RESEARCH

Open Access

Joint QoS provisioning and congestion control for multi-hop wireless networks

Weiqi Chen[1], Quansheng Guan[1*], Shengming Jiang[2], Quanxue Guan[1] and Tiancheng Huang[1]

Abstract

The capability of quality-of-service (QoS) provisioning is of particular importance for multi-hop wireless networks when the real-time applications boost in current days. The scheduling and delivery of data packets in a deficient method may probably cause network congestion, which will in turn decrease the capability of QoS provisioning in the network. To this end, we propose a joint QoS provisioning and congestion control scheme for multi-hop wireless network in this paper based on our previous works of Differentiated Queueing Service (DQS) and Semi-TCP, which provide per-packet granular QoS and carry out efficient hop-by-hop congestion control, respectively. While DQS and Semi-TCP are studied separately, we investigate the arising issues in the joint scheme and propose possible solutions accordingly, including a fast estimation of the latest departure time, a method to handle overdue packets, and an adaptive ACK scheme, as well as the design of a shared database cross-layer architecture for the implementation in the protocol stack. Simulation results show that our proposal improves the network performance in terms of goodput, delivery ratio, and end-to-end delay significantly, particularly in the scenario of mobile users. Our discussion and simulation results both indicate that the proposed joint scheme is flexible and adaptive to the dynamic multi-hop wireless network environment.

Keywords: Multi-hop wireless networks, QoS provisioning, Congestion control, Cross-layer design

1 Introduction

Multi-hop forwarding via wireless links has some beneficial advantages over single-hop transmissions. It may extend the coverage of cellular networks and wireless local area networks, where the single-hop wireless communications usually happen between the base station and the end users. Forwarding via multiple short and strong wireless links may also save the energy and achieve higher quality of transmissions compared to transmissions via single-hop long links. This kind of multi-hop wireless network has been widely developed in the forms of mobile ad hoc networks [1], wireless sensor networks [2], wireless mesh networks [3], and delay tolerant networks (DTNs) [4, 5]. Multi-hop relaying is also adopted by 3GPP [6], 5G networks [7], and IEEE 802.16j [8].

There are generally two types of data traffic in the network: delay-sensitive and non-delay-sensitive traffic. Delay-sensitive traffic has stringent requirements, which

needs quality-of-service (QoS) provisioning in the network. Non-delay-sensitive traffic has elastic data rate, which needs flow and congestion control to adapt network conditions. Most network protocols in multi-hop wireless networks inherit from those in wired communication networks. However, the end-to-end performance becomes problematic in multi-hop wireless networks. The end-to-end behaviors of QoS provisioning and congestion control over multiple hops are of particular challenges and demand for a rethink due to the issues of bandwidth limitation, signal fading, and interference in this kind of network [9–16]. When the majority of traffic over the network becomes multimedia in the near future, e.g., VoIP and video [17], QoS provisioning is critical for the user experience and it will probably affect the future applications of multi-hop wireless networks. The congestion problem is more serious in multi-hop wireless networks, since multiple copies of a packet are forwarded by intermediate nodes. When the network is in congestion, the throughput of the whole network will decrease sharply.

*Correspondence: eeqshguan@scut.edu.cn
[1]School of Electronic and Information Engineering, South China University of Technology, Guangzhou, People's Republic of China
Full list of author information is available at the end of the article

Chen *et al. EURASIP Journal on Wireless Communications and Networking* (2016) 2016:19

Page 2 of 11

Both delay-sensitive and non-delay-sensitive applications usually coexist in the same network, making QoS provisioning and congestion control highly coupled with each other. The scheduling algorithms for QoS provisioning are expected to deliver packets to the destination in time within their end-to-end delay requirements, while the non-delay-sensitive applications are serviced in a best-effort method. Delay-sensitive traffic is usually given higher priority over the non-delay-sensitive traffic by queueing management. However, the non-delay-sensitive applications are starved of any available network resources. The nodes with non-delay-sensitive traffic will be continuously contending and using the shared wireless channel to preempt delay-sensitive traffic from using the channel. The coexistence of these two types of applications will lead to a priority inversion in the network that the delay-sensitive traffic occupies the scheduling opportunities within a node, while the non-delay-sensitive traffic occupies transmission opportunities among nodes. Congestion will further happen to both types of traffic, resulting in packet collisions in the media and buffer overflow. In this case, packets will be dropped, or more queueing delay will be introduced. Then, the network may fail in serving both real-time and non-real-time applications. Thus, QoS provisioning and congestion control need to be considered jointly in multi-hop wireless networks [18].

