Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Communications 29 (2006) 1458–1469
computer communications
www.elsevier.com/locate/comcom

Study on nominee selection for multicast congestion control

Feng Xie, Gang Feng *, Chee Kheong Siew

School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore

Received 10 September 2004; received in revised form 25 August 2005; accepted 16 September 2005
Available online 14 October 2005

Abstract

Nominee selection plays a key role in nominee-based congestion control, which is essential for multicast services to ensure fairness and congestion avoidance. Without valid design of nominee selection mechanism, the design of congestion control protocols could be inefficient or even flawed. Existing nominee selection schemes choose nominees by comparing the calculated throughput of receivers using the TCP throughput equation with the measured loss rate and round-trip time. Since the calculated throughput varies with different transmission rates, it may not accurately indicate the eligibility of a receiver to be the nominee. This causes the problem that a new nominee is not necessarily 'worse' than the current one and the 'worst' receiver could not be selected accurately. In this paper, we study the nominee selection principles and mechanisms. First, we address the problem in existing schemes by identifying the conditions for the valid use of calculated throughput. Next, we propose a new general nominee selection algorithm (GNSA) as a solution and prove that GNSA converges to the 'worst' receiver and the expected number of iterations is less than $(1 + \ln n)$, where n is the group size. Finally, we demonstrate through ns-2 simulations the benefits of GNSA in terms of better fairness properties and less iterations to converge than existing nominee selection schemes such as that in TFMCC.

Keywords: Congestion control; Multicast; Nominee selection; Fairness; Convergence

1. Introduction

Congestion control is a key issue for the deployment of multicast services. Extensive research work has been conducted to achieve scalable, efficient, and fair multicast congestion control. Despite various congestion control schemes proposed to address this issue, whether it is safe to deploy IP multicast service is still an open question [1–4]. Currently, it is still a hot topic to develop scalable, TCP-friendly, and robust congestion control schemes for multicast.

Existing multicast congestion control schemes fall into two main categories: single-rate and multirate schemes. Single-rate schemes differ from multirate schemes in that they employ a universal transmission rate for transmission to all receivers while multirate schemes employ multiple transmission rates to different receivers. Among the existing schemes, multirate multicast congestion control schemes present good scalability for various environments especially with heterogeneous characteristics. Most of these schemes employ the layered approach to enable transmission at different rates adapted to receivers' receiving capabilities [1,2]. However, multirate schemes suffer from the complexity of mechanisms and are exposed to the doubt of safety. One of the major challenges for multirate schemes is caused by the correlation among receivers. Multirate congestion control provides receivers with the opportunity of determining their receive rate. As a result, the receiver-driven nature of multirate approaches can be challenged by the self-beneficial behaviors of receivers [3]. The solution to this problem, on the other hand, may lead to a more complex scheme.

In comparison, single-rate multicast congestion control takes advantages of its clear logic and structure [4–7]. Generally, the single-rate multicast congestion control scheme regulates the transmission rate, which is identical for all receivers, to the most stringent congestion control requirement in the multicast group. Accordingly, the 'worst' receiver is selected as the nominee and this congestion control method is also known as nominee-based congestion control. In this scheme, the 'worst' receiver or so-called nominee determines the upper bound of the transmission rate. Therefore, the nominee selection mechanism is essential for nominee-based congestion control as it determines in some degree the multicast performance in terms of multicast throughput, resource utilization, and fairness.

* Corresponding author. Tel.: +65 67904383; fax: +65 67922971.
E-mail address: egfeng@ntu.edu.sg (G. Feng).

0140-3664/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2005.09.006

The nominee-based congestion control scheme is composed of two parts, a unicast congestion control mechanism for the nominee as the representative of receivers and the nominee selection mechanism. Though much work has been done on nominee-based congestion control, not much attention is paid to the nominee selection mechanism. In fact, wrong acker selection has been observed in the simulation scenarios shown by Seada et al. [8]. As a result, the protocols may violate the fairness criteria and thus may not be TCP-friendly. However, the scenarios they present are some particular cases within specific protocols. What we are interested in is investigating the fundamental topics of nominee selection: the design objectives of nominee selection mechanisms and what kind of nominee selection mechanisms is needed to achieve those goals.

Nominee-based congestion control mechanism requires the nominee selection mechanism to be accurate and scalable. Given these design goals, nominee selection may fail under the following circumstances: (1) the nominee selection procedure converges to a wrong 'worst' receiver; (2) the convergence of the nominee selection procedure is too slow for large group size. Since the nominee selection mechanism is an essential component of nominee-based congestion control, the design of congestion control protocols could be inefficient or even flawed without a valid design of the nominee selection mechanism. For example, suppose the nominee selection procedure converges to a wrong 'worst' receiver. Even if the unicast congestion control mechanism for the nominee is TCP-friendly, the receivers with worse network condition than the nominee, especially the real 'worst' receiver, may experience significant packet loss and/or excessive transmission delay. Consequently, the multicast transmission could be unfair to other TCP-compatible flows competing for the bottleneck resources. More seriously, due to the failure of 'worst' receiver selection, the sustained traffic overload at the bottleneck links could even break the whole system by causing congestion collapse [9].

Among existing schemes of nominee-based congestion control, TFMCC [4,5] and pgmcc [6,7] are known to have good performance. They both use the TCP throughput equation to select the nominee by comparing the calculated throughput as a function of the loss rate and round-trip time experienced by receivers. As the calculated throughput varies with different transmission rates, a higher calculated throughput of a receiver does not necessarily indicate a higher actual throughput when it is selected as the nominee. In other words, the calculated throughput may not indicate the eligibility of a receiver to be the nominee. Therefore, we argue that the nominee selection mechanism based on the calculated throughput need to be further investigated. Moreover, whether this method is scalable to large group size is yet to be justified before the experimental deployment.

In this paper, we focus on the nominee selection principles and mechanisms for multicast congestion control. We first formulate the problem by formally defining the nominal throughput and estimated nominal throughput. Nominal throughput is the actual throughput when the receiver is the nominee but estimated nominal throughput is the calculated throughput by the TCP throughput equation with the measured loss rate and round-trip time of each receiver. As the nominal throughput represents the eligibility of a receiver to be the nominee, we can justify the use of estimated nominal throughput if and only if the comparison results of receivers' estimated nominal throughput and nominal throughput are equal. Following this motivation, we identify the condition under which valid comparisons of estimated nominal throughput for nominee selection can be made. Based on the findings, we propose a new general nominee selection algorithm (GNSA) for which every switch of nominee is justified to select a receiver 'worse' than the current one. We prove that GNSA converges to the 'worst' receiver and the expected number of iterations is less than $(1+\ln n)$, where n is the group size.

We also use ns-2 simulations to illustrate one of the potential threats of deploying the existing nominee selection schemes in current Internet: the multicast flows may share bottleneck bandwidth much more aggressively than TCP flows. The unfairness to TCP flows is caused by the inaccurate selection of the 'worst' receiver. This conclusion is also validated by simulations. In contrast, applying the proposed nominee selection algorithm can improve the fairness properties obviously. We also demonstrate through simulations that GNSA needs less iteration to converge than existing nominee selection schemes such as that in TFMCC.

The rest of this paper is structured as follows. In Section 2, we introduce some related work on nominee selection mechanisms for multicast congestion control. In Section 3, we revisit the logic of nominee-based congestion control and study the principles of nominee selection. We propose a new general nominee selection algorithm (GNSA) and prove its convergence to the 'worst' receiver in Section 4. Next, in Section 5 we illustrate and discuss the fairness properties of the nominee selection mechanisms. In Section 6, we evaluate and compare the convergence speed of GNSA and TFMCC. Finally, Section 7 concludes the paper by highlighting some directions for future work.

2. Previous research on nominee selection mechanisms

Nominee-based congestion control schemes encompass most of the single-rate multicast congestion control schemes. Multicast congestion control schemes may employ various nominee selection mechanisms. In spite of the various names of the nominee, most of the nominee selection mechanisms are equation-based. In detail, the nominee is selected by comparing the values calculated by the TCP throughput equation with the loss rate and round-trip time of the flows between the sender and receivers. Thus, nominee-based congestion control shares some common features with equation-based congestion control. We will first revisit some prime proposition of equation-based congestion control briefly.

