Ann. Telecommun.
DOI 10.1007/s12243-010-0199-6

Congestion-aware fair rate control in wireless mesh networks

Muhammad Mahbub Alam · Md. Shariful Islam · Md. Abdul Hamid · Choong Seon Hong · Sungwon Lee

Received: 5 March 2010 / Accepted: 17 August 2010

Abstract This paper presents a fair and efficient rate control mechanism, referred to as *congestion-aware fair rate control* (CFRC), for IEEE 802.11s-based wireless mesh networks. Existing mechanisms usually concentrate on achieving fairness and achieve a poor throughput. This mainly happens due to the synchronous rate reduction of neighboring links or nodes of a congested node without considering whether they actually share the same bottleneck or not. Furthermore, the achievable throughput depends on the network load, and an efficient fair rate is achievable when the network load is balanced. Therefore, existing mechanisms usually achieve a fair rate determined by the mostly loaded network region. CFRC uses an AIMD-based rate control mechanism which enforces a rate-bound to the links that use the same bottleneck. To achieve the maximum achievable rate, it balances the network load in conjunction with the routing mechanism. Furthermore, it allows the intra-mesh flows to utilize the network capacity, and the intra-mesh flows achieve a high throughput. Finally, we investigate the performance of CFRC using simulation in ns-2, and the results demonstrate that CFRC increases the throughput with the desired fairness.

Keywords Wireless mesh networks · Congestion control · Rate control · AIMD · Fairness

M. M. Alam · Md. S. Islam · C. S. Hong (✉) · S. Lee
Department of Computer Engineering,
Kyung Hee University,
1 Seocheon, Giheung, Yongin,
Gyeonggi 449-701, Korea
e-mail: cshong@khu.ac.kr

M. M. Alam
e-mail: mahbub@networking.khu.ac.kr

Md. S. Islam
e-mail: sharif@networking.khu.ac.kr

S. Lee
e-mail: drsungwon@khu.ac.kr

Md. A. Hamid
Department of Information and Communications Engineering, Hankuk University of Foreign Studies,
89 Wangsan-ri, Mohyun-myon, Cheoin-Gu,
Yongin-si, Gyronggi-do 449-791, Korea
e-mail: hamid@hufs.ac.kr

1 Introduction

Wireless Mesh Networks (WMNs) are envisioned to extend the coverage of the last mile wireless access by replacing existing wired technologies, and are getting popular due to their unique features such as ease of installation, easy network maintenance, robustness, and reliable service coverage [1]. A significant effort is being put forward in developing a separate standard for WMNs, IEEE 802.11s [2], to extend the single-hop Wireless LAN technologies to multihop wireless networks.

However, a widespread implementation of WMNs still suffers from significant technical challenges. One of the major challenges addressed by the research community is to find high throughput paths, and increase the network throughput [3]. In contrast, to meet the diverse user requirements in multihop networks, it is also important to develop mechanisms for fairly distributing the achieved network capacity among the active flows [4]. Furthermore, users might have different contributions to the networks, and the flows might have different priorities. Therefore, the fairness in accessing the network resource might become a more general problem, *weighted fairness*, where the throughputs achieved by the flows are proportional to their weight [5].

Furthermore, a prominent portion of the current Internet traffic uses UDP as the transport protocol, and becomes a threat for the fair operation. Moreover, the performance of the TCP rate and congestion control still suffers from many drawbacks in wireless networks. TCP does not explicitly account for the fact that multihop wireless networks require special handling for the rate convergence, and exhibits poor fairness. Therefore, the WMN needs to provide the fair network access at the lower layer regardless of the transport protocol used for the flows. In contrast, a majority of the traffic in WMNs is the Internet traffic which might coexist with the intra-mesh traffic. Most of the existing mechanisms consider only the intra-mesh traffic [4, 6]. However, the many-to-one paradigm [7] (or the one-to-many paradigm for the downstream traffic) of the Internet traffic demands that the flows should achieve a fair throughput irrespective of their location [8]. Therefore, the achievable capacity depends on the traffic passing through a link, its neighboring links, and the paths toward the base station. If the network traffic is not balanced, the fair rate of all the flows are determined based on the mostly loaded path. However, re-routing the traffic toward a lightly loaded network region or toward different base stations might increase the network throughput, which existing rate control mechanisms do not address.

In this paper, we focus to design a rate control mechanism based on the incipient network congestion, which controls the rates of the locally connected flows, and the transit flows at layer 2 using functionalities of both the MAC and routing mechanisms. CFRC maintains a rate for all the outgoing links, and controls the rates of the flows forwarded in a link based on the rate of the link. Furthermore, a weight is associated with each flow, and therefore, the rate of a link is maintained for a flow of unit weight, referred to as *unit-flow rate*,[1] which is the rate of a flow normalized by its weight.

CFRC uses the additive increase and multiplicative decrease (AIMD) to control the rates of the links, and converges the rates to the fairness. In wireless networks, AIMD needs the synchronization of rate decrements of the links to converge the rates to fairness [5]. However, synchronizing the rate decrements of all neighbors might decrease the network throughput, in particular, if multipath routing is used, and/or there are more than one base station. We find that congestion in multihop wireless networks is a path dependent phenomenon, and if the rate decrement is synchronized without considering the actual paths used by the links, the achievable rates of the links are determined based on the mostly loaded network region. Furthermore, it restricts the links to explore the rates based on the actual loads of the paths used by the links. Therefore, CFRC synchronizes the rate decrement of the links that use the same bottleneck link to forward their packets. The bottleneck link enforces a *rate-bounded backpressure* to all the links that use it to forward their packets. This increases the network throughput, and flows using the same bottleneck achieve a fair throughput.

However, if the links are allowed to achieve rates independently based on the loads of the paths used to forward their traffic, throughput of the flows using different bottleneck regions might be different. Therefore, CFRC balances the network load to achieve a global fairness. The routing functionalities within the WMN are performed by the MAC layer in IEEE 802.11s [9, 10], which opens the opportunity of efficient congestion and rate control jointly by both the MAC and routing mechanisms. The draft standard, however, just states the necessity of such a mechanism, and keeps the issue open. CFRC's load balancing mechanism distributes the network load fairly among the paths in corporation with the routing module, and increases the network throughput with the desired fairness.

The rest of the paper is organized as follows. Section 2 details the existing mechanisms. We explain the system model and assumptions in Section 3. The detailed design of the proposed mechanism is explained in Section 4, and Section 5 explains the selection of the parameters. The performance of the proposed mechanism is demonstrated in Section 6. Finally, we conclude in Section 7 with a direction to the future works.

2 Related works

Many papers exist in the literature of wireless networks, which consider the fairness of single-hop flows. Luo et al. proposes fair bandwidth sharing among the contending flows that maximizes the throughput using spatial reuse [11]. Vaidya et al. proposes the distributed

[1] In this paper, we use the term rate to indicate the rate of a flow of unit weight. Therefore, the terms rate and unit-flow rate are used interchangeably.

fair scheduling (DFS) protocol [12], where a node sets its backoff counter based on the finish tag of its next packet. In DFS, a virtual clock is updated by overhearing, and the final tag is computed based on that, and the packets are transmitted based on the final tag. An AIMD-based rate control mechanism is proposed in [5], which synchronizes the rate update of the contending nodes by jamming the channel with a radio signal for an extended period of time. Furthermore, proportional fairness is provided by increasing the rate of the flows based on weights of the flows. However, the jamming signal imposes a huge control overhead, and the protocol is proposed for single-hop wireless networks.

In contrast, an extensive research has been done to improve the performance of TCP in wireless and ad-hoc networks [13, 14]. Mostly these works differ in identifying the losses, and accordingly controlling the TCP window size. A more recent work, WCP [6], considers the congestion in multihop wireless networks as a neighborhood phenomenon, where a congested node shares congestion information with its two-hop neighbors. This ensures that each flow passing through the congested neighborhood gets the fair share of the bottleneck bandwidth. However, the congestion sharing enforces every flow in the neighborhood of the congested node to decrease its rate, even though the flow uses a different congested region to route its traffic. Therefore, the rates of the flows are determined based on the mostly loaded congested region, and the limited wireless resource is kept underutilized. Furthermore, in multipath routing or with multiple base stations in the network, this affects the network throughput severely. Moreover, WCP only considers the intra-mesh traffic.

Another class of works focuses on bandwidth management in multihop wireless networks, and controls the rates of the flows to provide a max-min fairness. These protocols mainly aim to achieve end-to-end fairness in wireless ad-hoc networks [15]. However, the achieved throughput of the flows are far below the max-min achievable rate. A recent paper proposes GMP [4], a global max-min fairness protocol for multihop wireless networks. GMP uses separate queues at the intermediate nodes for all the flows with the same destination. However, if the majority of the traffic are Internet traffic, the idea of using separate queues for different destinations becomes useless because most of the packets are forwarded toward the base station. Furthermore, each link estimates the normalized rate for a flow based on the set of clique that the link shares. The rate of a flow is set by sending an explicit control packet periodically based on the intermediate links in the path of the flow. Therefore, the control overhead is very high.

We propose a rate control mechanism at layer 2, which controls the data injection rates of the locally connected flows and the forwarding rates of the transit flows at any node. A rate of a unit-weight flow is associated with each link, which determines the rates of the flows. Therefore, CFRC aims at converging the rates of the links to the fairness with a view to maximize the network throughput.

3 System model and assumptions

We consider an IEEE 802.11s-based WMN which is primarily used as an access network. The network is connected to the Internet through one or more mesh point portals (MPPs). We assume there are N mesh nodes (i.e., mesh point (MP), mesh access point (MAP), and MPP) that form the mesh network, and the i-th mesh node is denoted as n_i. The clients are connected with the MAPs using the existing IEEE 802.11 WLAN.