We propose a joint QoS provisioning and congestion control based on our previous works of Differentiated Queueing Service (DQS) [19–21] and Semi-TCP [22, 23], which can provide packet-level QoSs and hop-by-hop congestion control, respectively. Both of them rely on the operations and judgements on queueing buffer, leading to competition and conflict of QoS provisioning and congestion control. Their packet-granular flexibilities make them both suitable for multi-hop wireless networks. So far, these two schemes are proposed and studied separately. In practice, DQS and Semi-TCP service delay-sensitive and non-delay-sensitive traffics, respectively. When we consider the integration with DQS and Semi-TCP together, some problems arise since they both operate on the same queueing buffer. On one hand, the queueing and scheduling policies in DQS will affect the performance of Semi-TCP because Semi-TCP adjusts its sending rate according to the buffer utilization. On the other hand, if the congestion is controlled at an inappropriate point and data packets are sent down to the buffer queue to occupy the channel only in the point of Semi-TCP's view, the QoS may become non-provisionable. Furthermore, without a proper queueing and scheduling scheme, the adoption of Semi-TCP in the joint scheme will intensify contention in the wireless channel, since it sends the TCP data greedily as long as the buffer queue in the link layer is available.

The contributions of this paper include the following:

- *Joint QoS provisioning and congestion control scheme*: We propose an integration scheme based on DQS and Semi-TCP to provide comprehensive services for multi-hop wireless networks.
- *Improvements to DQS and Semi-TCP for the joint scheme*: We propose some algorithms to improve the performance of the integrated scheme, including the following: (i) a fast estimation for a packet's latest departure time, which is a key step for the implementation of DQS in multi-hop wireless networks; (ii) a method to handle these packets whose lifetimes have expired, which improves bandwidth resource utilization; (iii) an adaptive acknowledgement (ACK) scheme that combines the ideas of SACK [24] and delayed ACK [25] to further decrease the ACK contention in the opposite flow direction.
- *Cross-layer implementation*: Since the implementations of DQS and Semi-TCP cross over the transport layer, network layer, and data link layer in the same protocol stack, we design a cross-layer architecture with a shared database for the cross-layer interactions and reduce the implementation complexity of the joint scheme.
- *Performance validation via simulations*: Extensive simulations in NS2 are conducted to verify the proposed scheme. The simulation results show that the joint scheme can significantly improve the end-to-end throughput and reduce the end-to-end delay in multi-hop wireless networks.

The remainder of this paper is organized as follows. Section 3 introduces overviews of DQS and Semi-TCP. Section 4 proposes the joint QoS provisioning and congestion control scheme in detail, while the simulation results are discussed in Section 5. Finally, the paper is summarized in Section 6.

2 Related work

Generally, there have been two QoS models for the Internet suggested by the Internet Engineering Task Force (IETF), i.e., Integrated Services (IntServ) and Differentiated Services (DiffServ). Research activities mainly focus on distributed scheduling schemes to provision QoS for multi-hop wireless networks. CSDPS + CBQ [26] categorizes traffic flows into classes. Each class is mapped into a physical queue, which is committed with a certain amount of bandwidth. This algorithm enables fair sharing of bandwidth resources while trying to maintain high throughput. However, it is unable to guarantee delay bound and packet dropping rates. Weighted fair queuing (WFQ) applies different scheduling priorities for

Chen *et al. EURASIP Journal on Wireless Communications and Networking* (2016) 2016:19

Page 3 of 11

statistical flow multiplexing. WFQ is proven to be able to guarantee the end-to-end delay bound when the data flow is leaky bucket constrained in wireline networks [27]. Early-deadline-first (EDF), which is able to separate the delay and throughput guarantee of a flow [28], is a dynamic priority scheduling algorithm that always schedules the packet with the earliest deadline. A coordinated EDF scheduler combined with wireless link scheduling is proposed in [29] to minimize the end-to-end delay in wireless networks.

The QoS granularity of above scheduling algorithms are based on either per-class (i.e., DiffServ-like schemes) or per-flow (i.e., IntServ-like schemes). With the per-flow QoS model, voluminous information needs to be stored in network nodes, which may complicate its implementation and influence its scalability. With the per-class model, although aggregating individual flows decreases the flow numbers to be handled by routers, it provides only a coarse QoS granularity with possible QoS over-provisioning. The recently proposed DQS [19–21] is quite different from the abovementioned QoS models, which schedules each packet according to its QoS requirement and end-to-end path situation. As a result, it provides a packet-level QoS granularity.

Congestion occurs more possibly in wireless multi-hop networks due to the shared and contention wireless channel. As one of the main functionalities of Transmission Control Protocol (TCP), many studies and experimental results have reported its performance degradation in multi-hop wireless networks [30, 31]. The losses of TCP packets, which results from buffer overflow in nodes, are usually reported as congestion in wired networks [32, 33]. Congestion control is triggered accordingly in TCP to slide down its congestion window. However, node mobility, poor channel quality, etc. will also damage TCP packets or result in the loss of TCP packets. In this case, TCP source nodes may misjudge the network situation and trigger unnecessary congestion control, which will decrease the sending windows and result in low network throughput [32]. Many TCP proposals have been proposed for multi-hop wireless networks [34, 35]. Some typical proposals include ATCP [36], Split TCP [37], negative acknowledgement (NACK) [38], explicit congestion notification (ECN) [39], and congestion-probing schemes [40].

