r) denotes a virtual session corresponding to session s. The set $L_{s,r}$ denotes all links that virtual session (s, r) traverses. Let S_l be the set of all sessions passing through link l and V_{sl} be the set of all virtual sessions of the session s sharing link l. let $x_{s,r}(t)$ denotes the rate of the virtual session (s, r) at time t, and $U_{s,r}(x_{s,r})$ denotes the utility function of the virtual session (s, r). Here $p_l(y)$ is defined to be a marking function. In networks where resources are shared by unicast and multirate multicast sessions, the aggregate user utility maximization problem can be viewed as the following nonlinear program.

$$\begin{cases} \max \sum_{s \in S} \sum_{r \in R_s} \omega_{s,r} U_{s,r}(x_{s,r}) \\ \text{s.t} \ \sum_{s \in S_l} \left(\max_{(s,r) \in V_{s,l}} x_{s,r} \right) \le C_l, \forall l \in L \end{cases} \tag{5}$$

Rewrite the above constraint maximization problem in the form of a penalty function formulation as

$$\max \sum_{s \in S} \sum_{r \in R_s} \omega_{s,r} U_{s,r}(x_{s,r}) - \beta \sum_{l \in L} \int_0^{\sum_{s \in S_l} \left(\max_{(s,r) \in V_{s,l}} x_{s,r} \right)} p_l(y) dy \tag{6}$$

Note that equation (6) is similar to equation (1), but multirate multicast brings a non-continuous function $\max(\cdot)$, which makes equation (6) to be a non-separable and non-differentiable problem. Observe that the continuous function $\left(\sum_i x_i^n\right)^{1/n} \to \max_i(x_i)$ as $n \to \infty$, when all x_i's are non-negative real numbers. The function $\max(\cdot)$ can therefore be approximated by the function $\left(\sum_i x_i^n\right)^{1/n}$, and (6) has an approximator:

$$\max \sum_{s \in S} \sum_{r \in R_s} \omega_{s,r} U_{s,r}(x_{s,r}) - \beta \sum_{l \in L} \int_0^{\sum_{s \in S_l} \left(\sum_{(s,r) \in V_{s,l}} x_{s,r}^n \right)^{1/n}} p_l(y) dy \tag{7}$$

It can be proved that equation (7) is a convex program, and has a unique optimal solution. Based on the gradient of the objective function, derive a set of rate control equations as

$$\frac{d}{dt} x_{s,r}(t) = \kappa_{s,r} \left(\omega_{s,r} - \frac{\beta}{U'_{s,r}(x_{s,r})} \sum_{l \in L_{s,r}} p_l \left(\sum_{m \in S_l} \left(\sum_{(m,j) \in V_{ml}} x_{mj}^n(t) \right)^{1/n} \right) \cdot I_{\left(x_{s,r} = \max_{(s,j) \in V_{sl}} x_{sj} \right)}(x_{s,r}) \right) \tag{8}$$

In the derivation of equation (8), suppose that there is only one virtual session having the highest rate among all the virtual sessions of a multicast session s through link l. Making this assumption is only to simplify the statement of results. In the case of more than one virtual session having the highest rate, a mark-distributing scheme can be used (to be presented later). The function $I_{\left(x_{s,r} = \max_{(s,j) \in V_{s,l}} x_{s,j} \right)}(x_{s,r})$ in equation (8) is an indicator function. Similar to the unicast case, constructing a Lyapunov function for the system described by the state equation (8) could prove that: the rates adapted according to equation (8) will converge to a unique stable point, which is also the unique optimal solution to problem (7).

From the rate control equation (8), obtain the heuristic rate control algorithm:

$$x_{s,r}(t_{k+1}) = \max \left(x_{\min}, x_{s,r}(t_k) + \delta \cdot \kappa_{s,r} \cdot \left(\omega_{s,r} - \frac{\beta}{U'_{s,r}(x_{s,r})} \sum_{l \in L_{s,r}} p_l(y) \cdot I_{\left(x_{s,r} = \max_{(s,j) \in V_{sl}} x_{s,j} \right)}(x_{s,r}) \right) \right) \tag{9}$$