Equation-based congestion control is developed based on the long-term TCP throughput equations [9–11]:

$$B(p,T) = \frac{s}{T\sqrt{\frac{2p}{3}}}, \tag{1}$$

$$B(p,T) = \frac{s}{T\sqrt{\frac{2p}{3}} + 4T\left(3\sqrt{\frac{3p}{8}}\right)p(1+32p^2)}. \tag{2}$$

These two equations express TCP throughput as a function of loss event rate p, round-trip time T, and packet size s. Eq. (1) is derived from a simple model [9] and (2) from a more complex model [11] approximating TCP throughput. Hence we call them simple TCP throughput equation and complex TCP throughput equation, respectively. Whetten and Conlan [12] use a reliable multicast (RM) throughput equation modified from the complex TCP throughput Eq. (2) by adding a scaling factor Q to estimate the TCP-compatible throughput for multicast:

$$B(p,T) = \frac{s}{TQ^{-1}\sqrt{\frac{2p}{3}} + 4T\left(3\sqrt{\frac{3p}{8}}\right)p(1+32p^2)}, \tag{3}$$

where Q is a scaling factor for adjusting the responsiveness of the scheme. If $Q=1$, the above equation reduces to (2) and the scheme presents same responsiveness as TCP. Schemes using equations with $Q<1$ present higher responsiveness than TCP and vice versa for $Q>1$. Fig. 1 depicts the calculated throughput using different equations with $T=100$ ms and $s=$ 1, 000 Bytes. Tsimple and Tcomplex represent the curves of simple and complex TCP throughput equations, respectively. TRM curves indicate the throughput as a function of loss rate using (3). We can observe that Tcomplex decreases faster than Tsimple with loss rate. The difference between Tsimple and Tcomplex grows significantly after loss rate increases to 0.01.

Based on the above TCP throughput equations, various TCP-friendly congestion control schemes are proposed for both unicast and multicast services [13]. Among the proposed congestion control schemes, some of unicast schemes can be extended to multicast schemes. For example, Widmer and Handley extend a unicast scheme, TCP-friendly rate control protocol (TFRC) [14,15], to TCP-friendly multicast congestion control (TFMCC) [4,5]. Accordingly, unicast schemes have been studied as the prototypes of multicast schemes on their transient and long-term behaviors [16–18]. Motivated by the success of extending unicast schemes to single-rate multicast congestion control, some multirate multicast congestion control schemes using similar ideas are also proposed [2,19,20]. Such extensions and experimental deployments help us to understand the complexity of multicast congestion control schemes. The findings during the investigation can also benefit the design of single-rate congestion control schemes.

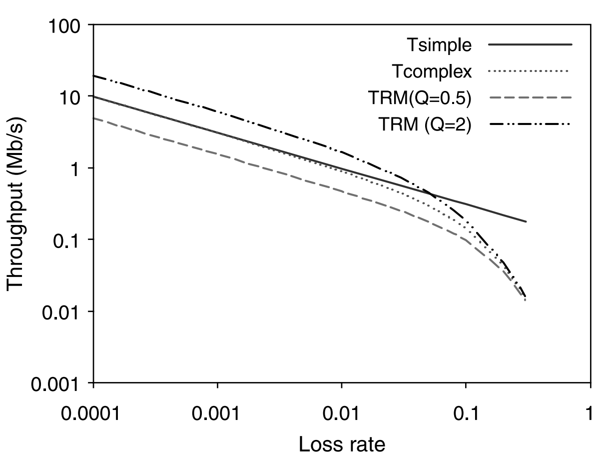

Fig. 1. Comparison of different equations ($T=0.1$ s).

After a brief introduction on nominee-based congestion control schemes, we now examine the nominee selection mechanisms they use. Most of the existing nominee selection mechanisms are based on TCP throughput equations. To have a clear overview, we discuss nominee selection mechanisms according to the categories of nominee-based congestion control schemes: rate-based and window-based.

The rate control mechanism proposed by DeLucia and Obraczka [21] selects a fixed number of representatives from group members to send feedback for dynamically adjusting the transmission rate. The sender determines the representatives based on the congestion indication (CI) and congestion clear (CC) feedback from receivers. For each group's largest round-trip time (GRTT), only one new representative is allowed to join the representative set to avoid unnecessary oscillations. Shi and Waldvogel propose to identify the worst receiver in the multicast group which is defined as the receiver behind the link with lowest bandwidth capacity [22]. The bandwidth capacity is calculated by an equation similar to simple TCP throughput Eq. (1) to achieve TCP-friendly throughput on the path to the worst receiver. Yamamoto et al. also adopt the simple TCP throughput equation to select the representative for congestion control [23]. Whetten and Conlan [12] use (3), which is modified from complex TCP throughput equation by adding the scaling factor Q, to estimate the TCP-compatible throughput for multicast. They also compare the double worst and worst path methods used to select the values of T and p for calculating the RM throughput. In TFMCC [4,5], Widmer and Handley extend TFRC [14,15], an equation-based congestion control scheme for unicast based on (2), to multicast services. In TFMCC, the sending rate is determined by the current limiting receiver (CLR), the receiver with the lowest expected throughput calculated by the complex TCP throughput equation. A feedback suppression method is proposed to suppress receivers' feedback unless their calculated throughput is less than 90% of the suppression rate notified from other receivers.

Similar to the nominee selection mechanisms in rate-based congestion control schemes, equation-based nominee selection mechanisms are also used for window-based schemes. In active error recovery/nominee congestion avoidance (AER/NCA) [24], a receiver behind the most bandwidth-constrained path is selected as the nominee according to the simple TCP throughput equation. The sender emulates TCP behavior by maintaining a single congestion window adjusted by loss indications from the nominee. In pgmcc, Rizzo et al. propose the so-called acker election/tracking mechanism to select the group representative, or acker [6,7]. However, the nominee selection mechanism of pgmcc is similar to that of AER/NCA. The innovative aspect of pgmcc is that it explicitly employs a mechanism to avoid unnecessary acker switches: a new acker is accepted only if its calculated throughput is less than α times of the calculated throughput of current acker. Values of α are recommended to vary between 0.6 and 0.8.

Although most of the nominee selection mechanisms are equation-based, different mechanisms for nominee selection are also possible. Rhee et al. propose a unicast congestion control scheme TCP emulation at receivers (TEAR) which takes advantages of both rate- and window-based schemes [25]. At the receiver side, TEAR employs the TCP emulation mechanism to maintain a congestion window for each receiver. The window is adjusted independently by congestion signals received by the receiver, which enables each receiver to estimate a TCP-friendly receiving rate. The value of the estimated receiving rate is sent to the sender periodically to determine the transmission rate. Simulations in [25] show that TEAR presents comparable or better performance than TFRC [15] in terms of fairness, TCP-friendliness, rate fluctuation reduction, and stability with high feedback latency. With proper mechanisms such as the scalable feedback mechanism in TFMCC [5], TEAR can be extended to multicast congestion control with expected good performance. For the multicast extension of TEAR, the receiver with the lowest estimation value of a TCP-friendly receiving rate can be selected as the nominee. This may offer a different nominee selection mechanism other than the equation-based approaches.

The existing work of nominee-based congestion control has made substantial efforts on the framework of protocols as well as some specifications of nominee selection such as scalable measurement of RTT [5]. Currently, TFMCC as a representative of rate-based schemes and pgmcc as a representative of window-based schemes are both substantially mature. Both of them use the calculated throughput of receivers to select the nominee. However, no deep work has been conducted to address the processes and principles of the nominee selection mechanism. Most importantly, what is the foundation of using the calculated throughput instead of the actual throughput of receivers for nominee selection? Does the nominee selection mechanism scale to a large group size? How nominee selection mechanisms affect the performance of single-rate multicast congestion control schemes? Our study will address these questions thoroughly and provide solid foundations for nominee selection.

