

# Priority Service Provisioning and Max–Min Fairness: A Utility-Based Flow Control Approach

Jiong Jin<sup>1</sup>  · Marimuthu Palaniswami<sup>2</sup> ·  
Dong Yuan<sup>3</sup> · Yu-Ning Dong<sup>4</sup> · Klaus Moessner<sup>5</sup>

Received: 11 January 2016 / Revised: 22 September 2016 / Accepted: 20 October 2016 /  
Published online: 25 October 2016  
© Springer Science+Business Media New York 2016

**Abstract** In this paper, a novel priority assignment scheme is proposed for priority service networks, in which each link sets its own priority threshold, namely, the lowest priority the link is willing to support for the incoming packets without causing any congestion. Aiming at a reliable transmission, the source then assigns each originated packet the maximum priority value required along its path, because links may otherwise discard the incoming packets which do not meet the corresponding priority requirements. It is shown that if each source sends the traffic at a rate that is reciprocal to the specified highest priority, a bandwidth max–min fairness is achieved in the network. Furthermore, if each source possesses a utility

---

✉ Jiong Jin  
jjiongjin@swin.edu.au  
Marimuthu Palaniswami  
palani@unimelb.edu.au  
Dong Yuan  
dong.yuan@sydney.edu.au  
Yu-Ning Dong  
dongyn@njupt.edu.cn  
Klaus Moessner  
k.moessner@surrey.ac.uk

- <sup>1</sup> School of Software and Electrical Engineering, Swinburne University of Technology, Melbourne, VIC 3122, Australia
- <sup>2</sup> Department of Electrical and Electronic Engineering, The University of Melbourne, Melbourne, VIC 3010, Australia
- <sup>3</sup> School of Electrical and Information Engineering, The University of Sydney, Sydney, NSW 2008, Australia
- <sup>4</sup> College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
- <sup>5</sup> Institute for Communication Systems, University of Surrey, Guildford GU2 7XH, UK

function of the available bandwidth and sends the traffic at a rate so that the associated utility is reciprocal to the highest link priority, a utility max–min fairness is achieved. For general networks without priority services, the resulting flow control strategy can be treated as a unified framework to achieve either bandwidth max–min fairness or utility max–min fairness through link pricing policy. More importantly, the utility function herein is only assumed to be strictly increasing and does not need to satisfy the strictly concave condition, the new algorithms are thus not only suitable for the traditional data applications with elastic traffic, but are also capable of handling real-time applications in the Future Internet.

**Keywords** Priority assignment · Congestion control · Utility-fair resource allocation · Quality of service · Real-time application

## 1 Introduction

Today’s “best effort” Internet has become a great success in providing efficient data transmission services, e.g., electronic mail and web browsing, but it is not sufficient to support the increasing demand for real-time services, such as audio, video and multimedia delivery through the network. These real-time applications usually have a strict quality of service (QoS) requirement, and are sensitive to time delay and bandwidth allocated, which are generally not easy to be guaranteed in the current TCP-based Internet methodology.

To provide a more reliable transmission in the Internet, one proposed means is to classify Internet traffic into priority classes and transmit prioritized packets using IETF adopted differentiated services (diffserv) technology [1]. In this approach, applications (users) with strict QoS requirements are intuitively assigned with a higher priority and therefore receive a better and faster service than the lower priority classes during their transmission in the network. To ensure priority services work properly, there must be a mechanism to determine the service requirements of individual applications and assign different traffic to the appropriate priority classes. Otherwise, each user may declare the highest priority for their own benefit, and the above priority scheme would degenerate to an inefficient “best effort” service. Meanwhile, an efficient flow control scheme is required to set the individual traffic rate, preventing network congestion and packet loss, as well as achieving maximal throughput and fair resource allocation among all different network users.

In order to better manage the traffic in communication networks than the current TCP does, an extensive study has been carried out in the last decade. One of the most successful results in the congestion control and resource allocation area is the “optimal flow control” (OFC) approach initially proposed by Kelly et al. [2]. In his seminal paper, network flow control problem is for the first time formulated as an optimization problem and an explicit rate flow control algorithm is derived by solving that optimization problem via link pricing policy. This pioneer work was further advanced by the researches for nearly all types of networks including wired networks [3–5], wireless cellular networks [6–8], wireless ad hoc and sensor networks [9–12].

Though different authors may use different formulations and optimization methods, the approaches are essentially the same in the literature. The main idea of OFC is, for each network application (user), there is an associated utility function of the transmitting rate that can be used as a measurement of application's QoS performance over the available bandwidth. The design objective is to maximize the total QoS utilities of all users under the link capacity constraints of the network. An OFC algorithm is then derived by solving the optimization problem distributively, which usually consists of a link algorithm to measure the congestion (link price) in the network and a source algorithm to adapt the transmission rate according to congestion feedback signals.

By selecting utility as a logarithmic function, Kelly et al. [2] shows that the OFC approach achieves (in equilibrium) a *proportional fairness* for bandwidth allocation. Using the OFC strategy, another important fairness criterion called *max–min fair allocation* [13] (which emphasizes equal sharing compared with *proportional fairness*) is first studied by Mo and Walrand [14]. In their work, the authors use a family of utility functions to arbitrarily closely approximate max–min fair allocation. But the chosen utility function eventually becomes *ill-conditioned* when the max–min fairness is reached, and the associated link prices at congested links either turn to 0 or diverge to  $\infty$ . Their max–min fair flow control algorithms are hence impractical from an engineering point of view. Furthermore, in order to cope with different users with different QoS requirements, a new criterion named *utility max–min fairness* is initially proposed in [15] and later on comes various bandwidth allocation algorithms [16, 17]. Particularly in [15], the links require the information of utility functions from all the traversed sources, which makes network implementation difficult. On the other hand, Marbach [18, 19] studies the resource allocation problem in the priority service networks, where the users are free to choose the priority for their traffic but pay the charge of priority services to the network. Based on a static link pricing model via packet loss probability, Marbach shows that there exists an equilibrium when all the users pursue their own maximal net benefit and such equilibrium is (weighted) max–min fair for bandwidth allocations. However, he does not propose an efficient max–min fair flow control algorithm in his papers. Before the OFC approach was introduced, max–min fair bandwidth allocation is achieved either by means of global information update in the network [13], or approximated by a packet scheduling process [20]; nevertheless, the network congestion problem is not well considered.