Here t_k denotes the k^{th} rate update time. In a practical implementation of the algorithm, $x_{\min}$ is the minimum rate that every virtual session should be allocated. δ is the discretization step of the differential equation (8). Link l will mark the ECN bit in the header of each passing packet with the probability $p_l(y)$ in the case of congestion. The product $p_l(y) \cdot I_A(x_{s,r})$ has a simple interpretation as follows. When a virtual session (s, r) has a rate $x_{s,r}$ less than the highest rate among all virtual sessions of a multicast session s sharing link l, it does not contribute to the congestion on link l, so link l will not mark any packet of it, and the marking probability $p_l(y) \cdot I_A(x_{s,r})$ is equal to zero. For the virtual session having the highest rate, link l marks its packets with the probability $p_l(y) \cdot I_A(x_{s,r})$ equal to $p_l(y)$, which depends on the congestion status. The term $\sum_{l \in L_{s,r}} p_l(y) \cdot I_A(x_{s,r})$ in equation (9) thus can be viewed as the sum of the marking probabilities at all links along the route of a virtual session (s, r). Assuming the marking probability at each link is small, a receiver can use the fraction of marked packets received to estimate $\sum_{l\in L_{s,r}} p_l(y)\cdot I_A(x_{s,r})$, and then updates its rate according to the computation result of equation (9).

3.3 Algorithm implementation in IP networks

(1) Marking function.

Traditional RED notifies traffic sources congestion with implicit feedback signals, *i.e.*, time-out or duplicate ACK (acknowledgement). Referring to RED in conjunction with one-bit ECN, a link's marking approach that is appropriate for the use of equation (9) is proposed. The calculation of the marking probability includes the following two steps.

(a) Calculation of average queue-length at routers.

The internet is characterized by bursty flows. To filter out all the transient fluctuations and accurately capture the congestion, the weighted average of queue-length is used. The equation is

$$\mathrm{avg} = (1 - W_q)\cdot \mathrm{avg} + W_q \cdot q \tag{10}$$

Here q denotes the current queue-length, and avg denotes the average queue-length. W_q, $0<W_q<1$, is the weight of queue-length. Equation (10) is similar to a low-pass filter, and the weight W_q determines the filter's time constant.

(b) Linear marking probability.

As mentioned before, the marking probability $p_l(y)$ is required to be a non-decreasing function of y. Choose its form as:

$$p_l(y)=\begin{cases}0 & ,\mathrm{avg}\le \mathrm{TH}_{\min}\\ \dfrac{\mathrm{avg}-\mathrm{TH}_{\min}}{\mathrm{TH}_{\max}-\mathrm{TH}_{\min}} & ,\mathrm{TH}_{\min}<\mathrm{avg}<\mathrm{TH}_{\max}\\ 1 & ,\mathrm{avg}\ge \mathrm{TH}_{\max}\end{cases} \tag{11}$$

Here $\mathrm{TH}_{\min}$ is the threshold which the average queue length must exceed before any marking is done, and $\mathrm{TH}_{\max}$ the threshold which the average queue length must exceed before all packets are marked. From experience, let $\mathrm{TH}_{\min}$ be 25 percent of the practical used FIFO (First-In First-Out) queue, and $\mathrm{TH}_{\max}$ be 75 percent of it. See from equation (11), as a link becomes more heavily loaded, it incurs an increasing marking probability. This method assures that packets of each session are marked in proportion to the session's rate, and thus guarantees good inter-session fairness.

(2) Distributed implementation.

The distributed implementation of the rate control algorithm described earlier includes three parts.

(a) Link's algorithm.

First, link l checks the queue length. Next, calculate the marking probability p_l using equation (10) and equation (11). Finally, it examines the ECN bit of each passing packet. If this bit is 0, set it to 1 with the calculated probability p_l, otherwise, leave it as it is.

(b) Node's algorithm

Non-junction nodes simply forward each input packet to output link leading to its destination. But at junction nodes, each input packet of a multicast session needs to be reproduced to a number of copies, which are sent to different downstream receivers. So for any receiver whose rate is less than the highest, the junction node should clear the ECN bit of any copy of a marked packet before forwarding. In this way, only the receiver having the highest rate receives marked packets. If more than one downstream receiver has rate equal to the highest rate, randomly select one of them to send the marked packets.

(c) Receiver's algorithm.

Each receiver counts the number of packets received with the ECN bit set to 1 through a fixed duration, and estimates the sum of the marking probabilities at all links along its route by the fraction of marked packets. Then, request the optimal rate calculated according to equation (9).