Although these schemes can improve TCP performance to some extent, they still rely on the TCP layer to carry out congestion control, which takes at least one round-trip latency. Unfortunately, this round-trip time fluctuates greatly in multi-hop wireless networks. Hop-by-hop congestion control schemes [41, 42] are proposed accordingly, considering the channel access time constraint in the MAC layer. Instead of maintaining the structure of TCP, Semi-TCP was proposed recently in [22, 23] by moving the congestion control from the transport layer down to the lower layer in order to realize hop-by-hop congestion control in the MAC layer for multi-hop wireless networks. In this case, nodes can respond quickly to the congestion status, avoid congestion misjudgment, and release congestion promptly. Furthermore, the congestion control in Semi-TCP is performed for every packet, not only for TCP packets. It is effective to release network congestion.

To address the coupled issues, joint congestion control and scheduling algorithms have also been considered in the literature [43]. Huang et al. combined a window-based flow control and a distributed rate-based scheduling with the objective of maximizing the total utility and achieving low end-to-end delay [18]. Zhou et al. proposed a joint rate control, routing, and scheduling algorithm to provide proportional delay services [44] or maximize network utility [45]. We focus congestion control and QoS provisioning from the protocol perspective and study a novel joint DQS and Semi-TCP scheme in this paper.

3 Overviews of DQS and Semi-TCP
3.1 Differentiated Queueing Service (DQS)

DQS allows each real-time packet carrying its QoS requirement to realize granular QoS at the packet level. Real-time applications have some QoS requirements, among which the end-to-end delay is the most important QoS metric. For each data packet, it should have a maximum end-to-end delay that can also be seen as the initial value of packets' remaining lifetime [19]. A packet originated by the source node will traverse several intermediate nodes towards the destination. At each intermediate node, the packet may face different path situations, including (i) the number of hops that has travelled, (2) the remaining journey, and (3) contention in the shared channel. The following notations are used to introduce DQS:

- a: the arrival time stamp of a packet;
- e: the latest leaving time of a packet;
- T: the remaining lifetime of a packet;
- D: the maximum end-to-end delay of a packet;
- $\tilde{d}$: the actual delay that a packet has experienced at a node ;
- $\widehat{d}$: the estimated maximum delay allowed for a packet to stay at a node by DQS.

DQS estimates a maximum delay for each arriving packet to stay at a node according to its remaining lifetime and the current path situation. Given a packet along a path of n nodes, the following relationship should be held at node i, [19, 20]:

$$e_i = a_i + \widehat{d}_i, \tag{1}$$

Chen *et al. EURASIP Journal on Wireless Communications and Networking* (2016) 2016:19

Page 4 of 11

where

$$
\begin{aligned}
\widehat{d}_i &= T_i - \sum_{j=i+1}^{n} \tilde{d}_j \\
&= D - \sum_{k=1}^{i-1} \tilde{d}_k - \sum_{j=i+1}^{n} \tilde{d}_j.
\end{aligned}
\tag{2}
$$

In the above equation, $\sum_{k=1}^{i-1} \tilde{d}_k$ is the time consumed by the intermediate nodes that the packet has experienced and $\sum_{j=i+1}^{n} \tilde{d}_j$ presents the estimated delay for the remaining journey along the routing path except the current node. Therefore, the packet is allowed to stay for a maximum of $\widehat{d}_i$.

DQS develops buffering and scheduling based on $\widehat{d}_i$: (1) buffer admission control (BAC), which calculates the new latest departure time e_i of an arriving packet according to (1) and arranges packets in a descending order in the queue based on their latest departure time e_i; (2) output scheduler, which simply picks up the packet in the head of line to dequeue for service. Consequently, DQS is adaptive to the varying path situations in multi-hop wireless networks. The service priorities of packets are adjusted dynamically in line with the latest departure time to determine the positions in the queue.

3.2 Semi-TCP

Semi-TCP rearranges TCP's functionalities. It retains only reliability control and moves congestion control down to the MAC layer [22, 23] in wireless networks. It is a consensus that the MAC layer captures congestion status more accurately and responds faster than the TCP layer. Therefore, Semi-TCP is beneficial for nodes to avoid misjudgements and release congestion status quickly. Meanwhile, Semi-TCP exploits the sharing nature of wireless channel and transmits congestion information with piggyback mechanisms when nodes contend for channels. Accordingly, the IEEE 802.11 MAC only needs slight modifications to implement hop-by-hop congestion control as discussed below.