The OFC approach not only provides an efficient and fair bandwidth allocation among competing users, but also is used to analyze the existing congestion control protocols and uncover the underlying working rationale of TCP. Within the OFC framework, we are able to see that various TCP protocols are merely different algorithms to solve the same optimization problem with different source utility functions [21]. It is further shown that, in fact, we could map the link price and feedback notification mechanism to a physical model. If each link  $l$  generates congestion notification randomly (either through packet dropping or ECN marking) at a mean rate  $n_l(t)$  at time  $t$ , the rate of end-to-end congestion notification is

$$N(t) = 1 - \prod_i (1 - n_i(t)).$$

Assuming  $n_i(t) \ll 1$ , we get approximately

$$N(t) \approx \sum_i n_i(t),$$

which exactly corresponds to the path price calculation rule (summation of the traversing link prices) in the OFC.

Despite great advances in optimal flow control theory and applications, there still exist serious limitations as indicated by our previous study [22, 23].

- The OFC approach is only suitable for *elastic* traffic attaining a strictly increasing and concave utility function, which ensures the feasible optimal solution and convergence of utility maximization process. It cannot deal with congestion control and resource allocation for communication networks where real-time applications are involved.
- In the utility maximization approach, if each user selects different utility function based on the real QoS requirement, the OFC approach usually leads to an extremely unfair resource allocation for the practical use. The applications with lower demand are usually allocated with higher bandwidth, and vice versa. This can be clearly shown by the counter example below.

**A Counter Example.** Consider a single link of capacity  $c$  shared by two sources. Source 1 attains a utility of  $\log(x_1 + 1)$  and source 2 attains a utility of  $2\log(x_2 + 1)$ , where  $x_1$  and  $x_2$  are source rates, respectively. With an easy calculation, the following bandwidth allocation is established by the utility maximization based OFC approach.

$$x_1 = \begin{cases} 0 & \text{if } c \leq 1, \\ \frac{c-1}{3} & \text{if } c > 1 \end{cases} \quad (1)$$

$$x_2 = \begin{cases} c & \text{if } c \leq 1, \\ \frac{2c+1}{3} & \text{if } c > 1 \end{cases} \quad (2)$$

It is obvious that source 1 always gets bandwidth lower than  $c/3$  and source 2 is favored with more than  $2c/3$  and achieves a much higher utility. In particular, when the link bandwidth is scarce, i.e.,  $0 < c \leq 1$ , by the OFC approach, source 1 is totally prevented from transmission and all the resource is granted to source 2. In the OFC literature, source 1 is usually taken granted as a low bandwidth demander and source 2 as a high demander, but this does not make sense from an engineering point of view. In contrast, source 1 is really the high demander since it needs more bandwidth to achieve the same utility performance than source 2 does.

In this paper, we will focus on the research of max–min flow control mainly from a utility perspective, and more importantly, attempt to develop its corresponding physical model, i.e., priority service networks. Based on the network model, we first propose a novel flow control algorithm to achieve the *bandwidth max–min* fair resource allocation. Taking into account the different QoS requirements of real-time applications, a *utility max–min* flow control algorithm is then presented in the sequel. These new algorithms can also be applied directly to general networks without priority services and can be considered as a unified framework to achieve either bandwidth max–min fairness or utility max–min fairness.

The rest of the paper is organized as follows. We give a new model of priority services in Sect. 2 and propose a max–min fair flow control algorithm in Sect. 3. In Sect. 4, we further study the network flow control problem for utility max–min fairness. The implementation issues are discussed in Sect. 5, where the application to general networks without priority services is discussed. Finally, in Sect. 6 we present the numerical results to illustrate the performance of the algorithms and conclusions are drawn in Sect. 7.

## 2 Problem Formulation and Link Priority Model

Consider a network that consists of a set  $\mathcal{L} = \{1, 2, \dots, L\}$  of links of capacity  $c_l$ ,  $l \in \mathcal{L}$ . The network is shared by a set  $\mathcal{S} = \{1, 2, \dots, S\}$  of sources. For each source  $s$ , define  $\mathcal{L}_s \subseteq \mathcal{L}$  be a subset of links that connect source  $s$  to the destination,  $x_s \in [m_s, M_s]$  be the associated transmission rate where  $m_s \geq 0$  is the minimum rate requirement and  $M_s < \infty$  is the maximum rate requirement,  $x = [x_1, x_2, \dots, x_S]^T$ . For each link  $l$ , define  $\mathcal{S}_l = \{s \in \mathcal{S} | l \in \mathcal{L}_s\}$  be the set of sources that traverse link  $l$ . Note that  $l \in \mathcal{L}_s$  if and only if  $s \in \mathcal{S}_l$ .

Suppose that the network uses priority services to provide differentiated QoS. In particular, we assume that the network supports a continuum of priorities given by the set  $\mathcal{P} = [0, \infty)$ . At each link in the network, traffic is served according to a strict priority rule, i.e., priority  $p$  traffic is transmitted only if all traffic with priority  $q > p$  has been served, and priority  $p$  traffic is dropped due to buffer overflow only if there is no traffic with priority  $q < p$  left in the buffer. To provide an efficient traffic management in the priority service network, the flow control mechanism should both determine the source transmission rate and assign an appropriate priority to the associated traffic.

In the general sense of differentiated services, a user may decide the priority of the traffic depending on the QoS requirement. A more strict QoS is required, a higher priority is assigned for better service. Although this simple priority scheme sounds reasonable, indeed how to properly assign the priority among competing users with different QoS requirements is a challenging problem. If users are allowed to freely choose the priority without any charge of the service, naturally every user would like to increase its own benefit by seeking the highest priority available in the network, and this does nothing but degrades priority services into a “best effort” scheme. One possible solution of this problem is to employ a network

controller, which assigns priorities to applications based on some pre-defined policies. Though this approach can help provide tight priority control over different users, it is often costly to implement, that is, it is not straightforward for the priority controller to obtain all the necessary information regarding the QoS requirements and traffic patterns of individual user in order to make the proper decision.

In this paper, we investigate the priority services and propose a new scheme for priority assignment, as well as a flow control algorithm in the network with QoS differentiated services. Unlike previous work that allows users to select priorities according to their QoS requirements, instead, we present a novel approach where links can dynamically adjust the priority thresholds based on the congestion information in the network. The more severe the congestion is, the higher the priority threshold is selected, and vice versa. In this case, to guarantee the application is serviced timely and reliably, each source assigns a priority that is equal to the maximal threshold required by the links along its path to each packet sent. Because the priority requirement at the same time indicates the congestion status of the network, the packet sending rate could also be adjusted by the assigned priority. The more congested the networks is, the slower the data is sent. In this way, with a proper selection of sending rate according to the priority assignment, we implicitly enforce an efficient flow control and yield max–min fair resource allocation in priority service networks.