Data packets generated (by the clients) within the mesh network go through the MPPs to the external world, and vice versa. Even the WMN is used as an access network, there might exist traffic for which both the end points are within the WMN. It is expected that the nodes within the WMN use multipath routing to forward the data packets.

We assume that the mesh nodes are static, and therefore, the topology of the network does not change frequently. Multiple MPPs are connected by an external network so that they can exchange their data. We assume that communication between mesh clients and MAP, and between mesh entities are performed in different frequency channels. A link exists between two nodes, if one node can send data to the other. The nodes associated with a link are termed as the sender and the receiver, where the sender transmits the packet. We denote the link between nodes n_x and n_y as $l_{x,y}$, if n_x is the sender, and as $l_{y,x}$ otherwise.

The set of data packets from a particular source (i.e., client) toward a particular destination is defined as a *flow*. The set of flows associated with n_i (i.e., flows generated from the clients connected to n_i) is denoted as F_i. The j-th flow of n_i is denoted as $f_{i,j}$, where $j \in F_i$. A flow is said to be *upstream flow*, if it is originated inside the WMN, and the destination is outside the WMN. In contrast, a flow is said to be *downstream flow*, if its source is outside the WMN, and the destination is inside the WMN. A weight is associated with each flow, which indicates the importance of the flow. A larger weight means the flow is more important, and deserves a proportionally higher rate.

The set of nodes from which node n_i receives data packets (for the upstream, downstream, and intra-mesh flows) are defined as the upstream nodes, and denoted as U_i. Note that $U_i = U_i^u \cup U_i^d$; where, U_i^u denotes the set of upstream nodes for the upstream flows, and U_i^d denotes the set of upstream nodes for the downstream flows. In contrast, the set of nodes to which node n_i forwards the packets are defined as downstream nodes, and denoted as D_i. Note that $D_i = D_i^u \cup D_i^d$; where D_i^u denotes the set of downstream nodes for the upstream flows and D_i^d denotes the set of downstream nodes for the downstream flows. In contrast, we consider a link as upstream link, if it forwards the packets of upstream flows, and a link as downstream link, if it forwards the packets of the downstream flows.

4 Congestion-aware fair rate control

4.1 Overview

CFRC aims at providing a fair network access at layer 2 in an IEEE 802.11s-based WMNs. It controls the rates of the locally connected flows and transit flows at each node. A separate queue is maintained for each of the local flows, and packets of the flows are injected at the achievable fair rates, which depends on the link used as the first-hop of the flows.

In CFRC, each outgoing link probes the rate of a flow with unit weight, and assigns this rate to all the locally connected flows forwarded in that link. The rates of the links are updated periodically using AIMD. A bottleneck link enforces a rate-bounded backpressure to all links that use the bottleneck link to forward their packets. The backpressure mechanism synchronizes the rate decrements of the links that use the same bottleneck, and the rates of the links converge to fairness. This also allows the links with different bottleneck to obtain the maximum achievable rates, and increases the network throughput.

However, if the links obtain the rates based on their respective downstream bottleneck links, flows using different bottleneck regions might achieve different throughput. Note that existing mechanisms restrict the flows that uses the lightly loaded region of the network to increase their rate, and achieve the fairness with a reduced throughput. In contrast, balancing the loads among different network regions might achieve both the fairness and high network throughput. Therefore, CFRC's rate control mechanism integrates the load balancing with the routing mechanism, and achieves an increased network throughput as compared to the existing mechanisms with the desired fairness.

4.2 Source data injection rate control

The data injection rate of a flow is controlled by the node (i.e., source node) to which the flow is connected. The source nodes for the upstream flows and the intra-mesh flows are the MAPs. In contrast, the source nodes for the downstream flows are the MPPs. Existing mechanisms control the injection rates of the flows in different ways which include controlling the rate of the flows individually [5], maintaining a single rate for all the flows with the same destination [4], or maintaining a rate for each node and using this rate for all the flows connected at the node [6]. To achieve the fairness, the injection rates of the flows should depend on the capacity of the links used to forward the flow [16]. However, the capacity of a link depends on its own load, the loads of the contending links, and the complex interference imposed by the links, which make the capacity estimation as a non-trivial task.

Therefore, CFRC assigns the injection rates of the flows in a different way. Instead of finding the capacity of the links, it maintains a rate for each of the outgoing links. The rate associated with a link is used to set the injection rate of the flows that use the link as the first hop. More specifically, a source node controls the injection rate of each of the locally connected flows forwarded in a link based on the maintained rate of the link. However, CFRC needs to control the injection rates of the flows proportional to their weights for providing weighted fairness. Therefore, the rates of the links are maintained for a certain weight, and the injection rates of the flows are set accordingly based on the weight of the flows and the weight for which the links' rate are maintained. CFRC assumes the rate of each of the links is maintained for a flow with unit weight (i.e., if a flow has weight 1, then the injection rate of the flow is exactly equal to the maintained rate of the first-hop link's rate), referred to as *unit-flow rate*. Therefore, the injection rate of a flow is the product of its weight and the unit-flow rate of the first-hop link. Furthermore, the intra-mesh flows might not use the bottleneck link(s) used by the Internet flows. The intra-mesh flows might achieve a higher rate than the Internet flows. Therefore, each link maintains separate rates for the Internet and the intra-mesh flows.

Let $r_{i,x}^e$ and $r_{i,x}^i$ denote the rates of the Internet and the intra-mesh flows for the link $l_{i,x}$, respectively. For brevity, however, we use $r_{i,x}$ as the rate of the link $l_{i,x}$ to indicate the rates of both the Internet and intra-mesh flows. The data injection rates of flows f_{i,j_1} and f_{i,j_2}

(where f_{i,j_1} and f_{i,j_2} are Internet and intra-mesh flows, respectively) originated at n_i, and forwarded using the link $l_{i,x}$, are respectively given by

$$s_{i,j_1} = w_{i,j_1} \times r^e_{i,x} \quad (1a)$$

$$s_{i,j_2} = w_{i,j_2} \times r^i_{i,x}, \quad (1b)$$

where s_{i,j_1} and s_{i,j_2} are the data injection rates of f_{i,j_1} and f_{i,j_2}, respectively; w_{i,j_1} and w_{i,j_2} are the weights of the flows, respectively.

Finally, a fixed size separate queue is used for each locally connected flow at the source node. The packets from the queues are serviced at a rate exactly equal to the data injection rates of the flows. If the queue associated with a flow is full, and a new packet of the flow is arrived, the newly arrived packet is dropped. The packet dropping at the source node ensures that the packets of a flow do not waste the network capacity if the data generation rate is higher than the achievable fair rate. In contrast, the congestion-aware rate control ensures that packets usually do not loss at the intermediate nodes due to queue overflow. Therefore, the TCP sources update their rate based on the dropped packets at the source node.

4.3 Congestion-aware rate update

CFRC uses a relatively simple method to detect the incipient congestion, and to measure the congestion level of each link. Each node maintains a separate queue for each of the outgoing links. If the average queue size of any link exceeds a predefined threshold, CFRC assumes that the link is congested. In the literature of wireless networks, there exist many other techniques to detect the congestion which include channel loading around a node [17, 18], average number of transmission attempts to forward a packet [19], average time to recover the lost packets [20], ratio of the forwarding and reception rates [21], or a combination of these mechanisms. Like [6], we choose to use the queue occupancy as a measure of congestion level, because (1) it detects the congestion sufficiently well in wireless networks [22, 23], and (2) it allows to detect the congestion level for each link individually, if a separate queue is used for each link.

The average queue size of the link $l_{i,x}$ of the node n_i is measured using exponentially weighted moving average (EWMA), and is given by

$$\bar{q}_{i,x}(cur) = (1-\alpha)\bar{q}_{i,x}(prev) + \alpha q_{i,x}, \quad (2)$$

where $\bar{q}_{i,x}(cur)$ and $\bar{q}_{i,x}(prev)$ are the current and the previous average queue size of $l_{i,x}$, respectively; $q_{i,x}$ is the instantaneous queue size, and α is the smoothing parameter. Every time a node inserts a packet into the queue, it updates the average queue occupancy. If the value of $\bar{q}_{i,x}(cur)$ is greater than a threshold, q_{thsld}, the link is assumed as congested, otherwise it is not congested.

To obtain an efficient rate maintaining the capacity constraint, CFRC periodically updates the rate associated with each link using AIMD. If an outgoing link of a node is not congested, the node periodically increases the rate of the link. The rate of $l_{i,x}$ is increased by n_i using

$$r_{i,x}(t) = r_{i,x}(t - T_{\text{inc}}) + \delta, \quad (3)$$

where $r_{i,x}(t)$ and $r_{i,x}(t - T_{\text{inc}})$ are the rates of $l_{i,x}$ at time t and $(t - T_{\text{inc}})$, respectively; T_{inc} is the *rate update interval*, and δ is the amount of additive rate increase. If the link is congested, and the duration of time since the last rate decrement is greater than a *rate decrement threshold*, T_{dec}, the rate of the link is decreased by

$$r_{i,x} = r_{i,x}/2. \quad (4)$$

In wired networks, AIMD-based rate update provides both the rate efficiency and fairness. Due to the many-to-one traffic in WMN, and the shared nature of the wireless medium, links closer to the MPP(s) congested more frequently than a distant link. Therefore, the links do not decrease their rate synchronously, and the rate of the links might not converge. The links that decrease their rates more frequently than others, achieve a lower rate. As a result, the flows achieve unfair throughput depending on their location of attachment (i.e., the link which is used as the first hop). In Sections 4.4 and 4.5, we explain how the rates of the links in CFRC converge to the steady-state rate.