- *Congestion avoidance*: On the basis of the RTS/CTS handshake protocol, a negative CTS (nCTS) frame is introduced to realize hop-by-hop congestion control in the MAC layer. A node sends a nCTS to inform others to suspend receiving packets when it is in the congestion status, which is measured by the occupancy of the MAC buffer.
- *Sending rate adjustment*: The TCP source node sends packets according to its MAC buffer occupancy instead of TCP-ACKs. At the Semi-TCP source, the sending window is cancelled. The sending rate of the source depends on the buffer occupation in the down layers. When the buffers of the link layer

and network layer are available, a packet request command is fed back to the TCP source. Packets are only sent when the buffer is available.

Note that the congestion control in MAC is carried out for all data packets including TCP packets and real-time packets. Therefore, it may also benefit for QoS provisioning.

4 Joint QoS provisioning and congestion control

The joint QoS provisioning and congestion control scheme adopts DQS to provision QoSs in the logic link layer and Semi-TCP to control network congestion in the transport and MAC layers, making DQS and Semi-TCP coexisting in the same protocol stack. Evidently, the joint scheme needs the cooperation of the transport layer, the link layer, as well as routing layer to provide the path information, as shown in Fig. 1.

Although DQS and Semi-TCP have been developed independently, the joint effect of the two schemes has not been investigated. Some issues remain open in the joint QoS provisioning and congestion control scheme for multi-hop wireless networks. In this section, we discuss the issues of the delay estimation, overdue packet handling, the ACK mechanism, and the cross-layer design for the joint scheme.

4.1 Estimation of latest departure time

The estimate of the latest departure time e_i is a key step to implement DQS as shown in Eq. (1). Although the difficulty comes from the estimate of $\tilde{d}_j$ for $(j > i)$, they should be available before calculating e_i.

DQS schedules real-time data packets according to their end-to-end delay requirements, which can ensure data packets to obtain granular delay-sensitive services. The control packets, such as route broadcast packets, usually have a higher priority than data packets. These packets are simply inserted into a first-in-first-out (FIFO) queue
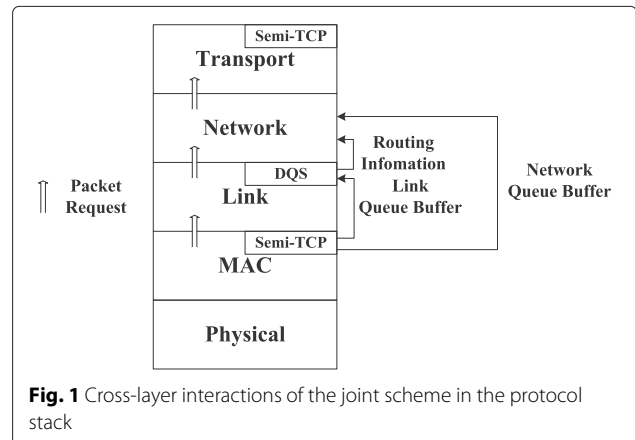

Fig. 1 Cross-layer interactions of the joint scheme in the protocol stack

with a higher priority over the data packet queue [20]. In order to obtain the latest departure time for each delay-sensitive data packet, Teng et al. [20] suggest sending probe packets periodically to estimate the delay over the remaining journey. However, this algorithm will introduce some problems. If the interval of sending probe packets is small, more accurate estimation may be obtained. However, it may occupy a large amount of wireless bandwidth and may even introduce broadcast storm to the network [46]. If the interval is long, the estimation may be not precise, making the time-sensitive e_i useless. In fact, it is not easy to have an accurate estimating approach in such dynamic multi-hop wireless networks.

In this paper, we propose a simple but effective approach, named average remaining lifetime, to estimate the latest departure time e_i. In the proposed scheme, (1) and (2) are rewritten as

$$
\begin{aligned}
e_i &= a_i + \widehat{d_i} \\
&= a_i + \frac{T_i}{n-i},
\end{aligned}
\tag{3}
$$

where i is the hop count in the upstream and n is the hop count in the downstream. Thus, e_i in Eq. (3) captures the path situation and provisions end-to-end QoS in a distributed manner by dispersing the left lifetime of a packet to the remaining hops on average. The accuracy of e_i in Eq. (3) may decline somewhat in the scene with static path situation since sending probe packets could get more accurate estimates. However, in the dynamic scene, it could avoid additional overhead, ease channel contention, as well as reduce the algorithm complexity, since it does not rely on the probe packets. In our joint scheme, nodes may compete fiercely for the access to the channel due to the adoption of Semi-TCP in the transport layer. In this sense, it is more appropriate to use the average remaining lifetime other than relying on a broadcast probing scheme.

4.2 Handle of overdue packets
Commonly, real-time applications have a requirement on the maximum tolerable end-to-end delay. Take the voice and the video as examples. Only intermittent voices could be caught if the end-to-end delay cannot be guaranteed. It is hard to make out what the voices are saying if a sentence is divided into several fragmentary words. Similarly, no one wants to hear and watch the meaningless replay when enjoying the streaming applications. The worse case for the network is that the overdue packets whose lifetimes have expired are still forwarded in the network, although they are probably dropped at the destination. The delivery of such overdue packets not only means a waste of network resources but also occupies the opportunities to service the regular real-time packets, making the case get even worse.