Let us now focus on the link priority model. To make the network congestion-free, the aggregate rate  $x^l$  at each link  $l$  must not exceed the physical link capacity  $c_l$ , namely,

$$x^l = \sum_{s \in \mathcal{S}_l} x_s \leq c_l, \quad \forall l \in \mathcal{L}. \quad (3)$$

In principle, if the link capacity constraints are violated, buffers may eventually overflow resulting in the packet loss. In this scenario, it will be difficult for the relevant sources to make the correct decision, i.e., whether to increase the packet priorities or to decrease the sending rates.

The above problem is caused by a lack of explicit congestion feedback mechanism in the network. Hence, to offer better congestion information for the sources, we introduce a new measurement  $p_l$  named “priority threshold” for each link  $l$ . It is likely to say that only the packets with priority  $p \geq p_l$  can be served timely by link  $l$  subject to link capacity and congestion, and the packets with lower priority  $p < p_l$  need to wait in the buffer or may be instantaneously discarded.

Suppose at time step  $t$ , each link  $l$  uses the following link algorithm to update the priority threshold  $p_l$  for the next time step  $t + 1$

$$p_l(t + 1) = [p_l(t) + \gamma(x^l(t) - c_l)]^+ \quad (4)$$

where  $\gamma > 0$  is a small step size, and  $x^l(t) = \sum_{s \in \mathcal{S}_l} x_s(t)$  is the aggregate source rate at link  $l$ . The projection  $[z]^+ = \max\{0, z\}$  ensures non-negativeness of the priority threshold at each link. Equation (4) reveals that if the aggregate source rate at link  $l$  exceeds the link capacity  $c_l$ , the priority threshold is increased, otherwise, it is decreased. The simple intuition is that the link will raise its priority threshold when

the network is congested, and vice versa. Meanwhile, the above priority threshold  $p_l(t)$  is completely self-defined and is not related to the assigned priorities for the incoming packets. We will next study in detail how this information can be used by the network application for both the priority assignment and rate adjustment.

### 3 Max–Min Fair Flow Control in Priority Services

Throughout the paper, we assume that in the network there exists an explicit congestion information feedback mechanism (we will address the implementation issue on explicit congestion feedback in Sect. 5) and the priority requirement  $p_l$  updated by link algorithm (4) is reachable by all the sources who use link  $l$  in their paths.

Suppose each source  $s$  receives priority requirements  $p_l$  from all the links  $l \in \mathcal{L}_s$  along its path, it is reasonable to assign a priority  $p^s$  to each originated packet that satisfies its traversing links, such that

$$p^s \geq p_l, \quad \text{for all } l \in \mathcal{L}_s \tag{5}$$

This is equivalent to say

$$p^s \geq \max_{l \in \mathcal{L}_s} p_l \tag{6}$$

indicating the priority assignment  $p^s$  should fulfil the strictest requirement, i.e., the highest priority requirement along its path.

If the priority services are free of charge, the above information might give little help with regard to priority assignment. Every source is inherently selfish, that is, a source will always select the highest priority class available in order to maximize the chance of transmitting its own packets successfully, even if a lower priority assignment could satisfy its requirement from (6).

Because the priority requirement also implies the congestion level of each link, we suggest that the priority assignment should not be free of charge any longer and should be further combined with congestion control. For a reliable transmission, a high priority needs to be assigned since links may otherwise discard the incoming packets if they do not meet the corresponding priority requirements. However, a high priority is assigned at the expense of sacrificing transmission rate, i.e., a higher priority class is accompanied by a lower transmission rate. It may sound counter-intuitive initially, but it could be interpreted as a higher priority is compromised by a lower bandwidth allocation, or in other words, a higher rate is possibly achieved by a lower priority assignment but at the risk of packet loss. This policy apparently works as it prevents sources from choosing both a high priority and a high rate, therefore, a fair resource allocation is yielded in the network.

Consider the following rate adaption rule [22–24] when priority  $p^s$  is assigned to every packet sent by source  $s$

$$x_s = \left[ \frac{1}{p^s} \right]_{m_s}^{M_s} \tag{7}$$

where  $[z]_a^b = \max\{a, \min\{b, z\}\}$ . Recall that the priority assignment  $p^s$  must also satisfy the requirement of (6), it is reasonable for each source  $s$  to choose the possibly lowest feasible priority

$$p^s = \max_{l \in \mathcal{L}_s} p_l \tag{8}$$

so as to achieve the highest rate  $x_s$  in (7) for its own benefit. Assuming that the minimum resource allocation  $x = [m_1, m_2, \dots, m_S]^T$  is achievable in the network such that Eq. (3) is satisfied, the distributed flow control algorithm for the priority services is summarized as Algorithm 1.

**Algorithm 1—Bandwidth fair max–min flow control**

• **Link  $l$ 's algorithm:** At time  $t = 1, 2, \dots$ , link  $l$ :

1. Measures the aggregate rate  $x^l(t)$  that goes through link  $l$

$$x^l(t) = \sum_{s \in \mathcal{S}_l} x_s(t) \tag{9}$$

2. Updates a new priority requirement

$$p_l(t + 1) = [p_l(t) + \gamma(x^l(t) - c_l)]^+ \tag{10}$$

3. Communicates the new priority requirement  $p_l(t + 1)$  to all sources  $s \in \mathcal{S}_l$  that traverse link  $l$ .

• **Source  $s$ 's algorithm:** At time  $t = 1, 2, \dots$ , source  $s$ :

1. Receives from the network the highest priority requirement of the links along its path

$$p^s(t) = \max_{l \in \mathcal{L}_s} p_l(t) \tag{11}$$

2. Assigns the packets with a priority  $p^s(t)$  and sends data at a new transmission rate  $x_s(t + 1)$  for the next period

$$x_s(t + 1) = \left[ \frac{1}{p^s(t)} \right]_{m_s}^{M_s} \tag{12}$$

For Eq. (12), if the priority  $p^s$  is viewed as the price paid by source  $s$  for the priority services, it is intuitively fair to specify that each source has the same capital of “1” to purchase the bandwidth resource. Before studying the property of Algorithm 1, we give the formal notion of (bandwidth) max–min fairness [13] for resource allocation.

**Definition 1** A bandwidth allocation  $x = [x_1, x_2, \dots, x_S]^T$  is (bandwidth) max–min fair, if it is feasible (i.e.  $m_s \leq x_s \leq M_s$  and Eq. (3) is satisfied) and for each user  $s$ , its source rate  $x_s$  cannot be increased while maintaining feasibility, without decreasing the source rate  $x_{s'}$  for another user  $s'$  with a rate  $x_{s'} \leq x_s$ .