4.4 Rate-bounded backpressure

The AIMD-based rate control in wireless networks requires to share the congestion information to synchronize the rate decrement and achieve the fairness [22]. To synchronize the rate decrement of the neighboring nodes, a jamming-based channel blocking mechanism is proposed in [5]. Congestion sharing among the two-hop neighbors is considered in [6] which forces to decrease the rates synchronously, and ensures the fairness of the flows passing through the bottleneck link (and, its neighboring links).

However, if a neighboring link uses a different bottleneck, enforcing that link to synchronize the rate decrement might restrict it to achieve a higher rate. If the rate decrement is synchronized without considering the actual paths used by the links, the achievable rates of the links are determined based on the mostly loaded bottleneck link, and the link toward the lightly loaded network region will be restricted to obtain a higher rate. Even when a link uses different bottleneck or different MPP, it is restricted to utilize the network capacity. Therefore, we find that congestion in multihop wireless networks is path-dependent. The links that route their traffic via the same bottleneck link need to synchronize their rate decrement, and converge to the same rate. CFRC uses a *rate-bounded* backpressure which synchronizes the rate decrement based on the *average forwarding rate* (AFR) of a link or node, and AFR of the downstream bottleneck link.

The AFR of a link is defined as the average of the instantaneous rate of the link. Each node updates the AFR of an outgoing link using EWMA, when the instantaneous rate of the link is updated. The AFR of $l_{i,x}$, denoted as $\bar{r}_{i,x}$, is updated using $\bar{r}_{i,x} = (1-\beta)\bar{r}_{i,x} + \beta r_{i,x}$, where β is a tuning parameter. We set the value of β based on simulation. The AFR of the node n_i, denoted as $\bar{r}_i$, is

$$\bar{r}_i = \frac{\sum_{x=1}^{D_i} \bar{r}_{i,x} w_{i,x}}{w_i}, \tag{5}$$

where $w_{i,x}$ and w_i are the total weight of the flows passing through $l_{i,x}$ and n_i, respectively. When n_i forwards a packet to n_x in the link $l_{i,x}$, it includes the AFR of the link in the packet, and n_x returns its AFR in the ACK. The *rate-bound* enforces a link to decrease its rate if its AFR is not less than the announced AFR of the receiver, when the receiver announces the backpressure.

CFRC uses a three-bit flag to announce the backpressure, and the flag value '000' indicates that no backpressure is imposed. The upstream node of a link snoops the flag value from the forwarded packets of its downstream node. As mentioned in Section 4.3, a link decreases its rate when the queue size exceeds the threshold. This link is rate-bounded by its own AFR, and is a bottleneck link. The sender of the bottleneck link announces a *direct backpressure* (DBP) with a value '001' in every outgoing packet in that link, when the queue size exceeds the threshold, and it decreases the rate. Consider Fig. 1, where the link $l_{3,4}$ is assumed as the bottleneck link. The rate-bound in DBP ensures that the AFRs of all the incoming links of a node are not greater than the AFR of the node. Therefore, links $l_{2,3}$, $l_{11,3}$, and $l_{14,3}$ decrease their rate when they hear the DBP from n_3, if their AFR is not less than the AFR of n_3. To converge the rate of both the upstream and downstream flows, $l_{4,3}$ also decreases its rate when it hears the DBP, if its AFR is not less than the

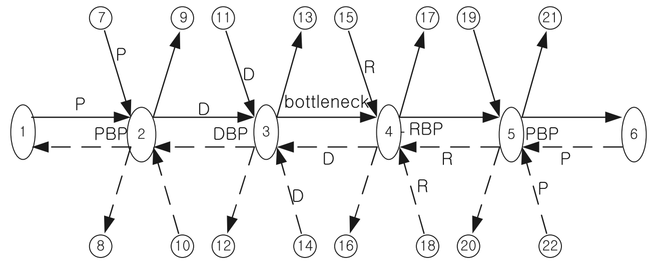

Fig. 1 Rate-bounded backpressure in CFRC. The upstream links are shown in *solid lines*, and the downstream links are shown in *dotted lines*. We assume that $l_{3,4}$ is the bottleneck link. Node n_3 enforces the DBP in every packet in $l_{3,4}$; whereas n_4 enforces the RBP in every packet forwarded in $l_{4,3}$. D, R, and P indicate that the links decrease the rate as a result of direct, reverse, and propagated backpressure, respectively

announced AFR. In contrast, the AFR of the incoming links at n_4 are required to be same as the AFR of $l_{3,4}$ to ensure fairness. Therefore, the receiver of the bottleneck link (i.e., n_4) announces a reverse backpressure (RBP) to all of its incoming links. When n_4 forwards a packet in $l_{4,3}$, it announces the congestion information '010', and includes the AFR of $l_{3,4}$. The AFRs of the incoming links at the receiver of the bottleneck links are rate-bounded by the announced AFR of $l_{3,4}$.

A rate-bounded link propagates the backpressure to its upstream links. When a link decreases its rate due to a rate-bound, it announces a *propagated backpressure* (PBP) with a value '011' in all outgoing packets. A PBP announced in an upstream link enforces a rate-bound (and hence, a rate decrement) to only an upstream link, and a PBP announced in a downstream link enforces a rate-bound to only downstream link. For example, the PBP announced in $l_{2,3}$ enforces a rate-bound to $l_{1,2}$, which decreases its rate and announces the PBP, if the AFR of the link is not less than the AFR of n_2. However, it does not enforce a rate-bound to $l_{10,2}$, because the traffic of $l_{10,2}$ does not use the bottleneck link. Accordingly, the PBP is propagated to all nodes that forward their data using the sender or the receiver of the bottleneck link.

We have considered that the majority of the flows are Internet flows, and the network becomes congested due to the Internet flows. Most of the links decrease their rates to achieve the fairness before their queue size exceeds the threshold, and do not utilize the full capacity. However, the destinations of intra-mesh flows are within the WMN, and they might not use the bottleneck link used by the Internet flows. Therefore, decreasing the rates of the intra-mesh flows with the Internet flows might keep the capacity underutilized. When a link decreases its rate due to a DBP or RBP, its flows are routed via the sender or receiver of the bottleneck link (whose queue size already exceeds the threshold), it decreases the rate of both of the Internet and intra-mesh flows. In contrast, when a link decreases its rate due to a PBP, it only decreases the rate of the Internet flows. Moreover, when the rates of Internet and intra-mesh flows in a link are not decreased together, its queue might exceed the threshold due to packets of the intra-mesh flows. Such a link is a bottleneck link for the intra-mesh flows. When the queue size of this link exceeds the threshold, it imposes a backpressure by announcing the congestion information with a value '101' in every outgoing packet, and the receiver of the link announces the RBP with the value '110'. Upon reception of the congestion information, the upstream links decrease the rate of the intra-mesh flows, but they do not propagate the congestion information, until their queues exceed the threshold.

4.5 Rate-equalizing load balancing

If multiple links from a single node (or two neighboring nodes) route their traffic via different loaded region toward a single MPP, the variation in load might allow the link, traversing the lightly loaded region, to achieve a higher AFR. The congestion sharing among the neighboring links, used in existing mechanisms (e.g., [5, 6]), restricts the high AFR link to increase the rate to achieve the fairness. The fair backpressure of CFRC enforces a rate-bound to the link with high AFR where the paths converge, and ensures the fairness. However, both these mechanisms are unable to obtain the maximum achievable throughput of the lightly loaded region. Furthermore, if the links are toward different MPPs, the backpressure mechanism cannot achieve the fairness because the links (or paths) do not converge; though it utilizes the available network capacity, and increases the throughput.

The reason behind the unfairness is the variations of the load of different paths. The available capacity of the network can be utilized with fairness, if the nodes forward their traffic in a balanced way. There are few routing protocols that considered the load balancing in route selection [24, 25]. However, the routing protocols usually react too slowly to the change of traffic of a path. Note that the variations in network loads occur when the flows join or leave. A balanced network becomes unbalanced when the flows leave, and CFRC aims to balance the network load by redirecting the flows toward the lightly loaded path. In contrast, when new flows join, CFRC aims to assist the routing protocol to select the outgoing link at a node based on the AFRs of the links.

In IEEE 802.11s-based WMNs, routing is performed at Layer 2, and to differentiate it from layer-3 routing, it is termed as path selection [2]. Hybrid wireless mesh protocol has been defined as the default path selection protocol for 802.11s-based WMNs which works in layer 2 [2]. The integration of routing functionalities at layer 2, therefore, allows a joint rate control and forwarding mechanism where both fairness and high throughput are achievable.

Furthermore, the majority of the flows in WMNs are Internet traffic where either the source or the destination of a flow is outside the WMN, and packets of the flows pass through an MPP to exit from the WMN or to enter into the WMN. More specifically, the upstream flows are directed toward an MPP, and it usually does not matter which path (or MPP, if there is more than one MPPs) is used to forward the packets of a flow. Therefore, the forwarding nodes can change the route of the flows, if the new route provides a higher throughput. In contrast, the MPPs can change the path of a flow if the changed path provide higher throughput, or a highly loaded MPP can redirect the traffic to a lightly loaded MPP.