Each data packet carries its lifetime in the joint scheme and will be served by the DQS scheduling algorithm. These mechanisms facilitate the design of an approach to handle the overdue packets. Generally, the occurrences of packet expiration may be detected in two cases: on receiving a packet and on sending a packet at a node. The former case has been addressed in [20] where the overdue packets are dropped immediately upon the arrival, while the latter one is still untapped. A node may receive and enqueue a packet whose remaining lifetime is close to expire. When the scheduler begins to schedule this packet, its remaining lifetime may have expired since the packets in the head of the buffer queue may take extra time to contend for the channel. If we insist on sending the expired packet, it will probably be dropped at the next hop because its T_i expires at its departure. Therefore, in the joint scheme, we will also check the remaining lifetime of each packet at its departure. We propose a scheme to drop the packet whose T_i is expired at its departure to further improve the utilization of bandwidth and the capability to provision QoS. The congestion will also be mitigated at the same time if the network is congested.

4.3 Adaptive ACK scheme
The source node with Semi-TCP does not rely on ACKs to judge the congestion status of the network and adjust the sending rate. However, it still depends on ACKs to perform reliability control. Studies have shown that lots of ACK packets in the opposite direction of data flows conflict with data packets, which exacerbates the channel contention. Cutting off duplicate ACKs could lessen inter-flow competitions and improve throughput [22]. However, the TCP source node will capture packet losses slowly without enough ACKs. This is because in this case, the source node can learn packet losses only by the expiration of the retransmission timer in at least a round-trip time. If a number of TCP packets are lost, the Semi-TCP source node needs to wait for multiple RTTs to retransmit the lost packets due to the fact that the retransmission timer equals to several times of RTTs. The retransmission timer will grow larger when the hops between source and destination nodes increase. Without ACKs, the sender retransmits packets even though the packets have already been received successfully.

The missing number of ACKs sometimes is not contributed to the losses of data packets but due to the conflict of data packets and ACKs in the channel in multi-hop wireless networks. ACKs from destination nodes will intensify competition to some extent, which also degrades TCP throughput.

Targeting at reducing the number of ACKs in the network, we combine the ideas of selective ACK (SACK) and delayed ACK schemes to design an adaptive ACK scheme for the joint scheme. SACK [24] allows the destination to

Chen *et al. EURASIP Journal on Wireless Communications and Networking* (2016) 2016:19

Page 6 of 11

acknowledge out-of-order segments and retransmit lost packets selectively. SACK makes the TCP source node retransmit multiple lost packets in time, reducing unnecessary ACKs and retransmissions. Consequently, the TCP source node can empty more acknowledged packets in the sending buffer, which are indicated by SACK. For mobile nodes, emptying sending buffer timely is helpful and efficient because they are usually storage capacity limited. Delayed ACK [25] does not need to acknowledge every TCP packet. It is designed to delay an interval of time to collect in-order segments and acknowledge them at one time. The ideas of SACK and delayed ACK are introduced into the joint scheme to not only improve the ratio of acknowledged TCP packets but also to provide more schedulable space for the real-time traffic.

The adaptive ACK scheme can be described as follows.

- A delay timer is adopted at the destination for acknowledgement. No ACK is fed back until the delay timer expires. Thus, a large amount of ACKs are avoided to push into the network, mitigating the channel contention. In-order segment blocks, which allow to be discontinuous between each block, are collected in the timing period.
- When the delay timer expires, the destination prepares to send ACKs. However, if the obtained discontinuous segment blocks N is less than the maximum allowable blocks N_0 in the SACK options, the delay timer will be extended for the next delaying period.
- There exists another case that many data packets are received at the destination in some period. This case will cause bursty ACKs since multiple SACKs are required to acknowledge these packets. The bursty ACKs are not preferable for the network that supports QoS. Therefore, we set a threshold number of segment blocks, denoted as N_{th}, for the timing period. The delay timer is adjusted based on N and N_{th}, which is summarized as follows:

$$T = \begin{cases} T + \text{tick}, & N < N_0 \\ \frac{T}{2^{N/N_0}}, & N_0 < N < N_{th} \\ 0, & N > N_{th}. \end{cases} \quad (4)$$

4.4 Cross-layer design

The sharing wireless channel and its dynamic nature have been witnessed to make the traditional strict layered design difficult in providing efficient QoS support [47, 48]. The problem is more severe in our joint scheme. As shown in Fig. 1, DQS requires the routing information in the network layer to estimate the latest departure time for each packet; the buffer occupation in the down layers is used to adjust the sending rate of Semi-TCP, and

the congestion control is conducted in MAC. It is a common way in multi-hop wireless networks to allow each layer access to the other layers directly for cross-layer information. For example, the routing table information used by DQS queue can be directly acquired from the routing layer. That is, the routing layer provides an interface to the DQS queue, which belongs to different layers. In our joint scheme, the cross-layer interactions become more complex, requiring more interfaces and increasing the complexity to manage and maintain these interfaces in each layer. The cross-layer interactions in the layered protocol stack make it a spaghetti-like architecture [49].