As mentioned, the max–min fairness emphasizes the equal sharing of the network resource (the stage 1 simulation result given in Sect. 6 could be viewed as a



concrete example). This definition, on the other hand, provides an alternative perspective to interpret a (bandwidth) max–min fair allocation, which gives the most poorly treated user (namely the user who receives the lowest rate) the largest possible share without wasting any network resource. Based on this, we are ready to state the main result of Algorithm 1.

**Theorem 1** *The sequence  $(x(t), p(t))$  generated by the Algorithm 1 will yield an max–min fair bandwidth allocation in the network.*

*Proof* This theorem will be shown as a special case of Theorem 2 given in Sect. 4. The proof could then be deferred upon Theorem 2.  $\square$

*Remark 1* It is critical to choose the parameter  $\gamma$  which has an impact on algorithmic convergence. Usually, larger  $\gamma$  will make the algorithm faster to reach the stable state. According to our earlier results shown in [23], however, it should not be chosen larger than some positive  $\gamma^*$ , otherwise, the algorithm will diverge.

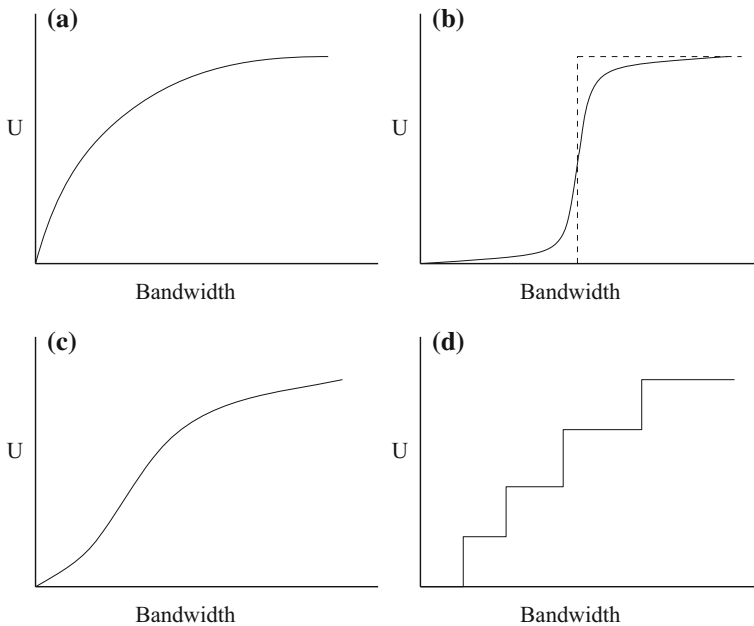
## 4 Utility Max–Min Fairness for Real-Time Applications

For a practical network application, bandwidth allocation may be a concern, but a more important and direct concern to an application is really the utility or QoS performance. The utility function of an application is a measure of its QoS performance based on provided network services such as bandwidth, transmission delay and loss ratio. In this paper, we characterize utility in terms of allocated bandwidth, which is a common modelling approach in the optimal flow control literature.

Back to an early paper by Shenker [25], it has been pointed out that traditional data applications such as file transfer, electronic mail, and web browsing are rather tolerant of throughput and time-delays. This class of applications is termed as *elastic* traffic, and the utility function can be described as a strictly concave function as shown in Fig. 1a. The utility (performance) increases with bandwidth, but the marginal improvement is decreased. It has been extensively studied in OFC literature.

In the priority service network, however, most users are real-time applications such as audio and video delivery, which are generally delay-sensitive and have strict QoS requirements. Unlike *elastic* traffic, they usually have an intrinsic bandwidth threshold because the data generation rate is independent of network congestion. The degradation in bandwidth may cause serious packet drops and severe performance degradation. Thus, a reasonable description of the utility is a sigmoidal-like function as shown in Fig. 1b (solid line), which is convex instead of concave at lower bandwidths. Certain hard real-time applications may even require an exact step utility function as in Fig. 1(b) (dashed line).

There exists another class of real-time rate-adaptive applications which adjust the transmission rate in response to network congestion [25]. At lower and higher bandwidth interval, the marginal utility increment is small with additional bandwidth and the utility curve may have a general shape as in Fig. 1c.



**Fig. 1** Utility functions for different classes of applications: **a** Elastic, **b** Real-time, **c** Rate-adaptive, **d** Stepwise

There are some applications that may take a stepwise utility function as shown in Fig. 1d. Such applications can be found in audio and video delivery systems employing a layered encoding and transmission model [26]. For these applications, bandwidth allocation is limited to some distinct levels. The utility is increased only when an additional level is reached provided an increase in available bandwidths.

Consider the network flow control model formulated in Sect. 2, in addition, each source  $s$  attains a non-negative QoS utility  $U_s(x_s)$  when it transmits at a rate  $x_s \in [m_s, M_s]$ , where  $m_s$  and  $M_s$  are the minimum and maximum transmission rates required by source  $s$ , respectively. The utility function  $U_s(x_s)$  is assumed to be continuous, strictly increasing and bounded (not necessarily to be concave) in the interval  $[m_s, M_s]$ . Without loss of generality, it can be assumed that  $U_s(x_s) = 0$  when  $x_s < m_s$  and  $U_s(x_s) = U_s(M_s)$  when  $x_s > M_s$ .<sup>1</sup>

When dealing with heterogeneous applications with different QoS requirements, it may not be desirable for the network to simply share the bandwidth as conventional max–min fairness does in Sect. 3. Instead, the network should allocate the bandwidth to the competing users according to their different QoS utilities. This motivates the proposal for the criterion of *utility (weighted) max–min fairness* [15, 27].

**Definition 2** A bandwidth allocation  $x = [x_1, x_2, \dots, x_S]^T$  is utility (weighted) max–min fair, if it is feasible and for each user  $s$ , its utility  $U_s(x_s)$  cannot be

<sup>1</sup> For the scalability, it can be further assumed that  $0 \leq U_s(x_s) \leq 1$  and  $U_s(M_s) = 1$ .

increased while maintaining feasibility, without decreasing the utility  $U_{s'}(x_{s'})$  for some user  $s'$  with a lower utility  $U_{s'}(x_{s'}) \leq U_s(x_s)$ . Bandwidth max–min fair allocation is recovered with

$$U_s(x_s) = x_s, \quad s = 1, \dots, S.$$

Based on the framework of Algorithm 1, the utility max–min fairness can be achieved with a minor modification of the source algorithm.

**Algorithm 2—Utility fair max–min flow control**

• **Link  $l$ 's algorithm:**

The same as in Algorithm 1.