CFRC aims at balancing the load of the contending links. The load balancing works in two ways. Firstly, if the links in a node have different loads, it balances the loads among the links by redirecting the flows. Secondly, if the links in neighboring nodes have different loads, and there is no active links between the nodes, it balances the loads of the links by creating a new link from the node with loaded link to the other (if the routing mechanism can find a valid path using the link). The loaded link forwards a fair amount of traffic (to be discussed shortly) toward the lightly loaded link. The newly added traffic share the capacity of the lightly loaded link, and the rates of the links converge toward the fair rate.

If a node has more than one outgoing link, it keeps a list of the links. The node stores the weight of the flows passing through the links, and the AFR of the links. The list is sorted according to the AFRs of the links. Each node balances the loads of the links so that the AFRs become fair. Therefore, each node periodically checks the AFRs of the links, and measures the amount of traffic changing (i.e., the weight of the flows that need to redirect) for the links with the highest and the lowest AFRs to make their AFRs equal to the AFR of the node. Since the link with high AFR forwards more packets, it will congest the downstream node earlier, and its AFR will be decreased. In contrast, the link with the lowest AFR now forwards less packets, it will be rate limited by the downstream node less frequently, and its AFR will be increased. As a result, both the links will converge to a single rate, which is the AFR of the node.

Let l_{i,x_1} and l_{i,x_2} denote the outgoing links of n_i with the maximum and the minimum AFRs, respectively. Note that the capacities achieved by the links are $w_{i,x_1} \times \bar{r}_{i,x_1}$ and $w_{i,x_2} \times \bar{r}_{i,x_2}$, respectively. The fair load balancing mechanism of CFRC changes the weight of the links so that the AFRs of the links converge toward the AFR of n_i. To balance the loads, the amount of weight increases/decreases for the links, denoted as w_{bal}, are given by

$$w_{\text{bal}} = \frac{w_{i,x_1} \times \bar{r}_{i,x_1}}{\bar{r}_i} - w_{i,x_1} = w_{i,x_2} - \frac{w_{i,x_2} \times \bar{r}_{i,x_2}}{\bar{r}_i}. \qquad (6)$$

Therefore, the load balancing does not change the achieved capacity of the links, rather, it changes the loads so that the flows that use the links will have fair rates. In contrast, when two neighboring links have different AFRs, the links with the lower AFR creates a path (if the routing protocol finds a valid path) toward the node which has the link with high AFR. CFRC uses the same method to find the amount of load forwarded in the new path as given in Eq. 6.

Finally, for the downstream flows, if there are more than one MPPs in the WMNs, different MPPs might have different downstream loads. Therefore, to achieve both the fairness and efficient rate, MPPs might need to redirect the downstream traffic. CFRC assumes that the MPPs periodically exchange their rates. If there is a variation in the rate of the MPPs, they redirect the traffic toward the lightly loaded MPP to balance the loads.

4.6 Queue management

As we have mentioned, CFRC maintains separate queues for the outgoing links, and the total weights of the links associated with the queues might not be same. Therefore, the queues require service rates proportional to their weights. CFRC uses a scheduler for the queues. The scheduler selects packets from head-of-line of the queues proportional to the forwarding rates of the queues, where the forwarding rates of the queues are given by the product of the total weight of the flows passing through the link associated with the queue and the AFR of the link.

In contrast, another scheduler selects the packets from the queues of the locally connected flows, and passes the packets to the queue associated with the first-hop of the flows. The service rates of the queues are exactly equal to the data injection rate of the flow. When the queue of a flow is full, newly arrived packets of the flow are dropped.

4.7 TCP rate control

TCP sources use AIMD rate control, and update the rates based on packet loss in the intermediate nodes. It is possible to adapt the CFRC's rate control with TCP in different ways, for example, TCP sources can set the rate based on the data injection rate of a flow. However, our study explores a mechanism keeping the TCP protocol unchanged.

The congestion-aware rate update of CFRC almost avoids the packet loss due to queue overflow, and we assume that all packet losses inside the WMN are due to link error or collision. Therefore, we need to keep the packet losses inside the WMN hidden to the TCP source, whereas packets lost at the queue of the flow or outside the WMN are known to the TCP source. Every source node keeps a copy of the transmitted packet till the ACK is received. The transport layer at the MPP, checks the TCP ACK (and identifies a duplicate ACK), and if there is a packet loss it finds whether the loss occurs inside or outside the WMN. For this, it maintains a list of forwarded (and unacknowledged) packets of all flows. If an entry is available in the list, the loss is outside the WMN, and the loss is inside the WMN otherwise. Then, it includes the ID of the packet in the ACK. When the source node gets the ID, it retransmits the packet, and keeps the TCP source unaware about the packet loss.

5 Parameter selection

The performance of CFRC (i.e., the convergence of rate of the links to the fairness, efficiency, and the long-term stability) depends on the careful selection of its parameters. We choose the values of the parameters based on the open-loop design of CFRC. Most of the traditional rate control mechanisms depend on the acknowledgment sent by the receiver, which makes the round trip time (RTT) as an obvious choice for the interval of rate adaption. However, CFRC controls the rate of the flows at the layer 2, and its operation is not coupled with the end-to-end feedback. Therefore, the parameter selection differs with the traditional mechanisms. Furthermore, considering the similarity of CFRC with the rate control mechanisms used for the wireless sensor networks (for example, [22] and [26]), the line of parameter selection of CFRC is biased by them.

5.1 Rate increment interval (T_{inc})

Each MPP maintains and propagates the rate increment interval. In order to maintain a fair rate by the links, and system stability, it is necessary that the links do not increase the rate repeatedly before the impact of a rate update reaches the bottleneck link. Therefore, it is essential that T_{inc} be large enough to ensure that the packets from the sources reach the bottleneck link, and the congestion feedback is propagated back to the source. Therefore, the value of T_{inc} depends on the network depth, and the link quality [26].

The MPP periodically measures the period of time requires to propagate a control information to every node in the network, and to get the feedback from them. For each measurement, it uses a new sequence number. Because the congestion information is propagated in every outgoing packet (transit and local) of a node, the MPP includes the sequence number in each outgoing packet (for an MPP, which is a packet of a downstream flow, if there is no active downstream flow it generates a control packet). Each node also includes the latest sequence number in every outgoing packet. In contrast, the impact of congestion reaches to the bottleneck link, when it receives the data packets from the sources. Therefore, each node acknowledges the reception of a new sequence number by including it in the packets of the locally connected flows. When the MPP receives the feedback from all the nodes, it considers the maximum of the periods as the current value of T_{inc}, denoted as $T_{inc}(\text{cur})$.

Finally, the average of the instantaneous values of T_{inc} is considered as the rate update interval, and is given by

$$T_{inc} = (1-\gamma)T_{inc} + \gamma T_{inc}(\text{cur}), \quad (7)$$

where γ is a smoothing factor, and its value is set based on extensive simulations. The MPP propagates this value to every node.

5.2 Queue threshold (q_{thsld}) and smoothing factor (α)

The relationship between q_{thsld} and α depends on the burst length, which determines the number of packets that a congested link can receive before being declared itself as congested. The relationship between these parameters are explored in [27]. If the allowable burst length is too short, a link will declare itself congested even when the network can deliver the traffic; whereas, if it is very large, the link might not announce the congestion even when the network is overloaded. Furthermore, a link might need to receive packets from the upstream nodes with a higher rate in a number of rate update intervals, even after announcing the congestion information (please see Section 5.3 for details). Therefore, it needs space in the queue to store these extra packets. Following the line of IFRC [22], we set the value of the burst length equal to half of the actual buffer size, which gives $\alpha = 0.001$ for $q_{thsld} = 8$ packets.

5.3 Rate increment intensity (δ)

The value of δ determines the intensity of the additive increase of AIMD. Therefore, it determines the stability and the speed of rate convergence. Node n_i increases the rate after every T_{inc} by an amount of δ if there is no congestion. In contrast, if the link is congested or rate limited, it decreases the rate to $r_{i,x}/2$. Let us assume the minimum, maximum, and average achievable rates of the link $l_{i,x}$ are $r_{i,x}(\text{min})$, $r_{i,x}(\text{max})$, and $r_{i,x}(\text{avg})$, respectively. Note that a link has the maximum rate just before it decreases the rate, and it has the minimum rate just after decreasing the rate. Furthermore, if the network status does not change, the link simply increases the rate from the minimum value by an amount δ after every T_{inc} period, eventually it reaches the maximum value, and continues the same cycle.

When the rate is above the average rate, the extra packets build up the queue. In contrast, when the rate is decreased, the rate is below the average rate, and the link forwards the queued packets. For stability, the amount of data transmitted above the average rate is required to be less than or equal to the unused capacity when the rate is below the average rate [22]. Therefore, the average rate of $l_{i,x}$ is given by

$$r_{i,x}(\text{avg}) \le \frac{r_{i,x}(\text{min}) + r_{i,x}(\text{max})}{2}. \quad (8)$$

And, due to the multiplicative decrease with a factor of $1/2$, we have $r_{i,x}(\text{max}) = 2 \times r_{i,x}(\text{min})$. Therefore, the average rate of the link becomes

$$r_{i,x}(\text{avg}) = \frac{3r_{i,x}(\text{max})}{4}. \quad (9)$$

If the value of δ is high, the rate might jump from the minimum value to the maximum value in a single step. In contrast, if δ is very small, it might take a long time for the rate convergence. Let us assume we need m

steps to reach the maximum value from the average value. Therefore, the value of δ for the link $l_{i,j}$ is given by

$$\delta = \frac{r_{i,j}(\max) - r_{i,j}(\min)}{2m} = \frac{r_{i,j}(\min)}{2m}. \tag{10}$$

In CFRC, the rate decrement information is propagated to the rate limited links within one rate update interval. However, even though the nodes decrease their rate simultaneously, the bottleneck link might receive packets from the upstream nodes at the higher rate. This happens because of the delay of the packets from the upstream links to the bottleneck link. If the network diameter is small or the bottleneck link is far away from the MPP, the bottleneck link receives packet from all the upstream links at the same rate. In contrast, if the network diameter is long or the bottleneck link is close to the MPP, the bottleneck link will receive packets at different rates due to the delay.