3. Nominee selection for multicast congestion control

3.1. Overview of nominee selection

The objective of the nominee selection mechanism is selecting the 'worst' receiver. Some researchers define the 'worst' receiver as the most congested receiver or the receiver behind the link with the lowest bandwidth capacity [22,24]. However, these definitions as well as those in other existing work are not clear and measurable enough for nominee selection. For this purpose, we define a new concept: nominal throughput. Nominal throughput of a specified receiver is defined to be the average throughput when the sender assigns that receiver as the nominee. As TCP-friendliness is a requirement for deploying multicast services, the sender maintains a TCP-friendly congestion control mechanism to manage the flow between the sender and the nominee. Based on (1), the nominal throughput of a receiver can be expressed as

$$r_j(p,T) = \frac{c}{T_j\sqrt{p_j}}, \tag{4}$$

where $c = s\sqrt{3/2}$ and T_j and p_j are the round-trip time and loss rate of receiver j when it is assigned as the nominee. As long as the unicast congestion control between the sender and nominee satisfies the general congestion control requirements[1], regulating the multicast transmission rate to the lowest nominal throughput will exactly satisfy the congestion control requirements for all multicast flows between the sender and all receivers. Thus, the receiver with the lowest nominal throughput is defined as the 'worst' receiver. Let C be the set of all receivers for a specific multicast group and let the nominal throughput of receiver j be denoted as r_j and the 'worst' receiver be receiver μ, the multicast throughput r could be

$$r = r_\mu = \min_j r_j, \quad \forall j \in C, \tag{5}$$

where $\min_j r_j$ means selecting the minimal value of r_j as the group throughput and the corresponding receiver j as the nominee. Notice that

$$r \le r_\mu = \min_j r_j, \quad \forall j \in C, \tag{6}$$

is the rate constraint for all single-rate multicast congestion control schemes.

Existing nominee-based congestion control schemes such as TFMCC [5] and pgmcc [7] adopt an equation-based nominee selection mechanism motivated by (4), the empirical equation of nominal throughput. We can estimate the nominal throughput for a receiver according to its experienced round-trip time T_j and loss rate p_j. The calculated throughput is thus named estimated nominal throughput. Denoted by A_j the estimated nominal throughput of receiver j, we have

$$A_j(p,T) \triangleq \frac{c}{T_j\sqrt{p_j}}, \tag{7}$$

where T_j and p_j are the round-trip time and loss rate of receiver j with a fixed nominee. Note that

$$r_j = A_j \tag{8}$$

when receiver j is the nominee.

Eq. (7) provides a way to estimate the nominal throughput of receivers. However, estimated nominal throughput is an equation-based value subject to conditions such as

[1] TCP-friendliness and congestion avoidance are deemed as the general congestion control requirements.

Table 1
Variables used for nominee selection mechanisms

R_i	Receiver i
r	Throughput of a multicast group
r_i	Nominal throughput of receiver i
r_μ	Nominal throughput of the 'worst' receiver
A_i	Estimated nominal throughput of receiver i
p_i	Loss rate experienced by receiver i
T_i	Round-trip time between the sender and receiver i

the transmission rate. For example, the loss rate and/or the round-trip time both increase with the transmission rate, which results in a decrease of estimated nominal throughput. Thus, we can expect that estimated nominal throughput is not necessarily a good estimation of nominal throughput. In the following part, we will discuss the principles of nominee selection in detail. For the ease of discussions, we list in Table 1 all notations used for nominee selection mechanisms in the rest part of this paper.

3.2. Principles of nominee selection

In the following analysis, we first consider the case that the network status (with background traffic) is stable[2]. In this way, the comparison result between two receivers' nominal throughput r_i and r_j is fixed. According to (7), estimated nominal throughput is a function of loss rate and round-trip time. As loss rate p_j and round-trip time T_j both change with the average transmission rate, the estimated nominal throughput A_j can be considered as a function of the average transmission rate r. Under the condition that the network status is stable, the average transmission rate r_j for a fixed nominee j is approximately invariable in the long term. Eq. (8) hence results in

$$r_j = A_j(r_j). \qquad (9)$$

With stable background traffic, the increase of the average transmission rate r leads to the increase of the aggregate traffic load on the multicast paths. As a result, the average queuing delay at the trespassing nodes will increase and the packet drop rate due to buffer overflow will either increase or keep unchanged, which ultimately leads to the increase of round-trip time T_j and/or the increase of loss rate p_j. In another word, both the loss rate p_j and the round-trip time T_j are nondecreasing functions of the average transmission rate r. From (7), the estimated nominal throughput A_j is thus a nonincreasing function of the average transmission rate r, i.e.,

$$A_j(r) \geq A_j(r') \quad \text{for} \quad r < r'. \qquad (10)$$

In fact, (9) and (10) are the only two expressions necessary for deriving the propositions in this paper and thus replacing

[2] This is an ideal condition for nominee selection. We use it to illustrate the principles and procedures of nominee selection. In fact, nominee selection mechanism is a statistical algorithm in that the 'worst' receiver could only be statistically selected in real network conditions. In short, the stable network condition is not a requirement for the effectiveness of nominee selection mechanisms.

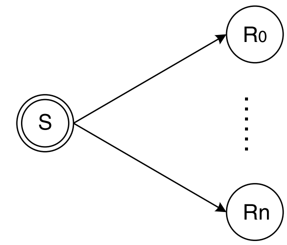

Fig. 2. A star topology for the experiment.

the simple TCP throughput equation with the complex one does not affect the following analysis.

Before going to the propositions, let us make a deep investigation on the characteristics of estimated nominal throughput. Due to the change of A_j with different average transmission rates, we can observe that the comparison results of estimated nominal throughput of two receivers can be different with different average transmission rates. We use an experiment of a multicast session based on a star topology to illustrate this phenomenon. As shown in Fig. 2, there are a number of receivers R_i ($i=0,1,\ldots,n$) in the multicast group. For the ease of presentation, the properties of each link are presented in a triple (bandwidth, delay, loss rate). Table 2 lists the link properties in ns-2 [26] simulations. We implement TFMCC [27] in our simulations and make the following modifications for our experiment. All receivers report their estimated nominal throughput to the sender. The sender selects R_0–R_9 in turn as the nominee by design. Hence, we can observe the change of estimated nominal throughput of receivers with the periodical switching of nominees. Without loss of generality, we choose to compare the estimated nominal throughput of R_3 and R_9 for 0–100 s (Fig. 3). We can observe that the result of comparing A_3 and A_9 changes with the time. For example, during the time around 10 s, $A_3 < A_9$. However, during the time around 60 s, $A_3 > A_9$. This phenomenon challenges the existing nominee selection schemes which directly use estimated nominal throughput for nominee selection.

The observation in the above example demonstrates that the estimated nominal throughput may not be a good estimation of nominal throughput. Recall that the nominal throughput represents the eligibility of a receiver to be the nominee. Hence, whether the comparisons of the estimated nominal throughput of receivers are valid for selecting the 'worst' receiver depends on the relationship between comparing the nominal throughput and estimated nominal throughput of receivers. In the rest of this section, we will try to find

Table 2
Link properties

Links	Properties	Links	Properties
to R0	(1 M, 20 ms, 0.005)	to R5	(1.1 M, 20 ms, 0.01)
to R1	(1 M, 40 ms, 0.005)	to R6	(1.1 M, 40 ms, 0.01)
to R2	(1 M, 60 ms, 0.005)	to R7	(1.1 M, 60 ms, 0.01)
to R3	(1 M, 80 ms, 0.005)	to R8	(1.1 M, 80 ms, 0.01)
to R4	(1 M, 100 ms, 0.005)	to R9	(1.1 M, 100 ms, 0.01)

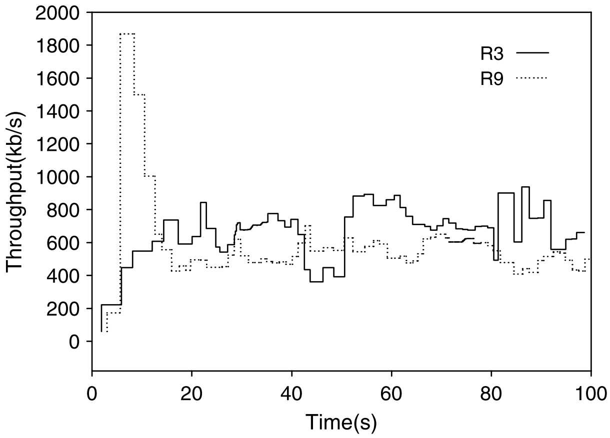

Fig. 3. Comparison of estimated nominal throughput.

the condition under which valid comparisons of estimated nominal throughput for nominee selection can be made.

Proposition 1. For any two receivers i and j, $A_i(r_j) < A_j(r_j) \Leftrightarrow r_i < r_j \Leftrightarrow A_i(r_i) < A_j(r_i)$.