• **Source  $s$ 's algorithm:** At time  $t = 1, 2, \dots$ , source  $s$ :

1. Receives from the network the highest priority requirement of the links along its path

$$p^s(t) = \max_{l \in \mathcal{L}_s} p_l(t) \tag{13}$$

2. Sends the packets with a priority  $p^s(t)$  and at a new transmission rate  $x_s(t + 1)$  for the next period

$$x_s(t + 1) = U_s^{-1} \left( \left[ \frac{1}{p^s(t)} \right]_{U_s(m_s)}^{U_s(M_s)} \right) \tag{14}$$

where  $[z]_a^b = \max\{a, \min\{b, z\}\}$  and  $U_s^{-1}(\cdot)$  is the inverse function of  $U_s(\cdot)$ .

By rearranging Eq. (14), we get

$$U_s(x_s(t + 1)) = \left[ \frac{1}{p^s(t)} \right]_{U_s(m_s)}^{U_s(M_s)} \tag{15}$$

Compared to Eq. (12), at each step  $t$ , each source  $s$  now is allocated with a new bandwidth  $x_s(t + 1)$  such that the associated utility is reciprocal to the highest link priority requirement to enforce utility fairness (instead of bandwidth fairness).

Similar to Theorem 1, we have the following theorem about the result of Algorithm 2.

**Theorem 2** *The sequence  $(x(t), p(t))$  generated by Algorithm 2 will yield a utility max–min fair bandwidth allocation in the network.*

*Proof* The structure of the following proof involves two steps: (1) a utility max–min fair rate allocation is uniquely existing; (2) the resulting rate allocation from Algorithm 2 is indeed utility max–min fair. □

**(1) Existence and Uniqueness** Given that all utility functions are continuous and increasing, by Lemma 1, there exists a unique utility max–min fair rate allocation.

**Lemma 1** [28] *Considering a mapping  $U$  defined by*

$$(x_1, \dots, x_S) \rightarrow (U_1(x_1), \dots, U_S(x_S)), \quad (16)$$

if  $U_s$  is continuous and increasing for all  $s$ , then  $U(x)$  is uniquely max–min achievable.

**(2) Utility Max–Min Fairness** When the algorithm reaches a stable state, denoted as  $(x^*, p^*)$ ,

$$U_s(x_s^*) = \begin{cases} U_s(m_s) & \text{if } p^{s^*} > \frac{1}{U_s(m_s)} \\ \frac{1}{p^{s^*}} & \text{if } \frac{1}{U_s(M_s)} \leq p^{s^*} \leq \frac{1}{U_s(m_s)} \\ U_s(M_s) & \text{if } p^{s^*} < \frac{1}{U_s(M_s)} \end{cases} \quad (17)$$

At the stable state, the associated utility  $U_s$  of source  $s$  is equal to  $\frac{1}{p^{s^*}}$  when  $p^{s^*} \in [\frac{1}{U_s(M_s)}, \frac{1}{U_s(m_s)}]$ , otherwise, it attains a utility  $U_s(m_s)$  of the minimum rate requirement whose value is greater than  $\frac{1}{p^{s^*}}$  (it cannot be decreased anymore due to QoS requirement), or a utility  $U_s(M_s)$  of the maximum rate requirement whose value is less than  $\frac{1}{p^{s^*}}$  (it needs not to be increased any further). For the latter two cases, the source rates are already fixed at their minimum and/or maximum requirements. We then need to focus considering the resource allocation among the sources who attain a normal utility  $U_s^* = \frac{1}{p^{s^*}}$ .

Theorem 2 states that the source rate will be determined by the *maximum* link priority threshold along the path. In a nutshell, each source will be bottlenecked by a particular link. Assume at the stable state, there are  $K$  different link priority thresholds in the network with

$$p_1 > p_2 > \dots > p_{K-1} > p_K.$$

We first select the links with the highest priority requirement  $p_1$  and refer them as  $l_{p_1}$ , then all the sources  $s \in \mathcal{S}_{l_{p_1}}$  which traverse links  $l_{p_1}$  attain the same utility  $U_s = 1/p_1$ , which are the smallest allocated utilities compared with others. If we apply the utility max–min condition only to this set of sources ( $\mathcal{S}_{l_{p_1}}$ ), we see that they are utility max–min fair. Because if there is a source  $s \in \mathcal{S}_{l_{p_1}}$  that increases the utility  $U_s$  by increasing its transmission rate  $x_s$ , there must be another source  $s' \in \mathcal{S}_{l_{p_1}}$  to decrease its rate  $x_{s'}$  and further decrease its utility  $U_{s'}$  which is previously equal to  $U_s$ . In other words, no source can increase its utility without decreasing another one's within  $\mathcal{S}_{l_{p_1}}$ , which is the definition of utility max–min fairness exactly. We now extend this argument to include sources bottlenecked by links with priority threshold  $p_2$ .

The  $l_{p_2}$  set of links are the links with the second highest link priority threshold  $p_2$ ,  $p_1 > p_2 > p_k, k \neq 1, 2$ . All the sources  $s \in \mathcal{S}_{l_{p_2}} \setminus \mathcal{S}_{l_{p_1}}$  which traverse link  $l_{p_2}$  but not traverse link  $l_{p_1}$  have the same utility  $U_s = 1/p_2$ . Since we have already shown that the sources in  $\mathcal{S}_{l_{p_1}}$  are utility max–min fair and the utility for the sources in  $\mathcal{S}_{l_{p_2}} \setminus \mathcal{S}_{l_{p_1}}$  are equal, if there is a source  $s \in \mathcal{S}_{l_{p_2}} \setminus \mathcal{S}_{l_{p_1}}$  that increases its rate and utility, there must be

another source  $s' \in \mathcal{S}_{l_{p_2}}$  to decrease its rate which already has a lower utility  $U_{s'} \leq U_s$ . Thus the utility max–min fairness holds for all the sources within  $\mathcal{S}_{l_{p_2}} \cup \mathcal{S}_{l_{p_1}}$ .

Continuing in this manner, selecting all the links with positive priority threshold in the order  $p_1, p_2, \dots, p_{K-1}, p_K$ , by induction it is concluded that the entire source rate allocation is utility max–min fair and the global fairness is achieved.  $\square$

*Remark 2* If we let  $U_s = x_s$  for all sources  $s \in \mathcal{S}$ , Algorithm 2 degenerates to Algorithm 1. Therefore, we could conclude that Algorithm 1 is a special case of Algorithm 2 and the stable state of Algorithm 1 is bandwidth max–min fair.