Let us first assume that the bottleneck link receives packets at its own rate from all the transit flows. When the link's rate is higher than the average achievable rate, the extra packets are queued. When the queued packets increase the average queue occupancy above the threshold, the link becomes congested, and decreases its rate. Let Q_0 denote the actual queue occupancy for which the average queue size exceeds the threshold.

The number of excess packets forwarded by a flow during m rate increment intervals, when its rate is above the average achievable rate, is $\sum_{i=1}^{m}(i\delta/p) = \frac{1}{2p}m(m+1)\delta$, where p is the size of a packet. The total weight of the flows passing through the bottleneck link $l_{i,j}$ is $w_{i,j}$. Therefore, the total number of packets stored at the queue, denoted as $Q_{i,j}$, when the rate is above the average achievable rate, and is given by

$$Q_{i,j} = \frac{1}{2p}m(m+1)\delta w_{i,j}. \tag{11}$$

Therefore, the condition for the congestion at $l_{i,j}$ is

$$Q_{i,j} > Q_0 \tag{12}$$

Combining Eqs. 10, 11, and 12 gives us the lower bound of m, which is given by

$$m > \frac{4pQ_0}{w_{i,j}r_{i,j}(\min)} - 1. \tag{13}$$

Let us consider the case, where the bottleneck link receives packets from the upstream links at a rate lower than its own instantaneous rate due to the packet delay. Assume that the bottleneck link receives packets from the distant upstream node at a higher rate for the next m^* rate increment intervals after the congestion notification. Furthermore, assume that $m^* \leq m$, i.e., at the time of congestion, the upstream link's rate is at least above the average rate. In the worst case, the transit flows might deliver packets at a higher rate for m update intervals after the rate decrement, and the queue size of the bottleneck link might continue to increase. The extra packets delivered by the transit flows should not exceed the actual queue size, denoted as B, to avoid buffer overflow. Therefore, the maximum value of m^* is given by

$$m^* \leq \frac{4p(B - Q_0)}{w'_{i,j}r_{i,j}(\min)} - 1, \tag{14}$$

where $w'_{i,j}$ is the total weight of the transit flows that are away from the bottleneck link, and experience the delay. Note that packets of all transit flows might not experience the delay. Therefore, the value of m^* depends on the specific network topology, and delay experienced by the transit flows. However, in CFRC, the maximum duration of the rate update interval is set in such a way, so that the maximum delay between the transmission of a packet from a source and the reception of it by the MPP is one rate update interval. In the worst case, the bottleneck link receives packets at a higher rate from the farthest node for 2 rate update intervals, which gives $m^* = 2$. Therefore, the impact of m^* is very less on the value of m, and the value of m is determined by Eq. 13. Finally, we set the value of m (and hence, the value of δ) based on extensive simulations.

5.4 Minimum rate decrement interval (T_{dec})

The minimum rate decrement interval restricts a link to repeatedly decrease its rate. Note that a congested link announces the rate decrement as long as it is congested or rate limited, though it decreases its rate only once. As we have mentioned in Section 5.3 that a link might receive packets from its upstream nodes for several update intervals due to the packet delivery delay. Therefore, its average queue size is less than the queue threshold after a number of rate update intervals. As a result, a link should not decrease its rate during this period again and again, because this will reduce the throughput. As we explained in Section 5.3, a congested link receives packets above the average rate for m^* rate update intervals. We intuitively assume that the congested link requires at least another m^* rate update intervals to forward these extra packets. However, we are a little conservative here, and set the value to be $(2m^* + 1)$, which gives the value of minimum rate decrement interval as $(2m^* + 1)T_{\text{inc}}$. In the simulation, we have found that most of the cases, the queue size of the congested link becomes less than the threshold within this period.

5.5 Total flow weight of a link

Every node includes the weight of the flows in the packets of the flows. The intermediate nodes maintain a flow table, which includes the flows forwarded by the nodes with their weights. Each node can easily calculate the total weight of the flows passing through the node, and forwarded in each outgoing link.

6 Performance evaluations

We have performed extensive simulations in ns-2 to evaluate the performance of CFRC. We have considered two different simulation scenarios. We have not considered the auto-rate control at the MAC layer, and used a fixed rate of 56 Mbps. We choose a fixed rate at the MAC layer, because the variation in transmission rates has a little impact on the performance of CFRC. A change in transmission rate of a link will change the value of the unit-flow rate of the link, and all the flows passing through the link will have the same proportional data injection rates. Therefore, the change in transmission rates of the links only changes the network throughput (and, the throughput of the flows), however, the throughput of the flows will remain proportional to their weight. An error-free channel is used for most of the simulations. However, we demonstrate the impact of channel error on CFRC using one simulation. For all the active flows, we have considered a fixed packet size of 1,000 bytes (Fig. 13 demonstrates the performance with two different packet-size). Other simulation parameters are given in Table 1. We have run each simulation for ten times to measure the average value, and we have run each simulation for 50 s. For all simulations (except the result shown in Fig. 8), we have considered that the weights of the flows are equal.

Table 1 Parameters used in simulation

Parameter	Value
Queue threshold (q_{thsld})	8 packets
EWMA weight (α)	0.001
EWMA weight (β)	0.15
EWMA weight (γ)	0.10
Rate increment intensity (δ)	0.768 packets
Queue size for a link (B)	256 packets
Queue size for a flow	20 packets
Packet size	1,000 bytes

We compare the performance of CFRC with WCP [6], and GMP [4]. We also demonstrate the performance where only TCP and UDP are used. Furthermore, we compare the performance with the achievable fair throughput for a certain network condition, which is defined for the Internet traffic with UDP traffic, where we gradually increase the sending rate of each flow, and measure the throughput. When the collective data rate exceeds the network capacity, packets drop due to excessive collisions, and the throughput of flows away from the MPP decreases. The maximum rate for which each flow achieves a 98% delivery ratio is defined as *achievable fair throughput* (AFT).

6.1 Scenario 1: WMN with single MPP

In this simulation scenario, we have considered a WMN with single MPP and 24 nodes. Three flows (upstream, downstream, and intra-mesh) are connected with each node. The destinations of the intra-mesh flows are randomly selected within the WMN. Figure 2 shows the average goodput achieved by the flows for different mechanisms. For AFT, we have gradually increased the data injection rate of the flows, and we measure the delivery ratio. As long as the collective data injection rate is below the network capacity, increasing the injection rate also increases the goodput of the flows. However, when the collective data injection rate is above the network capacity, goodput of flows connected to the nodes closer to the MPP increases only, whereas that of flows connected to the nodes far away from the MPP decreases. We consider the average goodput of the

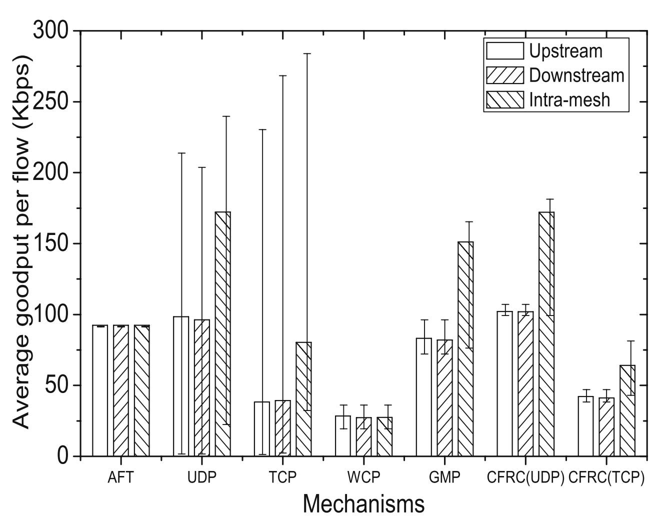

Fig. 2 Average goodput of the flows for different mechanisms

flows as the AFT, where every flow achieves a delivery ratio of more than 98%. Note that the AFT depends on the network topology, and the active flows in the network. For UDP, we have set the data rate twice the data rate of the AFT. This allows more packets from the closer nodes of the MPP, and less packets from the distant nodes. Though the average goodput increases a little, the difference between the maximum and the minimum goodput of the flows is very high. Like UDP, the maximum and minimum goodput of the flows in TCP is very high. The reason behind this is that the flows close to the MPP have shorter RTT, and achieve a higher throughput than the distant flows. WCP enforces strict fairness for all neighbors, and achieves a very low goodput. In contrast, GMP achieves lower throughput than AFT, due to its inefficient queue management and rate update. We have measured the goodput of CFRC for both the UDP and TCP flows separately. CFRC with UDP achieves a goodput higher than AFT and GMP. The efficient congestion propagation in CFRC allows the flows to achieve a higher goodput, and the load balancing equalizes the rates of different bottlenecks. As a result, CFRC outperforms both WCP and GMP.