According to the idea of a shared database [50], we build a shared database that can be accessed by the data link layer, network layer, and transport layer as shown in Fig. 2. The database will provide services of storing and searching information for the protocol layers. The shared database architecture makes the joint scheme more clear and practical for implementation. Since the packet request is carried out layer-by-layer, the layered architecture can support packet requesting essentially.

5 Simulation and evaluation

We study the performance of the joint QoS provisioning and congestion control scheme through simulations via NS-2.31 [51] in comparison with schemes listed in Table 1, where the scheme of Semi + DQS is our proposal. The following metrics are used to evaluate the schemes:

- *Goodput*: the number of data packets received by the destination node that are sent by the source node within a certain time. Here, we only consider the packets that satisfy the maximum end-to-end delay for real-time traffic.
- *Delivery ratio*: the ratio of the number of successfully delivered packets to the number of sending packets.
- *Average end-to-end delay*: the average delay for a packet to experience from the source node to its destination. It includes the route discovery delay,
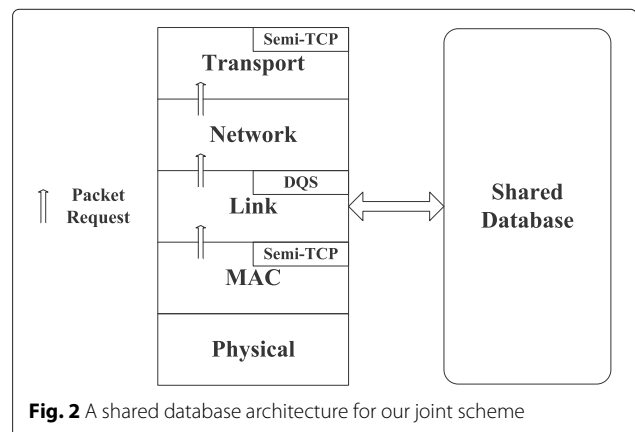

Fig. 2 A shared database architecture for our joint scheme

Table 1 Simulated schemes for comparison

Scheme	TCP protocol	Scheduling algorithm	MAC protocol
Semi + DQS	Semi-TCP	DQS	802.11 with congestion control
Semi + PriQ	Semi-TCP	PriQueue	802.11 with congestion control
Reno + DQS	TCP-Reno	DQS	802.11
Reno + EDF	TCP-Reno	EDF	802.11

queuing delay, accessing delay, transmitting delay, and propagation delay [20].

Two kinds of network topologies, static chain topology and mobile random topology, are investigated in the simulations. In the static chain topology, nodes are placed in a straight line, while in the mobile random topology, nodes are deployed randomly and move in random velocities.
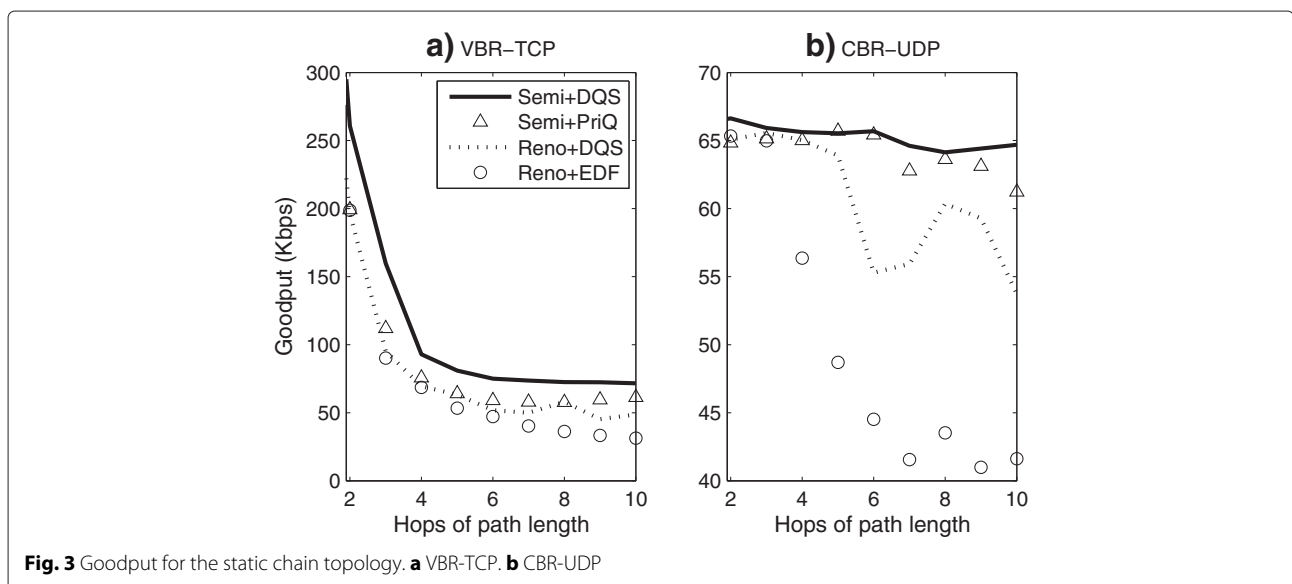
Constant bit rate (CBR) traffic and variable bit rate (VBR) traffic are generated at the source to present real-time applications and the best-effort applications, respectively, which are attached to User Datagram Protocol (UDP) agents and TCP agents in the NS-2 simulator. Since it is assumed in DQS that every applications should have an end-to-end delay requirement, the end-to-end delay requirements of the CBR and VBR traffic in the simulation are set to 150 and 250 ms [52], respectively. The basic delay interval of our adaptive ACK is set to 2 s. The whole simulation period in the study is 300 s.