*Remark 3* As from the flow control standpoint of view, our model emphasizes the relationship between bandwidth allocation and QoS performance of applications. It is *implicitly* assumed that as long as the application is allocated sufficient bandwidth, it will be served timely and reliably. However, especially for real-time applications, it will be more challenging to *explicitly* consider the packet delay effects. One possible extension in this direction is to follow the work suggested by Li [29] by defining a new utility function of source  $s$  in order to incorporate the delay as

$$U_s = U_s(x_s) - \beta_s \sum_{l \in \mathcal{L}_s} d_l(x^l)$$

where  $d_l(x^l)$  is the average delay incurred by a packet on link  $l$  and thus  $\sum_{l \in \mathcal{L}_s} d_l(x^l)$  is the end-to-end average packet delay.  $\beta_s > 0$  is some tuning parameter to reflect the relative importance of the source rate versus delay.

## 5 Network Implementations

In this section, we discuss the implementation issues of the proposed (utility) max–min fair flow control algorithms.

### 5.1 On-Line Buffer Measurement

When the (utility) max–min flow control algorithms reach the stable state, the aggregate source rate at each bottleneck link will be equal to the link capacity. Since there is no mechanism in link algorithm to control the buffer occupancy, due to the statistical process of packet transmission in the practical network, it will lead to serious buffer overflow and significant queuing delay from the queuing theory. Hence, we make the following enhancements to the basic link algorithm as in [23] by using well-known “on-line measurement” technique.

At time  $t$ , the buffer backlog  $b_l(t)$  of link  $l$  is updated automatically according to<sup>2</sup>

<sup>2</sup> Here we use a *deterministic* approach to estimate the buffer dynamics, and assume the updating time interval is 1. Otherwise

$$b_l(t) = [b_l(t-1) + \text{time\_interval}(x^l(t) - c_l)]^+$$

and this only results in a weighting coefficient change from  $\gamma$  to  $\gamma/\text{time\_interval}$  in the new link algorithm (20).

$$b_l(t) = [b_l(t-1) + (x^l(t) - c_l)]^+ \quad (18)$$

in which we assume the buffer size at each link is sufficiently large and never induces a buffer overflow.

Multiplying both sides of (18) by  $\gamma$ , the step size in link algorithm (10), we have

$$\gamma b_l(t) = [\gamma b_l(t-1) + \gamma(x^l(t) - c_l)]^+ \quad (19)$$

Comparing Eq. (19) with (10), we yield the alternative link priority adaptation rule based on the buffer backlog information  $b_l(t)$  at link  $l$

$$p_l(t+1) = \gamma b_l(t) \quad (20)$$

For the new link algorithm (20), the priority requirement is updated by the local buffer backlog information. It is not only much simpler than the basic algorithm (10), with the implementation of source algorithm, but also the buffer backlog at each link can be well maintained under such a built-in close loop feedback system. Furthermore, the packet loss due to overflow is greatly avoided.

## 5.2 Application to the Networks without Priority Services

Even though the (utility) max–min flow control algorithm is derived by the priority service model, it can be directly applied to the general networks and considered as a unified framework to achieve (utility) max–min fairness.

Because the link algorithm is independent of the priority assignment in the incoming packets, in the network where the priority service is not provided, we can treat the priority requirement  $p_l$ , adapted by (10) or (20), as the link price of link  $l$ , which indicates the network congestion status as in the OFC literature. The source rate is the same as adjusted by the algorithm (12) or (14), but the path price is defined as the *highest* link price in the path.

For instance, in the available bit rate (ABR) service of ATM network, the maximal link price  $p^*$  defined by (13) can be easily informed to the source  $s$  with the help of special resource management(RM) cells. Each ATM source sends a RM cell to its destination periodically (normally after 32 data cells are transmitted) to collect congestion information from the network and uses this information to update the transmitting rate. In this paradigm, ATM source can collect its path price (highest link price) through the 16 bit explicit rate (ER) field in RM cells. When a RM cell is sent by the source, its ER field is initialized to 0. As the RM cell circulates through the network, each link examines the ER field and compares it with the current link price. If its link price is greater than the value in the ER field, the link sets the ER field to its current link price, otherwise keeps it unchanged. In this way, when the RM cell reaches the destination, it contains the maximal link price along the path. The destination then transmits the RM cell back to the source. Therefore, the source is able to use the new link price in the ER field to updates the source rate according to (14).

In the Future Internet, the maximum link price can be fed back to each source by the differentiated services code point (DSCP) [1] set in the IP header. To support

real-time traffic without disturbing current IP structure, IETF adopted a new architecture named “differentiated services” (Diff-Serv), in which the first 6 b (with a potential for all 8 b) in the IPv4 type of service (ToS) octet and the IPv6 traffic class octet are reserved as differentiated services code point (DSCP). We advocate the use of DSCP as an explicit congestion feedback mechanism in order to provide a better solution for congestion control and resource allocation in the Future Internet. Following that, DSCP works similarly to the ER field of the RM cell in ATM network.

### 6 Numerical Example and Simulation Results

Consider the network topology as shown in Fig. 2, consisting of four links L1–L4, each with a capacity of 15 Mbps and shared by eight sources S1–S8. S1, S2, S3 and S4 traverse link L1, L2, L3 and L4 respectively, S5 traverses L1 and L2, S6 traverses L2 and L3, S7 traverses L3 and L4, and S8 traverses all the four links.

Their utilities are shown in Fig. 3a, in which S1 to S4 attain the same linear utility  $0.1x$ , S5 and S6 are elastic traffic with utility  $\log(x + 1)/\log 11$ , S7 and S8 are real-time applications which have a sigmoidal-like utility  $1/(1 + e^{-2(x-5)})$ . All the sources have their maximum rate requirement of 10 Mbps.

The simulation contains two stages:

- Stage 1:  $t = 0 \rightarrow 10$  s, Algorithm 1 is adopted for bandwidth max–min fairness.
- Stage 2:  $t = 10 \rightarrow 20$  s, Algorithm 2 is adopted for utility max–min fairness.

where the step size  $\gamma = 0.005$ , that is, the sources and links execute their respective algorithms iteratively every 5 ms. The simulation results are given in Fig. 3b–d. It can be observed that in both stages the source rates reach the stable state in less than 2 s, and the stable values of bandwidth and utility are further listed in Table 1.

In Stage 1, although bandwidth max–min fairness that emphasizes the equal sharing of bandwidth is achieved, it does not always make sense especially towards real-time applications. Particularly in this case, given the bandwidth allocated to S7 and S8 ( $x_7 = x_8 = 3.7500$ ), their associated utility is even  $<0.1$  ( $U_7 = U_8 = 0.0759$ ) that is far than sufficient and meaningful to the users. It indeed motivates the utility max–min fairness criterion to deal with heterogeneous applications.