Figure 3 shows the fairness index of the achieved goodput of the flows. We have used Jain's fairness index [28]. The AFT achieves the highest fairness index as expected. The fairness index is very low for both the UDP and TCP. WCP and GMP are designed to achieve the fairness, and they achieve this with a decreased goodput of the flows as compared to TCP and UDP, respectively. In contrast, CFRC increases the fairness without sacrificing the goodput of the flows. Even it achieves higher goodput due to the load balancing. Furthermore, like GMP, CFRC allows the intra-mesh flows to utilize the available network capacity, and the intra-mesh flows goodput are higher than that of the Internet flows. Therefore, the fairness index is lower for them. Figure 4 shows the average end-to-end delay of the packets for different mechanisms. The vertical lines show the maximum and minimum delays in different simulation runs.

Figures 5 and 6 show the average goodput of the individual flows in CFRC when all flows are UDP and TCP, respectively. The Internet flows achieve almost equal goodput in both cases, whereas the intra-mesh flows achieve relatively higher goodput depending on the destinations of flows. However, flows that use the

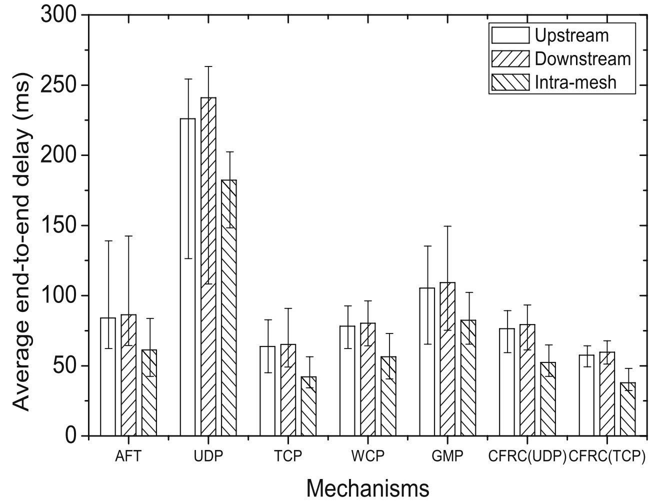

Fig. 4 Average end-to-end delay of the packets for different mechanisms

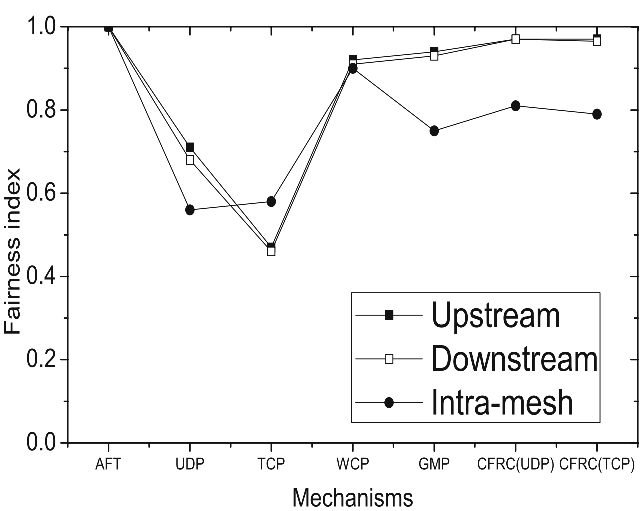

Fig. 3 Fairness index of the achieved goodput of the flows for different mechanisms

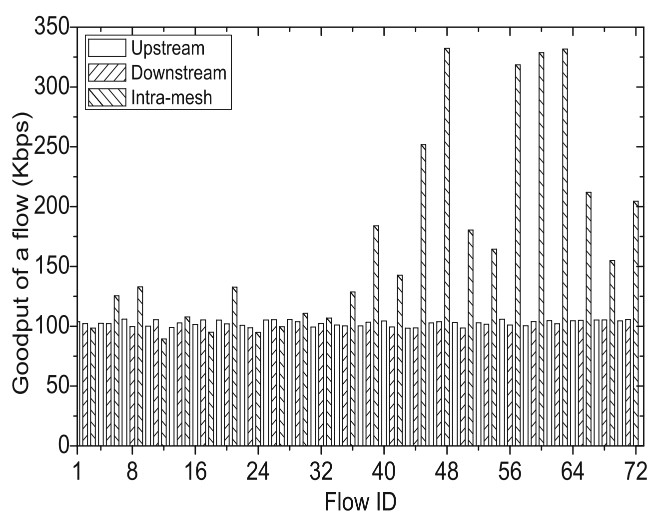

Fig. 5 Average goodput of the UDP flows in CFRC

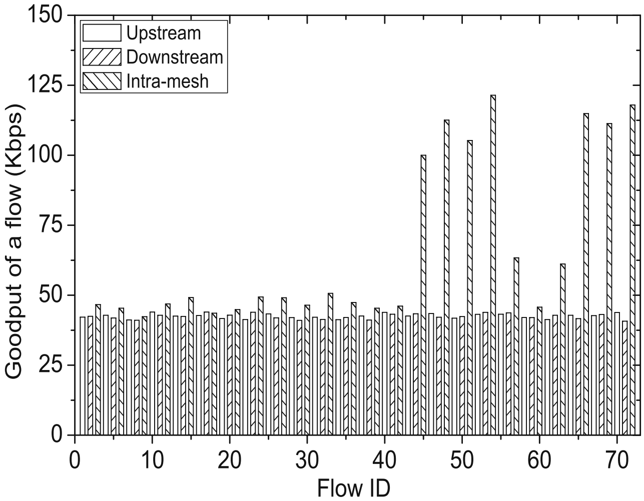

Fig. 6 Average goodput of the TCP flows in CFRC

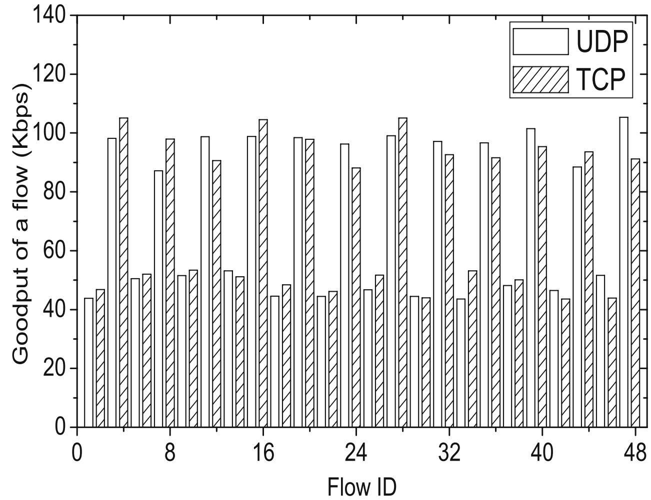

Fig. 8 Average goodput of the flows with different flow weights

bottleneck links used by the Internet flows achieve the same goodput as the Internet flows. Figure 7 shows the average goodput where we have used both the UDP and TCP flows simultaneously. The odd number of nodes use TCP for the upstream flows, UDP for the downstream flows, and vice versa for the even number of nodes. Since the rates of flows depend on the first hop of the flows, CFRC provides the desired fairness for both type of flows. Figure 8 shows the goodput of the flows for the same simulation setup used in Fig. 7, however, the flows connected with the nodes with odd ID have weight 1, and the flows connected with the nodes with even ID have weight 2. Since the packets of a flow is passed to the link queue based on its data injection rate, the flows achieve throughput based on their weights. This justifies that CFRC can provide weighted fairness.

Figures 9 and 10 show the instantaneous rate update, and the instantaneous queue occupancy of a bottleneck link for the time period of 50 to 100 s during the simulation run. When the queue size exceeds the threshold, the bottleneck link decreases the rate, and then linearly increases the rate again after every T_{inc} period. However, as mentioned in Section 5, the bottleneck link receives packet at a higher rate than the average sustainable rate, and the queue occupancy increases even after congestion notification. The queue occupancy decreases below the threshold after few update periods, and eventually it becomes almost empty, and again started to build up. Since there is no change in