Proof: If $r_i < r_j$, from (9) and (10) we have $A_i(r_j) \leq A_i(r_i) = r_i < r_j = A_j(r_j) \leq A_j(r_i)$. Thus, we have $A_i(r_j) < A_j(r_j)$ and $A_i(r_i) < A_j(r_i)$. Next, we show that $A_i(r_j) < A_j(r_j)$ leads to $r_i < r_j$. $A_i(r_j) < A_j(r_j)$ states that receiver i experiences worse network status with higher loss rate and/or higher round-trip time than the current nominee j. Thus, the average transmission rate will decrease when the nominee changes from receiver j to receiver i. After the average transmission rate is stabilized, the average transmission rate r_i meets $r_i < r_j$. Finally, $A_i(r_i) < A_j(r_i)$ also results in $r_i < r_j$. $A_i(r_i) < A_j(r_i)$ tells that receiver j experiences better network status with lower loss rate and/or lower round-trip time than the current nominee i. Thus, the average transmission rate will increase when the nominee changes from receiver i to receiver j. After the average transmission rate is stabilized, the average transmission rate r_j meets $r_i < r_j$.

Proposition 2. For any two receivers i and j, $A_i(r_j) = A_j(r_j) \Leftrightarrow r_i = r_j \Leftrightarrow A_i(r_i) = A_j(r_i)$.

Proof: If $r_i = r_j$, from (9) we have $A_i(r_j) = A_i(r_i) = A_j(r_j) = A_j(r_i)$. If $r_i \neq r_j$, then $r_i < r_j$ or $r_i > r_j$. Combining with Proposition 1 results in $A_i(r_j) < A_j(r_j)$ and $A_i(r_i) < A_j(r_i)$ or $A_i(r_j) > A_j(r_j)$ and $A_i(r_i) > A_j(r_i)$, i.e. $A_i(r_j) \neq A_j(r_j)$ and $A_i(r_i) \neq A_j(r_i)$. Hence, $A_i(r_j) = A_j(r_j) \Leftrightarrow r_i = r_j \Leftrightarrow A_i(r_i) = A_j(r_i)$.

Proposition 3. For any two receivers i and j, if there exists nominee e satisfying $A_i(r_e) < A_e(r_e) \leq A_j(r_e)$, then $A_i(r_i) < A_j(r_i)$.

Proof: From $A_i(r_e) < A_e(r_e) \leq A_j(r_e)$ and Propositions 1 and 2, we have $r_i < r_e \leq r_j$. $r_i < r_j$ leads to $A_i(r_i) < A_j(r_i)$ according to Proposition 1.

Proposition 1 can be illustrated by Fig. 4. The arrowed line $A(r) = r$ in the graph indicates that the estimated nominal throughput is equal to the current transmission rate. At the intersection of curve $A_i(r)$ and line $A(r) = r$, receiver i is the current nominee. Comparing receivers i and j results in $A_i(r_i) < A_j(r_i)$. Similarly, $A_i(r_j) < A_j(r_j)$ when receiver j is the nominee. Thus, the nominee selection procedure by comparing

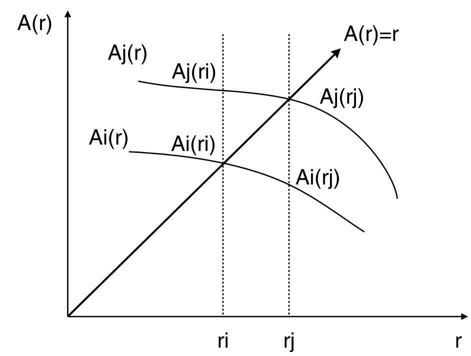

Fig. 4. Illustration of Proposition 1.

the estimated nominal throughput of the two receivers will converge to receiver i no matter which receiver is the initial nominee for the nominee selection procedure. Consequently, $A_i(r_j) < A_j(r_j) \Leftrightarrow A_i(r_i) < A_j(r_i)$ can be considered as the premise of using estimated nominal throughput for nominee selection.

To verify Proposition 1 by simulations, we again use the previous experiment with the topology of Fig. 2. Fig. 5 shows the estimated nominal throughput of R_3 and R_9. R_3 is the nominee during the period between 28.5 and 35.0 and R_9 is the nominee during the period between 70.6 and 77.7. In addition to the estimated nominal throughput of R_3 and R_9, we also plot the throughput of the multicast group. When a receiver is the nominee, the multicast throughput should be equal to the nominal throughput of the nominee. Referring to the figure, $r_3 = A_3(r_3)$ for the left interval when R_3 is the nominee and $r_9 = A_9(r_9)$ for the right interval when R_9 is the nominee. We can observe that $A_3(r_3) > A_9(r_3)$, $A_3(r_9) > A_9(r_9)$, and $r_3 > r_9$. These results exactly match Proposition 1.

Propositions 1–3 illustrate the relationship between comparing the nominal throughput and estimated nominal throughput of receivers. Accordingly, the comparisons of the estimated nominal throughput of receivers are valid for deciding whether a receiver is ‘worse’ than another one under the conditions shown by the propositions. In other words, the estimated nominal throughput used for comparison should be measured when the transmission rate is adjusted according to the status of either one of the two receivers. Otherwise, a new nominee is

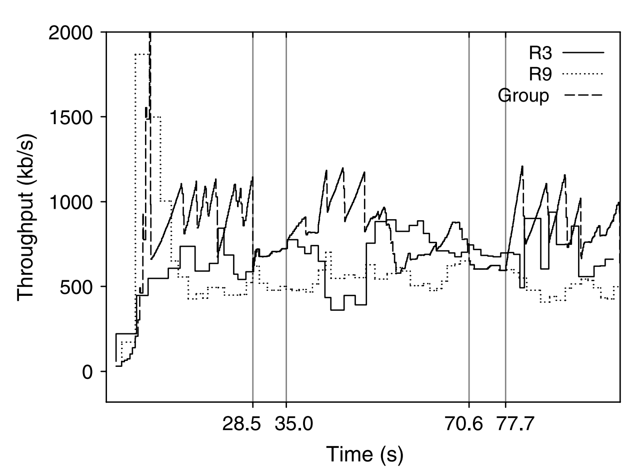

Fig. 5. Experiment for Proposition 1.

not necessarily 'worse' than the current one, which is a problem in existing nominee selection schemes. This conclusion and the three propositions are called the principles of nominee selection. We will show in the next section how these nominee selection principles can be used to design a new algorithm which converges to the 'worst' receiver gracefully.

4. A new general nominee selection algorithm and its convergence

The principles of nominee selection provide a way to use estimated nominal throughput for nominee selection so that every switch of nominee is intended to select a receiver 'worse' than the current nominee. For example, suppose R_3 is the current nominee among $\{R_1, R_2, R_3, R_4\}$ and $A_1(r_3) < A_2(r_3) < A_3(r_3) < A_4(r_3)$. By applying Proposition 1, we can obtain $r_1 < r_3$, $r_2 < r_3$ and $r_3 < r_4$. Notice that a common mistake is to conclude $r_1 < r_2$ from $A_1(r_3) < A_2(r_3)$. In fact, R_1 and R_2 both are eligible receivers for the 'worst' receiver. So we need another comparison between $A_1(r_1)$ and $A_2(r_1)$ or between $A_1(r_2)$ and $A_2(r_2)$. In fact, the principles of nominee selection suggest that, by comparing the estimated nominal throughput, we can select a receiver 'worse' than the current nominee for each iteration, instead of the 'worst' one by just one iteration. This example also shows that the convergence of nominee selection procedure needs a number of iterations. The average number of iterations necessary for the convergence of nominee selection determines whether using the estimated nominal throughput for nominee selection scales to large group size. In this section, we will address this question based on a new nominee selection algorithm proposed.

Based on Propositions 1–3, we propose a new general nominee selection algorithm (GNSA) shown in Fig. 6. The algorithm is supposed to tackle two requirements of nominee selection: nominee selection procedure should converge to the 'worst' receiver and the nominee selection mechanism should scale to large group size. For this purpose, every switch of nominee in the algorithm is intended to find a receiver 'worse' than the current nominee. This is the most important difference between the proposed algorithm and the existing nominee selection schemes.

The proposed nominee selection algorithm works in the following way. Line 1 gives the initial condition. C^0 is the set of eligible receivers, also named initial candidate set, from which we select the nominee for iteration step 0. C^0 is initialized to include all receivers R_i. The cardinality of C^0 is equal to the group size n. k is the iteration index. For each iteration, the sender randomly selects a receiver from the candidate set C^k as the nominee of the kth iteration e^k. Let the multicast throughput at the kth iteration be equal to the nominal throughput of nominee e^k. After some time for measuring the estimated nominal throughput of receivers at the kth iteration, the sender will find the receivers whose estimated nominal throughput A_i^k is smaller than the estimated nominal throughput of current nominee $A_{e^k}^k$ and update the candidate set with these receivers. Next, we increase the iteration index k by one and advance to the next iteration. Repeating the iterations until the cardinality of C^k is equal to 0, the latest nominee e^{k-1} is the 'worst' receiver.