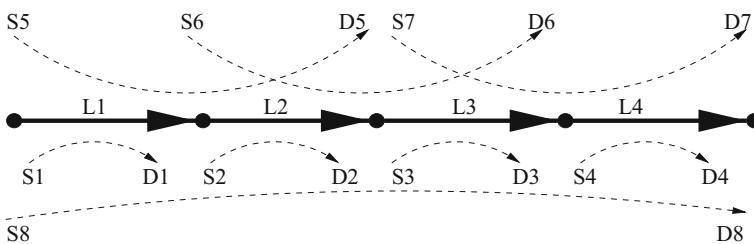
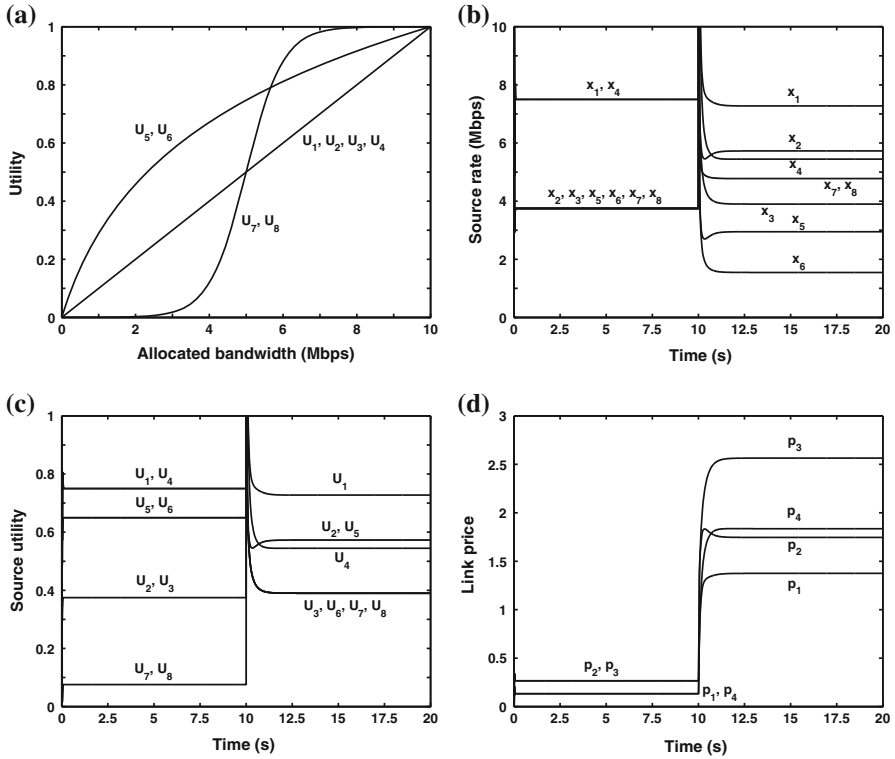


Fig. 2 Network topology



**Fig. 3** Simulation results of bandwidth fair and utility fair max–min flow control: **a** utility functions, **b** source rates, **c** source utilities, **d** link priorities (prices)

**Table 1** Comparison of bandwidth fair and utility fair max–min flow control

Source	Algorithm 1: bandwidth fair		Algorithm 2: utility fair	
	Bandwidth	Utility	Bandwidth	Utility
$S_1$	7.5000	0.7500	7.2750	0.7275
$S_2$	3.7500	0.3750	5.7274	0.5727
$S_3$	3.7500	0.3750	3.8998	0.3900
$S_4$	7.5000	0.7500	5.4474	0.5447
$S_5$	3.7500	0.6498	2.9487	0.5727
$S_6$	3.7500	0.6498	1.5476	0.3900
$S_7$	3.7500	0.0759	4.7763	0.3900
$S_8$	3.7500	0.0759	4.7763	0.3900

In Stage 2, the associated utilities of all the sources are  $U = (0.7275, 0.5727, 0.3900, 0.5447, 0.5727, 0.3900, 0.3900, 0.3900)$ , which can be verified through the computation to be utility max–min fair, and the link priority thresholds (prices) are  $p = (1.3746, 1.7460, 2.5642, 1.8357)$ . In details,  $S_3, S_6, S_7$  and  $S_8$  achieve the lowest utility of 0.3900 due to the highest link priority of 2.5642



at bottleneck L3 they all traverse. S4 achieves the second lowest utility of 0.5447 due to the second highest link priority of 1.8357 at L4. S2 and S5 share the bottleneck L2 with the link priority of 1.7460 and achieve the same utility of 0.5727. S1 attains the highest utility of 0.7275 due to the lowest link priority at L1. This confirms that the priority assignment scheme with flow control algorithm given in this paper works effectively and results in an efficient utility max–min fair resource allocation among heterogeneous applications. Compared to Algorithm 1, Algorithm 2 involves additional computation to inverse the utility functions when calculating the source transmission rates. In both cases, their utility functions may not need to satisfy the critical strictly concave condition that is required by the standard OFC approach.

## 7 Conclusions

In this paper, we propose a new priority assignment scheme for priority service networks and a distributed flow control algorithm is developed to achieve the (utility) max–min fairness among different users. Although the algorithm is derived based on the priority service model, it can be also deployed in the general networks where no priority services are provided. Indeed, it is a unified framework to achieve either bandwidth max–min fairness or utility max–min fairness through link pricing policy.

For the traditional max–min fair approaches, it is usually the links that attempt to make bandwidth allocation for different users based on global information. Whereas in our method, each link updates the priority requirement (link price) only according to the congestion situation in the network, and it is the sources that adjust their rates automatically according to the maximum priority requirement (highest link price) along their paths.

The max–min fair flow control algorithm proposed in this paper merely requires that the source utility function is positive, strictly increasing and bounded over bandwidth, and does not require the strictly concave condition that is stipulated by the standard OFC approach. Therefore, our new algorithms are not only suitable for traditional elastic data traffic, but are also capable of providing an efficient flow control and resource allocation strategy for real-time applications in the Future Internet.

**Acknowledgments** This work was supported by Swinburne University of Technology under the Early Research Career Scheme 2014. It is also partly supported by funding from the Faculty of Engineering and Information Technologies, The University of Sydney, under the Faculty Research Cluster Program.