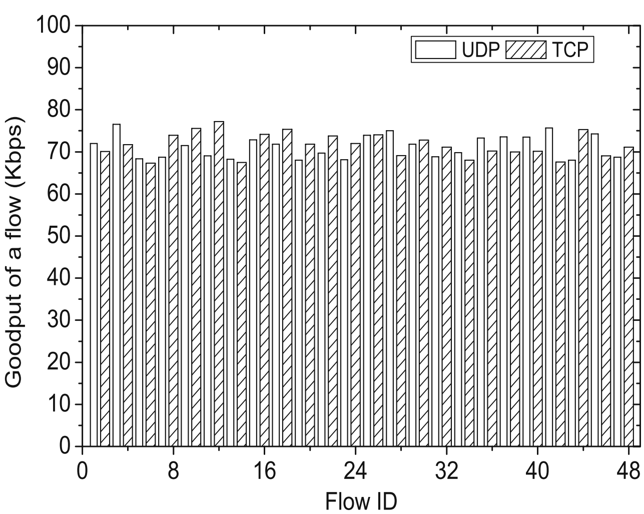

Fig. 7 Average goodput of the flows in CFRC when both TCP and UDP are used

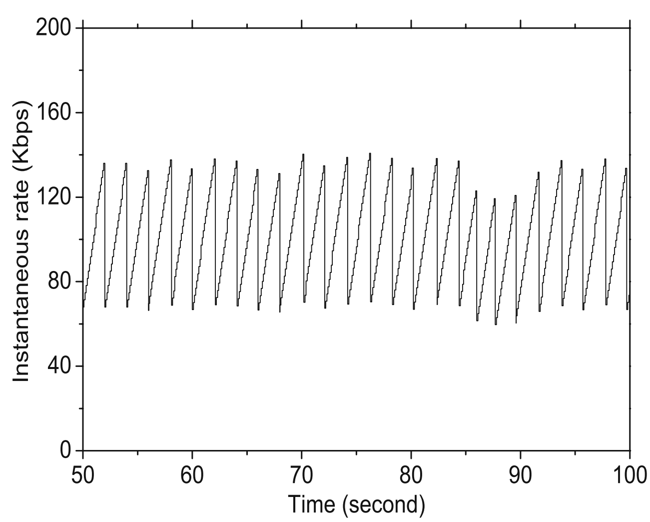

Fig. 9 Instantaneous rate update at the bottleneck link

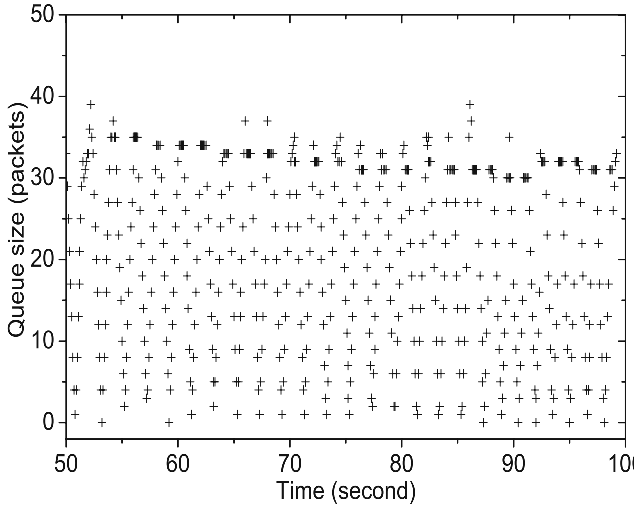

Fig. 10 Instantaneous queue size of the bottleneck link

the active flows in the network, the instantaneous rate of the bottleneck link maintains the same cycles for the whole period.

Figure 11 shows the average goodput achieved by the flows with different flow duration. We change the starting time of flows of different groups. There are four groups of flows. The first group is a set of upstream TCP flows connected to the nodes with odd ID, and the flows start at time zero. The second group is a set of downstream UDP flows connected to the nodes with even ID, and the flows start at time 10 s. The upstream UDP flows in the third group are connected to the nodes with odd ID, and start at time 20 s. Finally, a set of downstream TCP flows connected to the nodes with even ID, and start at time 30 s. All flows terminate at time 40 s. As the figure shows, when the number of active flows changes, only the goodput of the flows changes. Since the injection rates of the flows depend on the first hop of the flows, the achievable goodput is independent of the flow duration. The vertical max–min values show that the flows almost achieve the same goodput.

Figure 12 shows the goodput achieved by the flows with erroneous channel. We consider that 10% of the transmitted packets are lost due to channel error at the MAC layer. Though most of the lost packets are recovered by the MAC layer retransmissions, even then few packets are dropped. In case of the UDP flows, the dropped packets do not have any impact at the source. Some of the TCP versions might react due to the dropped packets, and decrease the window size. However, in CFRC TCP sources decrease their rates only when packets are lost due to queue overflow at the source nodes or outside the WMN. Therefore, there is almost no impact of channel error, and TCP flows achieve same throughput as the UDP flows.

Figure 13 shows the average goodput achieved by the flows with different packet sizes. Flows connected with nodes 1–24 have packet size of 1,000 bytes, and flows connected with nodes 25–48 have packet size of 500 bytes. Since CFRC controls the data injection rate of flows in bytes/sec, all of the flows achieve almost fair throughput. A large variation in packet size might hinder the short-term fairness of the flows, because flows with larger packet size might need to wait before forwarding the packets. However, variation in packet size will have very small effect on long-term fairness.

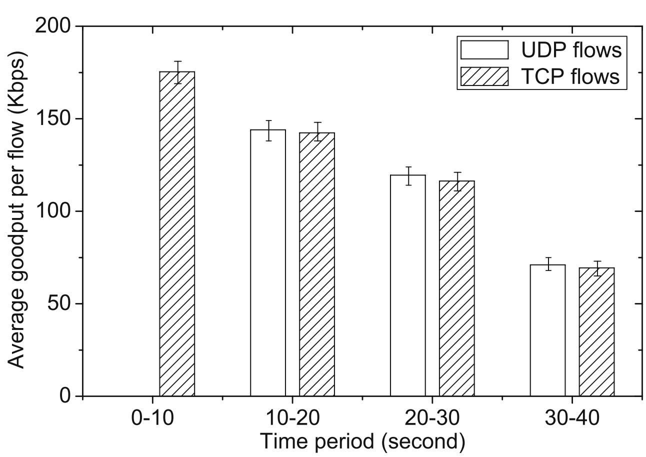

Fig. 11 Average goodput achieved by the flows with different flow durations with both UDP and TCP flows

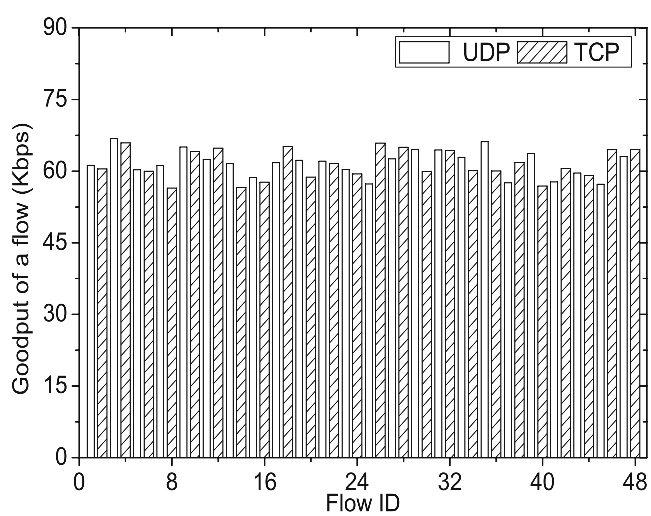

Fig. 12 Goodput of the flows with erroneous wireless link, we assume that 10% of the transmitted packets are lost at the MAC layer

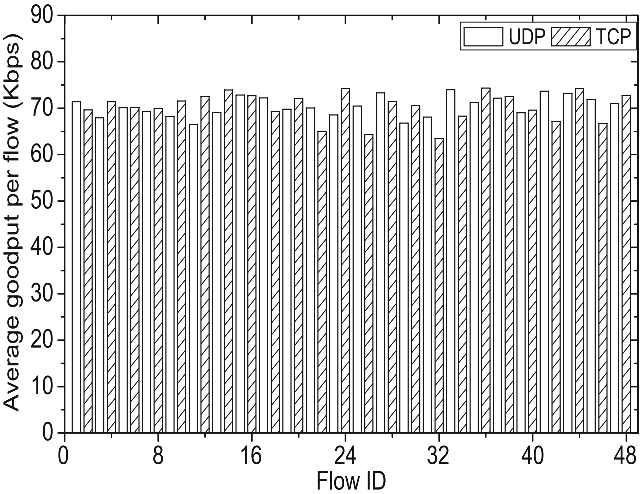

Fig. 13 Average goodput of the flows for CFRC with two different packet sizes

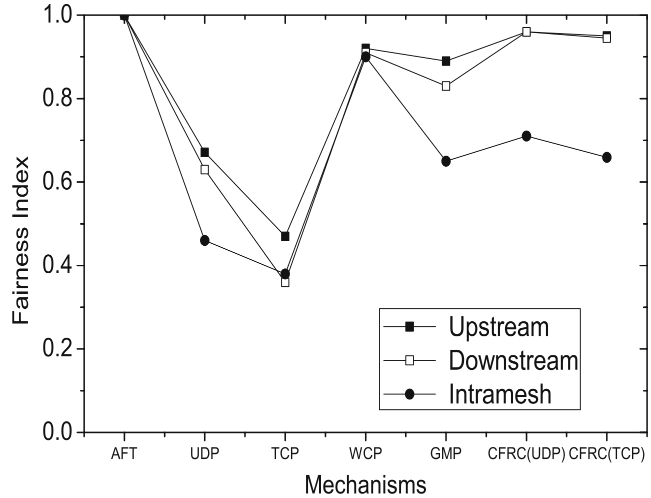

Fig. 15 Fairness index of the achieved throughput of the flows for different mechanisms with two MPPs and 48 nodes

6.2 Scenario 2: WMN with multiple MPPs

In scenario 2, we have considered a WMN with two MPPs, and 48 nodes. Furthermore, clients are not attached to all the nodes, because some of the nodes only work as MPs. There are 72 flows in the network, and the sources are randomly selected. Flows are equally divided into upstream, downstream, intra-mesh flows, and the destinations of the intra-mesh flows are selected randomly.