The new algorithm differs from the existing nominee selection schemes in some important aspects. The new algorithm guarantees that every switch of nominee is destined to find a receiver with lower nominal throughput than the current nominee while existing schemes cannot. For this purpose, the estimated nominal throughput of receivers compared to that of the nominee is measured when the transmission rate is adjusted according to the status of the same nominee. In the existing nominee selection schemes such as TFMCC [5] and pgmcc [7], there are no such mechanisms and hence a new nominee is not necessarily 'worse' than the current one. Thus, we can expect that the new algorithm may present better fairness properties and need less iteration to converge than existing nominee selection schemes.

1. $C^0 = \{R_i\}$; $|C^0| = n$; $k = 0$
2. while $|C^k| > 0$
3. randomly select $e^k \in C^k$ as nominee; $r^k = r_{e^k}$
4. $C^{k+1} = \{R_i : A_i^k < A_{e^k}^k\}$
5. k++
6. end while
7. current nominee e^{k-1} is the "worst" receiver

Fig. 6. A new algorithm for nominee selection: the general nominee selection algorithm (GNSA).

Theorem 1. The algorithm of Fig. 6 converges to the 'worst' receiver in finite iterations. The expected number of iterations is less than $(1+\ln n)$, where n is the group size.

Proof: The multicast group is composed of n receivers. C^k is the set of eligible receivers from which we select the nominee for the kth iteration. For each iteration, we randomly choose a receiver e^k from the candidate set C^k as the nominee. After comparing $A_{e^k}^k$ with all other A_i^k $(\forall i, i \neq e^k)$, the receivers may fall into three subsets: $A_i^k > A_{e^k}^k$, $A_i^k = A_{e^k}^k$, or $A_i^k < A_{e^k}^k$. From Proposition 1, receivers with $A_i^k > A_{e^k}^k$ could not be the 'worst' receiver since $r_i > r_{e^k}$. From Proposition 2, receivers with $A_i^k = A_{e^k}^k$ are not 'worse' than the current nominee since $r_i = r_{e^k}$. Hence, the 'worst' receiver should be selected from the receivers with $A_i^k < A_{e^k}^k$ since $r_i < r_{e^k}$. As a result, the candidate set of next iteration C^{k+1} is $\{R_i : A_i^k < A_{e^k}^k\}$, from which we select the next nominee e^{k+1}. From Proposition 3, if a receiver j with A_j^k meets $A_j^k \geq A_{e^k}^k > A_{e^{k+1}}^k$, then $A_j^{k+1} > A_{e^{k+1}}^{k+1}$. It states that $C^{k+2} \subset C^{k+1}$. Thus, $C^{k+1} = \{R_i : A_i^k < A_{e^k}^k\}$ in the algorithm is equal to $C^{k+1} = \{R_i \in C^k : A_i^k < A_{e^k}^k\}$. Therefore, at least one receiver (nominee e^k) will be removed from the candidate set for each iteration. As a result, the algorithm in Fig. 6 converges to the 'worst' receiver in finite iterations no more than group size n.

Next, we will calculate the expected number of iterations to select the 'worst' receiver. We first consider the case that no two receivers have approximately equal nominal throughput, i.e.,

$$r_i \neq r_j (\forall i, j, i \neq j). \quad (11)$$

Denote by $f(n)$ the expected number of iterations to select the 'worst' receiver from a group with n receivers. The calculation in the Appendix shows that

$$f(n) = \sum_{k=1}^{n} \frac{1}{k}, \quad (12)$$

which is known as the harmonic numbers denoted by H_n [28].

Now we consider the case that more than one receiver have approximately equal nominal throughput. For receivers with approximately equal nominal throughput, any one of the receivers can be used as the 'worst' receiver in this subset to represent all of them. No matter which one of the receivers in the subset is selected as the nominee, all receivers in the subset can be excluded from the candidate set for the next iteration, making the convergence procedure faster. Thus, the expected number of iterations $f(n)$ will be smaller than H_n. In summary, the expected number of iterations to select the 'worst' receiver from a group with n receivers is less than or equal to H_n, i.e.

$$f(n) \leq H_n. \quad (13)$$

It has been proved in [28] that

$$\ln n < H_n < 1 + \ln n. \quad (14)$$

In consequence, the expected number of iterations to select the 'worst' receiver is less than $(1+\ln n)$.

The above analysis is conducted for the static network environment. In this case, the nominal throughput of receivers does not change and GNSA converges to the 'worst' receiver. Now we consider the dynamic variation of network status and its influence on nominee selection mechanism. In real network conditions, the nominal throughput of receivers may not be fixed due to the change of network status with background traffic. As a result, the 'worst' receiver may change from time to time but it can be considered as a moving target that the nominee selection algorithm tries to approach. Since the background traffic is an aggregation of traffic from many sources, the change of network status is typically slower than the dynamics of individual sessions [29]. Although the 'worst' receiver may not be selected exactly, with a sufficient convergence speed provided by GNSA, the nominee selection mechanism should be able to keep up with the pace of network changes and follow the 'worst' receiver closely.

Theorem 1 demonstrates that nominee selection by GNSA scales to large group size well. The expected number of iterations increases logarithmically with the group size. In the implementations, we can use an alternative nominee selection rule

$$C^{k+1} = \{R_i : A_i^k < \alpha \bullet A_{e^k}^k\}, \quad 0 < \alpha \leq 1, \quad (15)$$

for better flexibility, where α is a scaling factor for adjusting the stability of nominee selection. We recommend to select α in the range of (0.9 1.0). The setting can suppress unnecessary nominee switches due to measurement fluctuations of estimated nominal throughput without significantly compromising the accuracy of nominee selection. For related work on measuring loss rate and round-trip time to calculate the estimated nominal throughput, please refer to the work in TFMCC [4,5].

5. Illustrations and discussions of fairness properties of nominee selection mechanisms

In this section, we illustrate the risks of nominee selection by using the estimated nominal throughput in the existing nominee selection schemes: the multicast flows may share bandwidth much more aggressively than TCP flows. We also examine the fairness properties of modified congestion control schemes with the proposed nominee selection algorithm. The comparisons on the fairness properties of the existing schemes and the proposed algorithm will provide deep insights for the design of nominee selection mechanisms in multicast congestion control.

We use the topology shown in Fig. 7 to simulate scenarios for examining fairness properties. There are two bottleneck links, L1 and L2, in the topology. The multicast flows compete with one TCP flow, TCP1, on Link L1 and also with two TCP flows, TCP2 and TCP3, on Link L2. Each TCP session consists of a TCP sender (TS) and a TCP receiver (TR). For example, TS1 sends data to TR1 for TCP1. The paths connecting the multicast sender, MS, and its two receivers, MR1 and MR2, have identical link properties except the bandwidth (1 Mb/s, 1.1 Mb/s) on the bottleneck links L1 and L2, respectively. With the configurations, r_1 will be 36% higher than r_2.

For the simulations, TCP flows start transmission much earlier than the multicast sender so that the TCP flows reach their steady states when the multicast transmission starts at time 0. We again implement TFMCC [27] in our simulations. Fig. 8 shows the evolution of the TFMCC flow and the three TCP flows sharing the two bottleneck links L1 and L2. The slope of each curve indicates the bandwidth sharing of the corresponding flow. We can observe that TFMCC shares much more bandwidth than TCP2 and TCP3 on the link L2. During the last 50 s, TFMCC shares 40% of the available bandwidth on L2, 33% higher than the bandwidth sharing of TCP2 or TCP3. Recall that r_1 is just 36% higher than r_2. This significant unfairness to TCP is caused by the failure of nominee selection. We can observe in the upper part of Fig. 9 that the nominee is

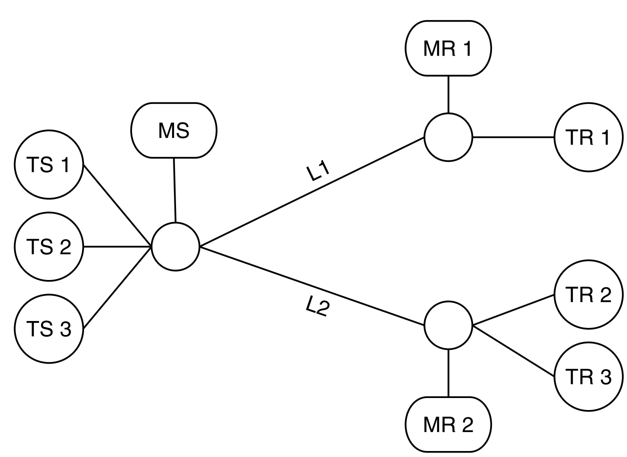

Fig. 7. Topology for examining fairness properties.