4 Simulation and results

To verify the effectiveness of the rate control algorithm presented before, packet level simulations were conducted on a network with "Y" topology as shown in **figure 2**. Nodes R_1, R_2, R_3 and R_4 denote the routers in core network. R_1 and R_2 are connected by an 80Mbps link (Link1) with a one-way propagation delay 20 ms (this roughly corresponds to a distance of 4000 km). R_2 and R_3 are connected by a 60 Mbps link (Link2) with 20 ms one-way propagation delay. R_2 and R_4 are connected by a 40 Mb·ps link (Link3) with 10 ms delay. The rest nodes are all access nodes, and connected to the core network by access links of 1000Mb·ps bandwidth and 0.005 ms delay. Therefore Link1, Link2 and Link3 are bottleneck links. There are three classes of sessions in the network. Class 1 consists of 40 multicast sessions. Each session's source is connected to R_1, two heterogeneous receivers are connected to R_3 and R_4 respectively and multicast tree covers all the three bottleneck links. Class 2 has 40 unicast sessions with source connected to R_1 and receiver connected to R_3. Class 3 includes 40 unicast sessions with source connected to R_1 and receiver connected to R_4. So Link1, Link2 and Link3 are shared by 120, 80 and 80 sessions separately. The packet sizes are taken to be 1000 bytes long. So the capacity of Link1, Link2 and Link3 are equivalent to 10000, 7500 and 5000 packets per second respectively. Routers use output queueing, and the FIFO queue corresponding to each output link is limited to 100 packets.

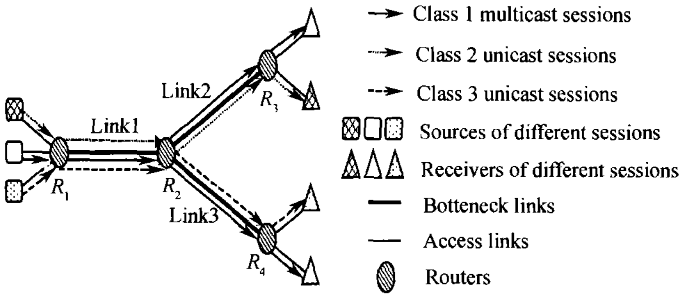

Figure 2 Simulation network topology.

4.1 Experiment 1: max-min fairness

All unicast sessions adopt the window based AIMD (Additive Increase Multipticative Decrease) congestion control mechanism. Since using ECN marks to replace NACKs (Negative Acknowledgement) and choosing appropriate marking function will result in negligible packet losses [9], in this simulations the effect of duplicate ACK or time-out isn't taken into account. Meanwhile, ignore slow-start and suppose that all TCP-based unicast sessions are in the congestion avoidance phase. That means every received ACK increases the current window size W_r by $^1\!/_{W_r}$, but if that ACK has a marked ECN bit carried back from the receiver the current window size will be halved. All multicast sessions use the rate based congestion control algorithm as given by equation (9). To assure resource allocation fairness, choose the parameters of algorithm equation (9) on the principle of approximating the TCP characteristics. So the utility function has the form $-^1\!/_{x_{s,r}}$, and the weight $\omega_{s,r}$, which makes the virtual session's rate linearly increase with time, is $^1\!/_{d_{s,r}^2}$. Here $d_{s,r}$ is the virtual session's RTT (Round Trip Time). The discretization step δ is 0.1, and the constant β is ln2. The rate update interval is 0.1s.

Randomly select a session as representative from each class to plot instantaneous throughput as shown in **figure 3**. It proves that the algorithm proposed guarantees unicast and multicast sessions compete with each other for network resources in a fair manner. Calculate the average throughput over a period of 100s, a multicast virtual session and a unicast session both passing through Link1 and Link2 are 78.78 and 79.86 packets per second respectively. This result shows that this algorithm is TCP-Friendly [13], and resource allocation nearly satisfies max-min fairness.

However, note that the average throughputs of a virtual session and a unicast session, which are both connected to *R*4, are 60.65 and 56.88 packets per second respectively. Two values are much different, and the former is higher. The reason for this is that the virtual session connected to *R*4 only contributes to the congestion on Link3 where its packets are marked, but packets of the unicast session traversing both Link1 and Link3 are marked at both. So the result is reasonable. Known from this, the fairness of resource allocation should be judged by comparing the throughputs of sessions that go through the same set of bottleneck links.

4.2 Experiment 2: proportional fairness

All the unicast and multicast sessions use the rate based congestion control algorithm as given by equation (9). The utility functions are chosen to be $\ln x_{sr}$ according to the definition of proportional fairness. Other parameters are same as in experiment 1. **Figure 4** shows the instantaneous throughputs of the same sessions as in experiment 1. Comparing figure 3 and 4, observe that the magnitude of throughput fluctuations is greater in the former case. The average throughputs of a virtual session and a unicast session both going through Link1 and Link2 are 88.98 and 89.41 packets per second respectively. For a unicast session traversing bottleneck Link1 and Link3, the average throughput is 60.65 packets per second. This value of a virtual session, which only affects the congestion on Link3, is 66.27 packets per second. These four values are all bigger than that obtained in experiment 1. Observed results show that resource allocation could be more efficient and resource competing could be more peaceful if all unicast and multicast sessions use the same rate control algorithm as equation (9) and choose the same utility function as $\ln x_{s,r}$.