Figure 14 shows the average goodput achieved by a flow in different mechanisms. Like the single MPP WMN, the AFT shows the fairly achieved goodput of a flow, which is the fair optimum goodput for a given network setup. UDP achieves a slightly higher throughput than AFT, with a very high variation between the lowest and highest goodput among the flows. In contrast, GMP tries to achieve the fairness like AFT, but due to the huge control overhead it achieves a much lower goodput than AFT. However, when UDP is used for CFRC, it not only achieves a better fairness than GMP, but also the average goodput of the flows is higher than AFT. The load balancing toward different MPPs increases the overall throughput. When TCP is used as the transport layer, the average goodput is much lower than AFT, and the variation in goodput among the flows is very high. WCP tries to achieve the goodput achieved by TCP with fairness, but the average goodput

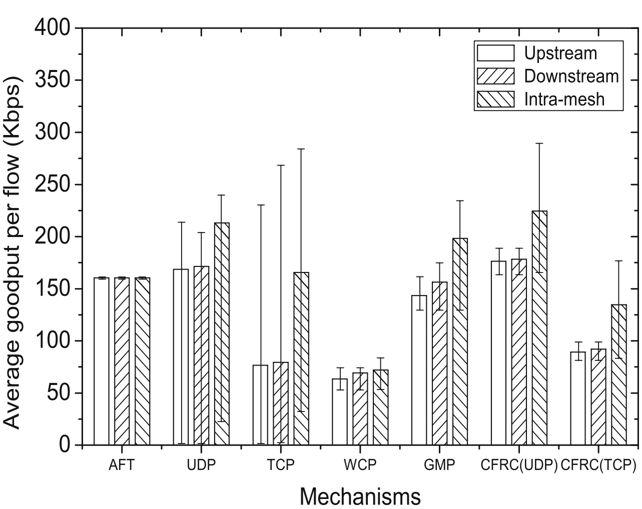

Fig. 14 Average goodput of the flows for different mechanisms with two MPPs and 48 nodes in the network

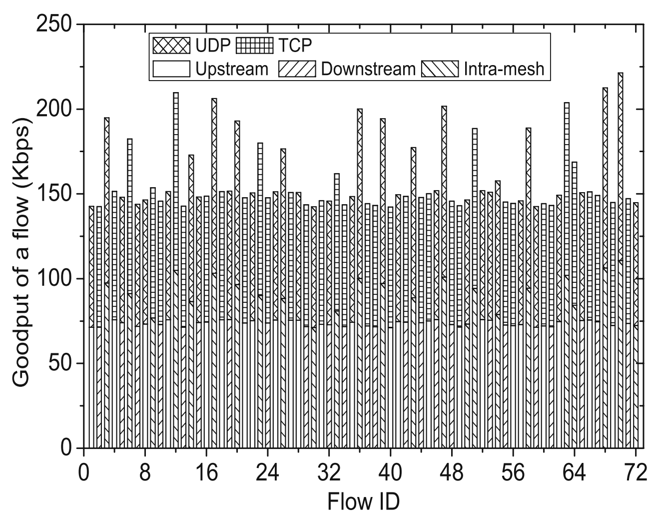

Fig. 16 Average goodput of the flows for CFRC with two MPPs and 48 nodes in the network

is lower than TCP due to very aggressive congestion sharing. In contrast, CFRC achieves a higher average goodput than TCP with a high degree of fairness. The fairness index of different mechanisms is shown in Fig. 15.

Figure 16 shows the goodput of the individual flows for CFRC. Among the flows, half of the flows are UDP flows and the rest are TCP flows. In the figure, the lower portion of the bar represents the flow types (upstream, downstream, and intra-mesh), whereas the upper portion indicates the transport layer used for the flows. As shown in the figures, both the upstream and downstream flows fairly share the network capacity; whereas the intra-mesh flows achieve goodput depending on the paths used by the flows.

7 Conclusion

In this paper, we have proposed a fair rate control mechanism for WMNs, which works at layer 2, and fairly controls the rate of the flows independent of the flow types. CFRC aims at providing the fairness without sacrificing the goodput achieved by the flows, and the path-dependent and rate-bounded backpressure facilitate to achieve this. The load-balancing ensures the fairness of flows that different bottlenecks, and ensures the utilization of the network capacity. Furthermore, the association of the flow rates with the rates of the first hops of the flows ensures that the goodput achieved by the flows does not depend on the flow duration. Finally, we have used simulation to evaluate the performance of CFRC, and the simulation results demonstrate the effectiveness. In the future, we like to find the rates of the flows based on the estimated capacities of the links.

Acknowledgements This work was supported by a grant from the Kyung Hee University in 2009. "(KHU-20090437)". Dr. CS Hong is the corresponding author.

References

1. Akyildiz IF, Wang X, Wang W (2005) Wireless mesh networks: a survey. Comput Netw ISDN Syst 47(4):445–487
2. IEEE 802.11s (2007) Task group, draft amendment to standard for information technology-telecommunications and information exchange between systems–local and metropolitan area networks-specific requirements—part 11: wireless lan medium access control (MAC) and physical layer (PHY) specifications: amendment IEEE p802.11s/d1.06: mesh networking
3. Draves R, Padhye J, Zill B (2004) Routing in multi-radio, multi-hop wireless mesh networks. In: MobiCom '04: proceedings of the 10th annual international conference on mobile computing and networking. ACM, New York, NY, USA. pp. 114–128
4. Zhang L, Chen S, Jian Y (2008) Achieving global end-to-end maxmin in multihop wireless networks. In: Distributed computing systems, 2008. ICDCS '08. The 28th international conference on. pp. 225–232
5. Jian Y, Chen S (2008) Can CSMA/CA networks be made fair? In: MobiCom '08: proceedings of the 14th ACM international conference on mobile computing and networking. ACM, New York, NY, USA. pp. 235–246
6. Rangwala S, Jindal A, Jang KY, Psounis K, Govindan R (2008) Understanding congestion control in multi-hop wireless mesh networks. In: MobiCom '08: proceedings of the 14th ACM international conference on mobile computing and networking. ACM, New York, NY, USA. pp. 291–302
7. Ee CT, Bajcsy R (2004) Congestion control and fairness for many-to-one routing in sensor networks. In: SenSys '04: proceedings of the 2nd international conference on embedded networked sensor systems. ACM. pp. 148–161
8. Gambiroza V, Sadeghi B, Knightly EW (2004) End-to-end performance and fairness in multihop wireless backhaul networks. In: MobiCom '04: proceedings of the 10th annual international conference on mobile computing and networking. ACM, New York, NY, USA. pp. 287–301
9. Faccin S, Wijting C, Kenckt J, Damle A (2006) Mesh WLAN networks: concept and system design. IEEE Wirel Commun 13(2):10–17
10. Wang X, Lim AO (2008) IEEE 802.11s wireless mesh networks: framework and challenges. Ad Hoc Networks 6:970–984
11. Luo H, Cheng J, Lu S (2004) Self-coordinating localized fair queueing in wireless ad hoc networks. IEEE Trans Mob Comput 3(1):86–98
12. Vaidya N, Dugar A, Gupta S, Bahl P (2005) Distributed fair scheduling in a wireless LAN. IEEE Trans Mob Comput 4(6):616–629
13. Biaz S, Vaidya N (2005) "De-randomizing" congestion losses to improve TCP performance over wired-wireless networks. IEEE/ACM Trans Netw 13(3):596–608
14. Liu J, Singh S (2001) ATCP: TCP for mobile ad hoc networks. IEEE J Sel Areas Commun 19(7):1300–1315
15. Li B (2005) End-to-end fair bandwidth allocation in multi-hop wireless ad hoc networks. In: ICDCS '05: proceedings of the 25th IEEE international conference on distributed computing systems. IEEE Computer Society, Washington, DC, USA. pp. 471–480
16. Dukkipati N, Kobayashi M, Zhang-Shen R, McKeown N (2005) Processor sharing flows in the internet. In: IWQoS
17. Xu K, Gerla M, Qi L, Shu Y (2003) Enhancing TCP fairness in ad hoc wireless networks using neighborhood red. In: MobiCom '03: proceedings of the 9th annual international conference on mobile computing and networking. ACM, New York, NY, USA. pp. 16–28
18. Wan CY, Eisenman SB, Campbell AT (2003) CODA: congestion detection and avoidance in sensor networks. In: SenSys '03: proc. of the 1st international conf. on embedded networked sensor sys. ACM. pp. 266–279
19. Fu Z, Luo H, Zerfos P, Lu S, Zhang L, Gerla M (2005) The impact of multihop wireless channel on TCP performance. IEEE Trans Mob Comput 4(2):209–221
20. Paek J, Govindan R (2007) RCRT: rate-controlled reliable transport for wireless sensor networks. In: SenSys '07: proceedings of the 5th international conference on embedded networked sensor systems. ACM. pp. 305–319

21. Wang C, Li B, Sohraby K, Daneshmand M, Hu Y (2007) Upstream congestion control in wireless sensor networks through cross-layer optimization. IEEE J Sel Areas Commun 25(4):786–795
22. Rangwala S, Gummadi R, Govindan R, Psounis K (2006) Interference-aware fair rate control in wireless sensor networks. In: SIGCOMM '06: proceedings of the 2006 conference on applications, technologies, architectures, and protocols for computer communications. ACM. pp. 63–74
23. Hull B, Jamieson K, Balakrishnan H (2004) Mitigating congestion in wireless sensor networks. In: SenSys '04: proceedings of the 2nd international conference on embedded networked sensor systems. ACM, New York, NY, USA. pp. 134–147
24. Ma L, Denko M (2007) A routing metric for load-balancing in wireless mesh networks. In: Advanced information networking and applications workshops, 2007, AINAW '07. 21st international conference on, vol 2. pp. 409–414
25. Islam M, Alam M, Hamid M, Hong C, Lee S (2010) EFT: a high throughput routing metric for IEEE 802.11s wireless mesh networks. Ann Télécommun 65(5–6):247–262. doi:10.1007/s12243-009-0130-1
26. Sridharan A, Krishnamachari B (2009) Explicit and precise rate control for wireless sensor networks. In: SenSys '09: Proceedings of the 7th ACM conference on embedded networked sensor systems. ACM, New York, NY, USA. pp. 29–42
27. Floyd S, Jacobson V (1993) Random early detection gateways for congestion avoidance. IEEE/ACM Trans Netw 1(4): 397–413
28. Jain R, Chiu D, Hawe W (1984) A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Tech. Rep. TR-301, DEC Research