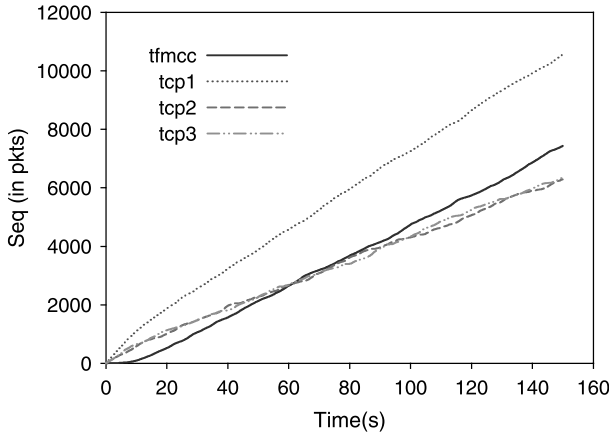

Fig. 8. Bandwidth sharing—TFMCC.

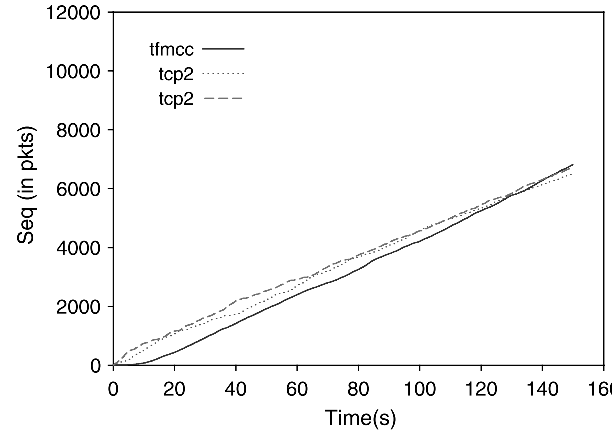

Fig. 10. Bandwidth sharing with fixed nominee—TFMCC.

switching between the two receivers MR1 and MR2 although MR2 is the real 'worst' receiver. This is the problem that nominee selection does not converge to the 'worst' receiver, one of the cases that nominee selection fails we discussed in Section 1. Fig. 9 also shows the estimated nominal throughput of the two receivers and how the comparison results between A_1 and A_2 determine the nominee selection. During the start phase of TFMCC, $A_1 < A_2$ as expected. After about 11 s, $A_1 > A_2$ and the real 'worst' receiver MR2 is selected as the nominee for the first time. After that, the estimated nominal throughput of both receivers fluctuates and the nominee switches frequently. The observation clearly demonstrates the risks of nominee selection by the estimated nominal throughput in the existing nominee selection schemes.

To clarify the doubt that the unfairness of TFMCC is not caused by the nominee selection mechanism but the unicast congestion control scheme for the nominee, we conduct the simulation again without MR1. Thus, MR2 is always selected as the nominee and we can obtain the results shown in Fig. 10. We can observe that TFMCC with fixed nominee selection shares bandwidth almost fairly with TCP flows. This experiment justifies the statement that the failure of nominee selection may lead to the failure of achieving the goals of multicast congestion control.

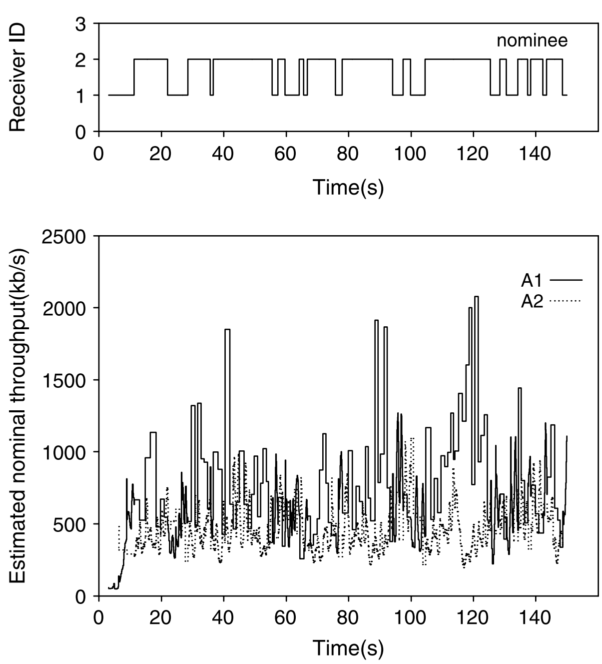

Fig. 9. Behaviors of nominee selection—TFMCC.

Next, we examine the fairness properties of the proposed nominee selection algorithm. For the simulations, we apply the new general nominee selection algorithm (GNSA) to TFMCC platform to replace the nominee selection scheme in TFMCC with the proposed algorithm for two considerations. First, the nominee-based congestion control scheme TFMCC is based on a TCP-friendly equation-based congestion control scheme TFRC which has been studied substantially [5,15]. Second, TFMCC is a mature scheme with scalable mechanisms for round-trip time measurement and feedback suppression. These two features make TFMCC a good platform of nominee-based congestion control. For the algorithm, we use the more flexible nominee selection rule $C^{k+1} = \{R_i : A_i^k < \alpha \bullet A_{c^k}^k\}$ mentioned in Section 4. The scaling factor α is set to 0.95 to avoid unnecessary nominee switches without significantly compromising the accuracy of nominee selection. The new implementation is denoted as GNSA.

Fig. 11 shows the evolution of the GNSA flow and the three TCP flows competing for the bottleneck resources on links L1

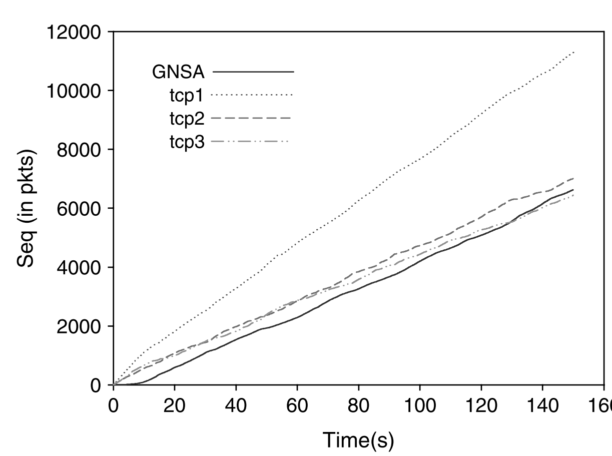

Fig. 11. Bandwidth sharing—GNSA.

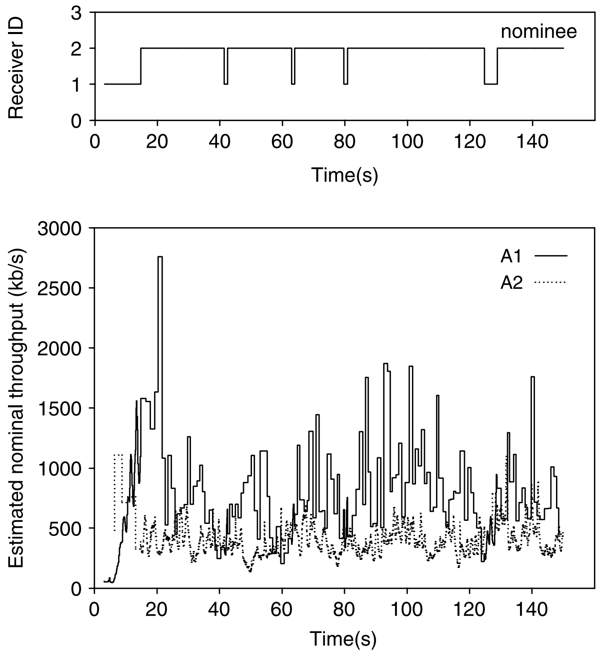

Fig. 12. Behaviors of nominee selection—GNSA.

and L2. We can observe that GNSA shares the 'worst' link L2 fairly with the two competing TCP flows TCP2 and TCP3. This is due to the accurate nominee selection of GNSA (Fig. 12). The estimated nominal throughput of MR1 and MR2 also fluctuates heavily, making the nominee switches possible. The fluctuations come from the loss rate and/or round-trip time measurement process and thus makes it necessary in the future work to add some smoothing techniques to the measurement mechanism. However, compared with the nominee selection procedure of TFMCC in Fig. 9, it is obvious that the fluctuations of estimated nominal throughput in GNSA are less violent and $A_2 \ll A_1$ during most of the time. Thus, it is reasonable to observe that GNSA selects the 'worst' receiver more accurately than TFMCC. This experiment shows that GNSA is more TCP-friendly than TFMCC due to the improvement of more accurate nominee selection.