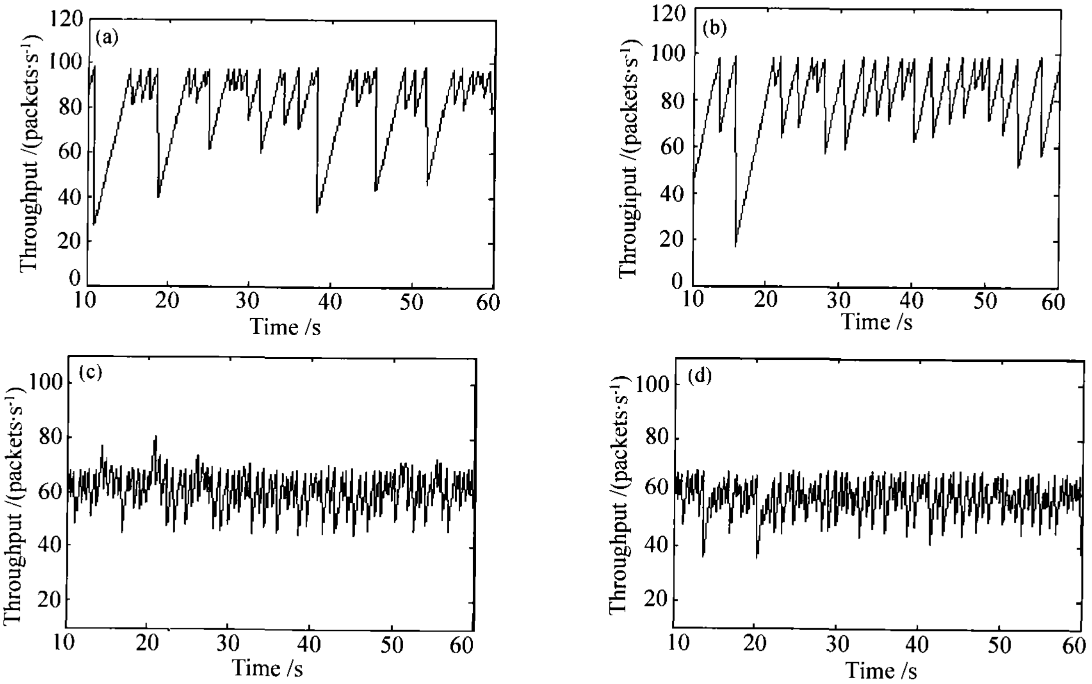

Figure 3 Instantaneous throughputs of different kinds of sessions in experiment 1, (a) a virtual session of class 1 connected to R_3; (b) a unicast session of class 2; (c) a virtual session of class 1 connected to R_4; (d) a unicast session of class 3.