We first show the performance of the four schemes in Table 1 versus path lengths in the static chain topology. As illustrated in Fig. 3, with the increase of hops, our proposal always obtains the maximum TCP and UDP

goodput. This is because our adaptive ACK scheme compresses ACK flows by delaying ACKs, which contributes to reducing competition, improving throughput of TCP, and providing more schedulable space for QoS. As shown in Fig. 3a, the two schemes adopting Semi-TCP have better throughput without sending duplicated ACKs. In addition, the schemes that adopt DQS to schedule data packets according to its latest departure time receive more packets satisfying the maximum end-to-end delay at the destination. The performance gain also comes from the modification discussed in Section 4.2, which checks the remaining lifetime and drops the overdue packets immediately to improve utilization of bandwidth. With PriQueue and EDF, some packets arriving at the destination node may have already expired, resulting in a decreased throughput.

From Fig. 4, we can see that the schemes with Semi-TCP achieve higher delivery ratios than those with Reno-TCP in the chain topology. The reason is attributed to the hop-by-hop congestion control in our proposed Semi-TCP which avoids the packet losses due to congestion efficiently. Semi + DQS can achieve higher throughput than Semi + PriQ as shown in Fig. 3, and meanwhile, it also outperforms slightly the Semi + PriQ in terms of delivery ratio.

As shown in Fig. 5a, our joint scheme outperforms the other three schemes in terms of the average end-to-end delay for the TCP flow. The performance gain gets more evident as the number of hops increases, where the time for a TCP packet to travel from a source node to its destination node increases. As mentioned earlier, DQS can consider the end-to-end path situation for every packet. However, PriQueue schedules packets with the FIFO policy, while EDF schedules packets based on a given per-hop delay without considering the end-to-end
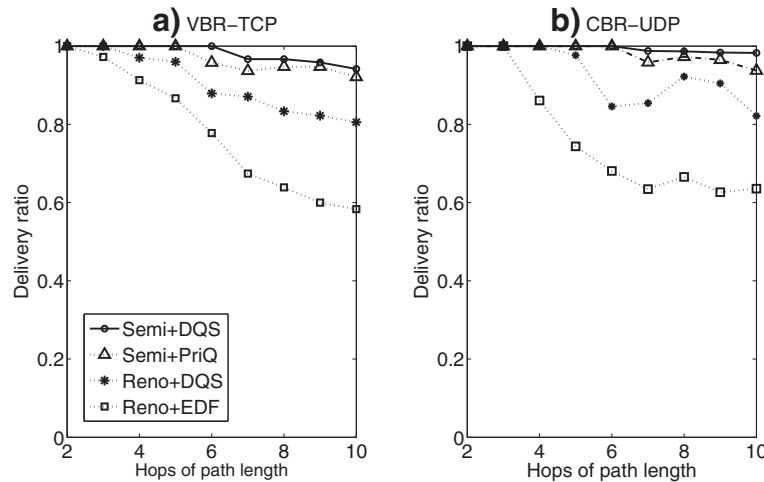

Fig. 3 Goodput for the static chain topology. **a** VBR-TCP. **b** CBR-UDP

Chen *et al. EURASIP Journal on Wireless Communications and Networking* (2016) 2016:19

Page 8 of 11

Fig. 4 Delivery ratios for the static chain topology. **a** VBR-TCP. **b** CBR-UDP

path situation [20]. As a result, they are more sensitive than DQS to the number of hops along the path. Although the Reno + DQS scheme adopts DQS to schedule TCP packets, the time to relieve congestion in TCP-Reno is much longer than that in Semi-TCP, which degrades its delay performance.

In Fig. 5b, Reno + DQS performs better than our joint scheme since more bandwidth is allocated to TCP flows with the integrated scheme, making the UDP delay a little larger. This reason is also indicated in Fig. 3b that Reno + DQS achieves a UDP throughput much less than our joint scheme does. Thus, Reno + DQS supports less real-time applications.