6. Evaluation of convergence speed of nominee selection mechanisms

In this section, we use ns-2 [26] simulations to evaluate the convergence speed of GNSA and the existing schemes such as TFMCC. The simulation results can be also used to validate Theorem 1 and examine if the GNSA needs less iterations to converge to the 'worst' receiver than the existing schemes such as TFMCC. Before discussing the details, we look into a phenomenon shown in the previous section which may affect the convergence of nominee selection.

In Section 5, we observe that the estimated nominal throughput of receivers fluctuates heavily. As a result, the nominee may switch occasionally in GNSA since $A_i > A_j$ and $A_i < A_j$ appear intermittently (Fig. 12). We can expect that the phenomenon can be more likely to happen for receivers with approximately equal nominal throughput. This observation

Table 3
Link properties in the experiment network

Link type	Bandwidth (Mb/s)	Delay (ms)	Packet loss prob.
Transit–transit	500–1 024	1–5	0
Transit–stub	100–500	10–20	10^{-6}–10^{-4}
Stub–stub	1–10	10–50	10^{-4}–5×10^{-3}

affects the convergence speed of nominee selection mechanisms. We have shown in Section 4 that the expected number of iterations for the implementation algorithm of GNSA to converge will be less than H_n. However, considering the inaccuracy of measurement, we have observed in the previous simulations that $A_i > A_j$ and $A_i < A_j$ may appear intermittently for the receivers with approximately equal nominal throughput. As a result, an old nominee with reduced estimated nominal throughput may fall into the group $C^{k+1} = \{R_i : A_i^k < \alpha \cdot A_{e^k}^k\}$, making it possible to be selected as the nominee again. This will slightly increase the expected number of iterations. Therefore, the expected number of iterations for convergence may exceed H_n. This is highly probable to happen if α is equal or very close to 1.

Next, we use ns-2 [26] simulations to evaluate the convergence speed of nominee selection mechanisms. Since the convergence of nominee selection needs a number of iterations, some receivers, which are not the 'worse' receiver, can be selected as the nominees. They are named transient nominees. The number of transient nominees (NTN) is an indication of the convergence speed of the nominee selection procedure. We will use the average number of transient nominees (ANTN) as the measurement.

In our simulations, we use Georgia Tech Internetwork Topology Model (GT-ITM) [30,31] to randomly generate transit-stub topologies. The number of nodes in transit-stub topologies ranges from 20–1, 000 nodes. The bandwidth, delay and packet loss rate on the links are uniformly distributed within the ranges shown in Table 3. The values are set for generating heterogeneous network topologies. Fig. 13 depicts

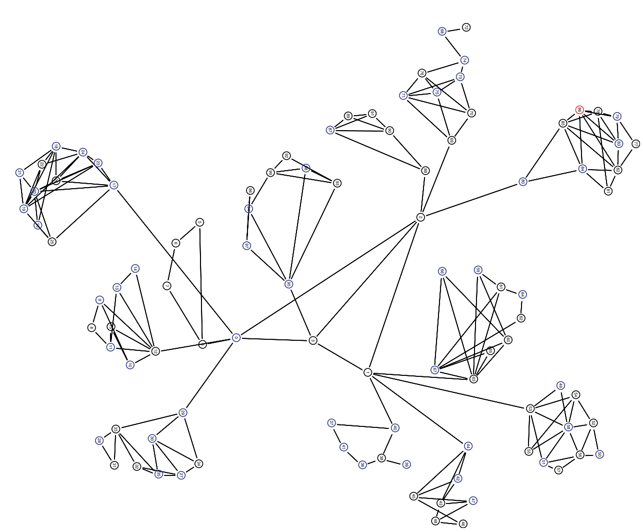

Fig. 13. Sample topology generated by GT-ITM.

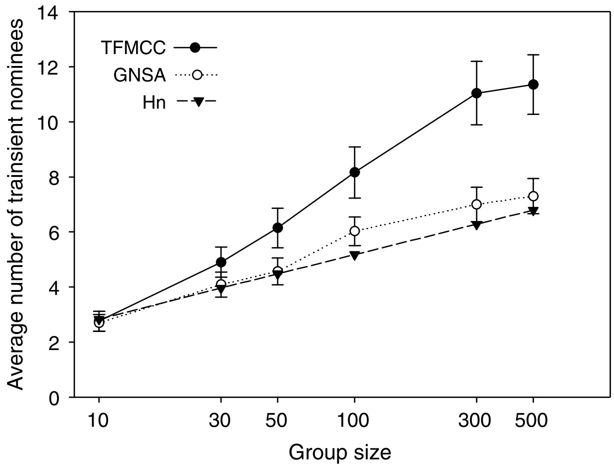

Fig. 14. Convergence of nominee selection mechanisms.

an example of a random topology with 100 nodes generated with GT-ITM for the experiments. For each topology, a number of senders and receivers, also named agents for both, are placed at some of the nodes for data transmission. The ratio of the number of agents and the number of nodes is called the deployment level [32]. For the experiments, multicast transmission agents are deployed randomly with a deployment level of 0.5. Referring to the range of topology size, the multicast group size (including the sender) ranges from 10 to 500.

For our simulations, we compare GNSA to the original TFMCC to show the difference of their convergence speed. To allow the feasible comparison between different nominee selection mechanisms, our simulations always use the same topologies, agent deployments and initial conditions to ensure that the comparison is carried out in absolutely identical scenarios.

Fig. 14 shows the average number of transient nominees (ANTN) of GNSA. We can observe that the curve of GNSA approximately matches the curve of H_n and ANTN increases logarithmically with the group size. The observations roughly support Theorem 1. For large group size, the expected number of iterations for convergence slightly exceeds H_n. This can be explained by the previous analysis on the phenomenon that $A_i > A_j$ and $A_i < A_j$ appear intermittently for the receivers with approximately equal nominal throughput. Accordingly, an old nominee with reduced estimated nominal throughput may fall into the group $C^{k+1} = \{R_i : A_i^k < \alpha \cdot A_{e^k}^k\}$, making it possible to be selected as the nominee again. This will slightly increase the expected number of iterations.

As a comparison, ANTN of TFMCC is also shown. For TFMCC, ANTN also increases logarithmically with the group size, similar to the behavior of GNSA. However, TFMCC needs more iterations in average to converge to the 'worst' receiver. This is because that TFMCC does not guarantee that a new nominee is 'worse' than the current one. This incurs more unnecessary nominee switches.

7. Conclusions and future work

Nominee selection plays a key role in nominee-based congestion control, which is essential for multicast services to ensure fairness and congestion avoidance. Existing nominee selection schemes choose nominees by comparing the calculated throughput of receivers using the TCP throughput equation. Since the calculated throughput varies with different transmission rates, it may not accurately indicate the eligibility of a receiver to be the nominee. This causes the problem that a new nominee is not necessarily 'worse' than the current one and the 'worst' receiver could not be selected accurately. As a result, the multicast service using the nominee-based congestion control mechanism may behave more aggressively than TCP and thus violates the TCP-friendly requirement.

In this paper, we have studied the nominee selection principles and mechanisms. First, we addressed the problem in existing schemes by identifying the conditions for the safe usage of calculated throughput. Next, we presented a new general nominee selection algorithm (GNSA) as a solution and proved that GNSA converges to the 'worst' receiver and the expected number of iterations is less than $(1+\ln n)$, where n is the group size. Finally, we demonstrated through ns-2 simulations the benefits of GNSA in terms of better fairness properties and less iteration to converge than existing nominee selection schemes such as that in TFMCC.