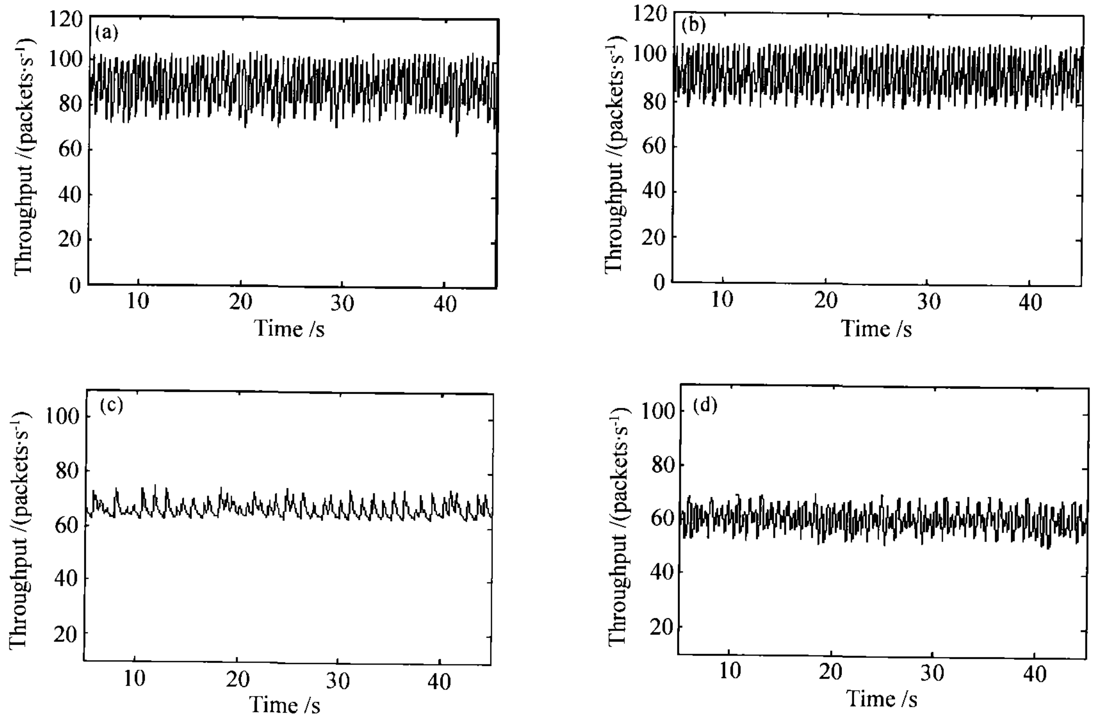

Figure 4 Instantaneous throughputs of different kinds of sessions in experiment 2.

5 Conclusions

In this paper, the characteristics of TCP unicast were analyzed at first. By regarding TCP as a special case of multicast, unicast and multicast were integrated into a unified nonlinear program formulation of the aggregate utility maximization problem. Based on some reasonable assumptions, finally a heuristic rate control algorithm was obtained by using a penalty function approach to solve the constraint problem. The algorithm is distributed. Network equipments (*i.e.*, routers and switches) mark the ECN bit of a packet with a probability that depends on the marking function proposed. End-hosts (*i.e.*, users) distill the congestion feedback information and update to the optimal rates calculated. The algorithm requires simple computations both for the users and routers. The overhead of information exchange is also very low. Using this algorithm, networks can deliver low-loss and low-delay service without the need for a centralized admission control, resource reservation or complicated scheduling mechanisms.

References

[1] F.P. Kelly, Charging and rate control for elastic traffic [J], *European Transactions on Telecommunications*, 8(1997), No.1, p.33.

[2] L. Massoulie and J. Roberts, Bandwidth Sharing, Objectives and algorithms, [in] *Proceedings of IEEE Infocom'99* [C], New York, 1999.

[3] D. Rubenstein, J. Kurose, and D. Towstey, The impact of multicast layering on network fairness, [in] *Proceedings of ACM Sigcomm'99[C]*, Cambridge, 1999.

[4] S. Sarkar and L. Tassiulas, Fair allocation of utilities in multirate multicast networks, [in] *Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing* [C], Allerton House, USA, 1999.

[5] S. Sarkar and L. Tassiulas, Distributed algorithms for computation of fair rates in multirate multicast trees, [in] *Proceedings of IEEE Infocom 2000* [C], Tel Aviv, Israel, 2000.

[6] K. Kar and S. Sarkar, Optimization based rate control for multirate multicast sessions, [in] *Proceedings of IEEE Infocom 2001* [C], Anchorage, USA, 2001.

[7] S. Deb and R. Srikant, Convergence of distributed congestion control schemes in networks with multicast and unicast sessions, [in] *Proceedings of Conference on Information Sciences and Systems 2001* [C], Baltimore, USA, 2001.

[8] S. Deb and R. Srikant, Congestion control for fair resource allocation in networks with multicast flows, [in] *Proceedings of Conference on Decision and Control 2001* [C], Orlando, USA, 2001.

[9] S. Kunniyur and R. Srikant, A time scale decomposition approach to adaptive ECN marking, [in] *Proceedings of IEEE Infocom 2001* [C], Anchorage, USA, 2001.

[10] S. McCanne, V. Jacobson, and M. Vetterli, Evaluation of hierarchical rate control, [in] *Proceedings of ACM Sigcomm'96* [C], California, USA, 1996.

[11] F.P. Kelly, A.K. Maulloo, and D.K.H. Tan, Rate control in communication Networks: shadow prices, proportional fairness and stability [J], *Journal of Operations Research Society*, 49(1998), No.3, p.237.

[12] S. Kunniyur and R. Srikant, End-to-end congestion control schemes: utility functions, random losses and ECN marks, [in] *Proceedings of IEEE Infocom 2000* [C], Tel Aviv, Israel, 2000.

[13] S. Floyd and K. Fall, Promoting the use of end-to-end congestion control in the internet [J], *IEEE/ACM Transactions on Networking*, 7(1999), No.4, p.458.