## References

1. Differentiated Services (Diffserv) Working Group. The Internet Engineering Task Force. Accessed Oct 2015 (online). <http://datatracker.ietf.org/wg/diffserv/charter/>
2. Kelly, F.P., Maulloo, A., Tan, D.: Rate control for communication networks: shadow prices, proportional fairness and stability. *J. Oper. Res. Soc.* **49**(3), 237–252 (1998)

3. Low, S.H., Lapsley, D.E.: Optimization flow control—I: basic algorithm and convergence. *IEEE/ACM Trans. Netw.* **7**(6), 861–874 (1999)
4. Vo, P.L., Tran, N.H., Hong, C.S., Lee, S.: Network utility maximisation framework with multiclass traffic. *IET Netw.* **2**(3), 152–161 (2013)
5. Li, S., Sun, W., Tian, N.: Resource allocation for multi-class services in multipath networks. *Perform. Eval.* **92**, 1–23 (2015)
6. Xiao, M., Shroff, N.B., Chong, E.K.P.: A utility-based power-control scheme in wireless cellular systems. *IEEE/ACM Trans. Netw.* **11**(2), 210–221 (2003)
7. Gong, S.-L., Roh, H.-T., Lee, J.-W.: Cross-layer and end-to-end optimization for the integrated wireless and wireline network. *J. Commun. Netw.* **14**(5), 554–565 (2012)
8. Tychogiorgos, G., Leung, K.K.: Optimization-based resource allocation in communication networks. *Comput. Netw.* **66**, 32–45 (2014)
9. Xue, Y., Li, B., Nahrstedt, K.: Optimal resource allocation in wireless ad hoc networks: a price-based approach. *IEEE Trans. Mob. Comput.* **5**(4), 347–364 (2006)
10. Hou, Y.T., Shi, Y., Sherali, H.D.: Rate allocation and network lifetime problems for wireless sensor networks. *IEEE/ACM Trans. Netw.* **16**(2), 321–334 (2008)
11. Jin, J., Palaniswami, M., Krishnamachari, B.: Rate control for heterogeneous wireless sensor networks: characterization, algorithms and performance. *Comput. Netw.* **56**(17), 3783–3794 (2012)
12. Tychogiorgos, G., Gkelias, A., Leung, K.K.: A non-convex distributed optimization framework and its application to wireless ad-hoc networks. *IEEE Trans. Wirel. Commun.* **12**(9), 4286–4296 (2013)
13. Bertsekas, D., Gallager, R.: *Data Networks*, 2nd edn. Prentice-Hall, Upper Saddle River (1992)
14. Mo, J., Walrand, J.: Fair end-to-end window-based congestion control. *IEEE/ACM Trans. Netw.* **8**(5), 556–567 (2000)
15. Cao, Z., Zegura, E.W.: Utility max–min: an application oriented bandwidth allocation scheme. In: *Proceedings of the IEEE INFOCOM 1999*, pp. 793–801 (1999)
16. Jin, J., Wang, W.H., Palaniswami, M.: A simple framework of utility max–min flow control using sliding mode approach. *IEEE Commun. Lett.* **13**(5), 360–362 (2009)
17. Han, B.Q., Feng, G.F., Chen, Y.F.: Heterogeneous resource allocation algorithm for ad hoc networks with utility fairness. *Int. J. Distrib. Sens. Netw.* **2015**, 686189 (2015)
18. Marbach, P.: Priority service and max–min fairness. *IEEE/ACM Trans. Netw.* **11**(5), 733–746 (2003)
19. Marbach, P.: Analysis of a static pricing scheme for priority services. *IEEE/ACM Trans. Netw.* **12**(2), 312–325 (2004)
20. Hahne, E.L.: Round-robin scheduling for max–min fairness in data networks. *IEEE J. Sel. Areas Commun.* **9**(7), 1024–1039 (1991)
21. Srikant, R.: *The Mathematics of Internet Congestion Control*. Birkhauser, Cambridge (2004)
22. Wang, W.H., Palaniswami, M., Low, S.H.: Application-oriented flow control: fundamentals, algorithms and fairness. *IEEE/ACM Trans. Netw.* **14**(6), 1282–1291 (2006)
23. Jin, J., Wang, W.H., Palaniswami, M.: Utility max–min fair resource allocation for communication networks with multipath routing. *Comput. Commun.* **32**(17), 1802–1809 (2009)
24. Harks, T., Poschwatta, T.: Congestion control in utility fair networks. *Comput. Netw.* **52**, 2947–2960 (2008)
25. Shenker, S.: Fundamental design issues for the Future Internet. *IEEE J. Sel. Areas Commun.* **13**(7), 1176–1188 (1995)
26. Xiong, N., Jia, X., Yang, L.T., Vasilakos, A.V., Li, Y., Pan, Y.: A distributed efficient flow control scheme for multirate multicast networks. *IEEE Trans. Parallel Distrib. Syst.* **21**(9), 1254–1266 (2010)
27. Jin, J., Yuan, D., Zheng, J., Dong, Y.N.: Sliding mode-like congestion control for communication networks with heterogeneous applications. In: *Proceedings of the IEEE ICC 2015, London, UK* (2015)
28. Radunovic, B., Boudec, J.-Y.L.: A unified framework for max–min and min–max fairness with applications. *IEEE/ACM Trans. Netw.* **15**(5), 1073–1083 (2007)
29. Li, Y., Papachristodoulou, A., Chiang, M., Calderbank, A.R.: Congestion control and its stability in networks with delay sensitive traffic. *Comput. Netw.* **55**, 20–32 (2011)

**Jiong Jin** received his Ph.D. degree from the University of Melbourne, Australia, in 2011. He is currently a Lecturer at Swinburne University of Technology, Australia. His research interests include network design and optimization, nonlinear systems and sliding mode control, networked robotics, wireless sensor networks and Internet of things, cyber-physical systems and applications in smart grids and smart cities.

**Marimuthu Palaniswami** received his Ph.D. degree from the University of Newcastle, Australia. He is currently a Professor at the University of Melbourne and Director of Research Network on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP). His research interests are in the fields of computer networks, sensors and sensor networks, machine learning, neural network, pattern recognition, signal processing and control.

**Dong Yuan** received his Ph.D. degree from Swinburne University of Technology, Australia, in 2012. He is a Lecturer at the University of Sydney. His research interests include cloud computing, data management in parallel and distributed systems, scheduling and resource management, Internet of Things, BPM and workflow systems.

**Yu-Ning Dong** received his Ph.D. degree from Southeast University, and M.Phil. degree in Computer Science from The Queen's University of Belfast. He is currently a Professor with College of Communications and Information Engineering, Nanjing University of Posts and Telecommunications. His research interests include multimedia communications, wireless networking, and network traffic identification.

**Klaus Moessner** is a Professor in the Institute for Communication Systems (formerly CCSR) at the University of Surrey, UK. His research work includes reconfigurability on different system levels, including object and function virtualization, management and control of virtualized resources, and cognitive networks. He currently leads the work area on Access System Architecture and Coexistence in the 5G Innovation Centre, UK.