Another contribution of this paper is to draw attentions to investigating the nominee selection mechanisms. The simulations in Section 5 clearly show that nominee-based congestion control without appropriate nominee selection mechanisms may lead to severe consequences like fairness violations. In fact, the nominee selection mechanism is an essential component of nominee-based congestion control and should be carefully designed in the future work. The design goals may include safety, accuracy, scalability, stability, and responsiveness. At the same time, appropriate performance measures should be identified to evaluate the performance of the nominee selection schemes. Finally, it is also an important issue to study the influence on nominee-based congestion control of different nominee selection schemes.

Appendix. Calculating the expected number of iterations

In this section, we calculate the expected number of iterations for GNSA to converge to the 'worst' receiver. For each iteration, we randomly select a receiver e^k from candidate set C^k. At the end of this iteration, only receivers with $A_i^k < A_{e^k}^k$ remains in the candidate set of C^{k+1}. Denote by $f(n)$ the expected number of iterations to select the 'worst' receiver from a group with n receivers. We have supposed that no two receivers have approximately equal nominal throughput. After one iteration, we have

$$f(n) = \frac{1}{n}[f(0) + f(1) + \cdots + f(n-2) + f(n-1)] + 1. \quad \text{(A.1)}$$

Similarly we have

$$f(n-1) = \frac{1}{n-1}[f(0) + f(1) + \cdots + f(n-2)] + 1. \quad (A.2)$$

By denoting

$$S \triangleq f(0) + f(1) + \cdots + f(n-2), \quad (A.3)$$

we can obtain

$$f(n-1) = \frac{1}{n-1} \cdot S + 1, \quad (A.4)$$

or

$$S = (n-1) \cdot [f(n-1) - 1]. \quad (A.5)$$

Substituting (.5) and (.3) into (.1) we have

$$f(n) = \frac{1}{n}\{(n-1) \cdot [f(n-1) - 1] + f(n-1)\} + 1. \quad (A.6)$$

Rearranging the equation we can obtain

$$f(n) = f(n-1) + \frac{1}{n}. \quad (A.7)$$

As $f(0)=0$,

$$f(n) = \sum_{k=1}^{n} \frac{1}{k}, \quad (A.8)$$

which is known as the harmonic numbers denoted by H_n [28].

References

[1] J.W. Byers, G. Horn, M. Luby, M. Mitzenmacher, W. Shaver, FLID-DL: congestion control for layered multicast, IEEE Journal on Selected Areas in Communications 20 (8) (2002) 1558–1570.

[2] G.-I. Kwon, J. Byers, Smooth multirate multicast congestion control, in: Proceedings of the INFOCOM'03, San Francisco, CA, 2003.

[3] S. Gorinsky, S. Jain, H. Vin, Y. Zhang, Robustness to inflated subscription in multicast congestion control, in: Proceedings of the ACM SIGCOMM'03, Karlsruhe, Germany, 2003.

[4] J. Widmer, M. Handley, TCP-friendly multicast congestion control (TFMCC): Protocol specification, Internet Draft draft-ietf-rmt-bb-tfmcc-02.txt, a work in progress, Jul. 2003.

[5] J. Widmer, M. Handley, Extending equation-based congestion control to multicast applications, in: Proceedings of the ACM SIGCOMM'01, San Diego, California, United States, 2001.

[6] L. Rizzo, G. Iannaccone, L. Vicisano, M. Handley, PGMCC single rate multicast congestion control: Protocol specification, Internet Draft draft-ietf-rmt-bb-pgmcc-02.txt, a work in progress, Jun. 2003.

[7] L. Rizzo, Pgmcc: a TCP-friendly single-rate multicast congestion control scheme, in: Proc. ACM SIGCOMM'00, Stockholm, Sweden, 2000.

[8] A.H.K. Seada, S. Gupta, Systematic evaluation of multicast congestion control protocols, in: SCS International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), San Diego, USA, 2002, full version to appear in JSAC 4th Quarter 2004.

[9] S. Floyd, K. Fall, Promoting the use of end-to-end congestion control in the internet, IEEE/ACM Transactions on Networking 7 (4) (1999) 458–472.

[10] M. Mathis, J. Semke, J. Mahdavi, The macroscopic behavior of the TCP congestion avoidance algorithm, ACM SIGCOMM Computer Communication Review 27 (3) (1997) 67–82.

[11] J. Padhye, V. Firoiu, D.F. Towsley, J.F. Kurose, Modeling TCP Reno performance: a simple model and its empirical validation, IEEE/ACM Transactions on Networking 8 (2) (2000) 133–145.

[12] B. Whetten, J. Conlan, A rate based congestion control scheme for reliable multicast, Technical white paper, GlobalCast Communication, Oct. 1998.

[13] J. Widmer, R. Denda, M. Mauve, A survey on TCP-friendly congestion control, IEEE Network Magazine 15 (3) (2001) 28–37.

[14] M. Handley, S. Floyd, J. Padhye, J. Widmer, TCP friendly rate control (TFRC): Protocol specification, RFC 3448, Jan. 2003.

[15] S. Floyd, M. Handley, J. Padhye, J. Widmer, Equation-based congestion control for unicast applications, in: Proc. ACM SIGCOMM'00, Stockholm, Sweden, 2000.

[16] Y.R. Yang, N.S. Kim, S.S. Lam, Transient behaviors of TCP-friendly congestion control protocols, in: Proceedings of the INFOCOM'01, Anchorage, Alaska, 2001.

[17] M. Vojnović, J.-Y. Le Boudec, On the long-run behavior of equation-based rate control, in: Proceedings of the ACM SIGCOMM'02, Pittsburgh, Pennsylvania, USA, 2002.

[18] D. Bansal, H. Balakrishnan, S. Floyd, S. Shenker, Dynamic behavior of slowly-responsive congestion control algorithms, in: Proceedings of the ACM SIGCOMM'01, San Diego, California, United States, 2001.

[19] M. Luby, V.K. Goyal, Wave and equation based rate control building block, Internet Draft draft-ietf-rmt-bb-webrc-04.txt, a work in progress, Dec. 2002.

[20] M. Luby, V.K. Goyal, S. Skaria, G.B. Horn, Wave and equation based rate control using multicast round trip time, in: Proceedings of the ACM SIGCOMM'02, Pittsburgh, Pennsylvania, USA, 2002.

[21] D. DeLucia, K. Obraczka, A multicast congestion control mechanism for reliable multicast, in: Proceeding of Third IEEE Symposium on Computers and Communicaiton (ISCC'98), Athens, Greece, 1998.

[22] S. Shi, M. Waldvogel, A rate-based end-to-end multicast congestion control protocol, in: Proceedings of Fifth IEEE Symposium on Computers and Communication (ISCC'00), Antibes, France, 2000.

[23] K. Yamamoto, M. Yamamoto, H. Ikeda, Congestion control for reliable multicast achieving TCP fairness, IEICE Transactions on Communications E85-B (1) (2002) 183–190.

[24] S. Kasera, S. Bhattacharyya, M. Keaton, D. Kiwior, S. Zabele, J. Kurose, D. Towsley, Scalable fair reliable multicast using active services, IEEE Network 14 (1) (2000) 48–57.

[25] I. Rhee, V. Ozdemir, Y. Yi, TEAR: TCP emulation at receivers—flow control for multimedia streaming, Tech. Rep., NCSU, Apr. 2000.

[26] ns-Network Simulator, URL http://www.isi.edu/nsnam/ns/.

[27] TFMCC Implementation, URL http://www.informatik.uni-mannheim.de/informatik/pi4/projects/CongCtrl/tfmcc/.

[28] R.A. Brualdi, Introductory Combinatorics, third ed., Prentice-Hall, Englewood Cliffs, NJ, USA, 1999.

[29] D. Bertsekas, D.P. Bertsekas, Data Networks, second ed., Prentice-Hall, Englewood Cliffs, NJ, USA, 1991.

[30] GT-ITM: Georgia Tech Internetwork Topology Models, URL http://www.cc.gatech.edu/projects/gtitm/.

[31] E. Zegura, K. Calvert, S. Bhattacharjee, How to model an internetwork, in: Proceedings of the INFOCOM'96, San Francisco, CA, 1996.

[32] C. Papadopoulos, E. Laliotis, Incremental deployment of a router-assisted reliable multicast scheme, in: Proceedings of NGC 2000 on Networked Group Communication, Palo Alto, California, United States, 2000.