For the random mobile topology, 27 nodes are placed randomly in an area with a dimension of 800×800 m. The

numbers of TCP and UDP sources are both set to 4. The end-to-end delays of CBR and VBR are again set to 150 and 250 ms, respectively. The ACK delay interval is still set to 2 s. The minimum speed of mobile nodes is 0.5 m/s, and the maximum speed is changed to analyze the goodput and the average end-to-end delay of the four schemes. In this topology, node mobility will often cause links to be broken and routes between a source and a destination to be changed.

In the simulations, we investigate the schemes in Table 1 in 20 different scenes and average the results. As illustrated in Fig. 6, with the increase of the maximum speed, the goodput of UDP and TCP fluctuates, but our joint scheme performs best. In the random mobile topology, the end-to-end connectivity is often
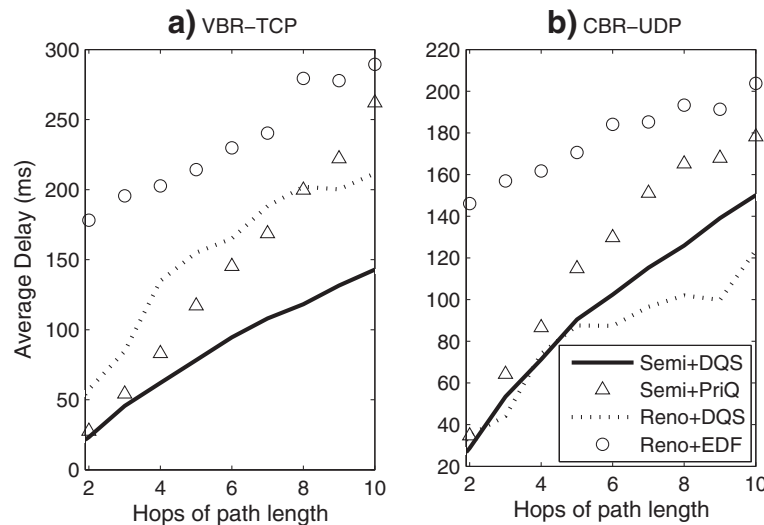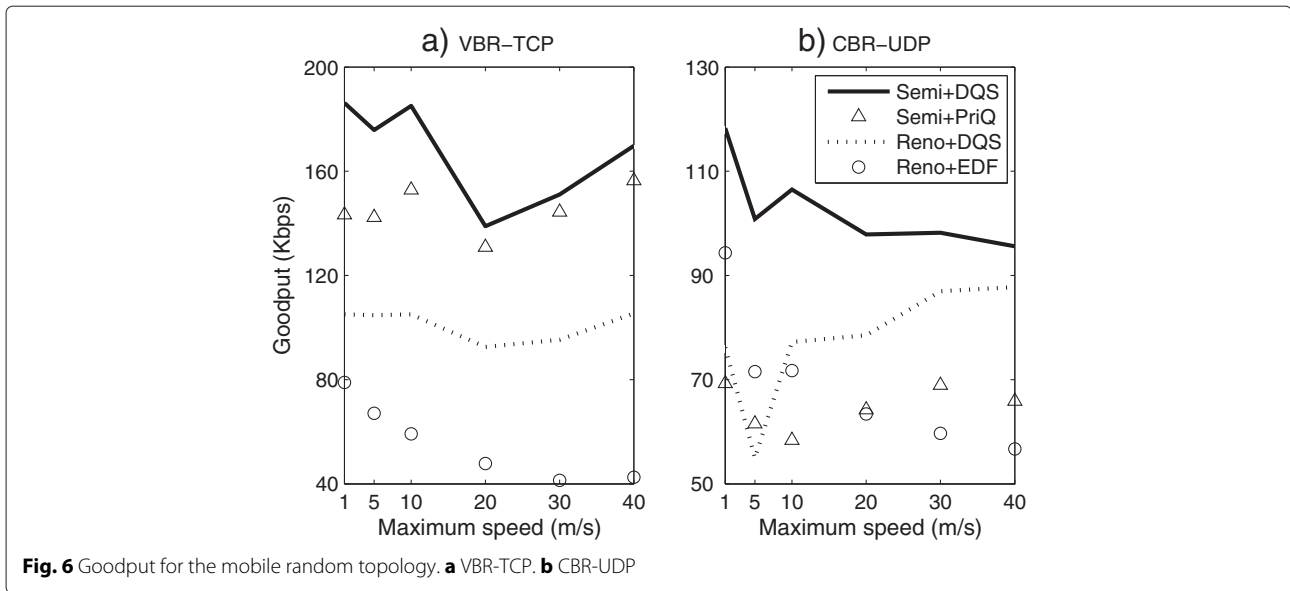

Fig. 5 Average end-to-end delay for the static chain topology. **a** VBR-TCP. **b** CBR-UDP

Chen *et al. EURASIP Journal on Wireless Communications and Networking* (2016) 2016:19

Page 9 of 11

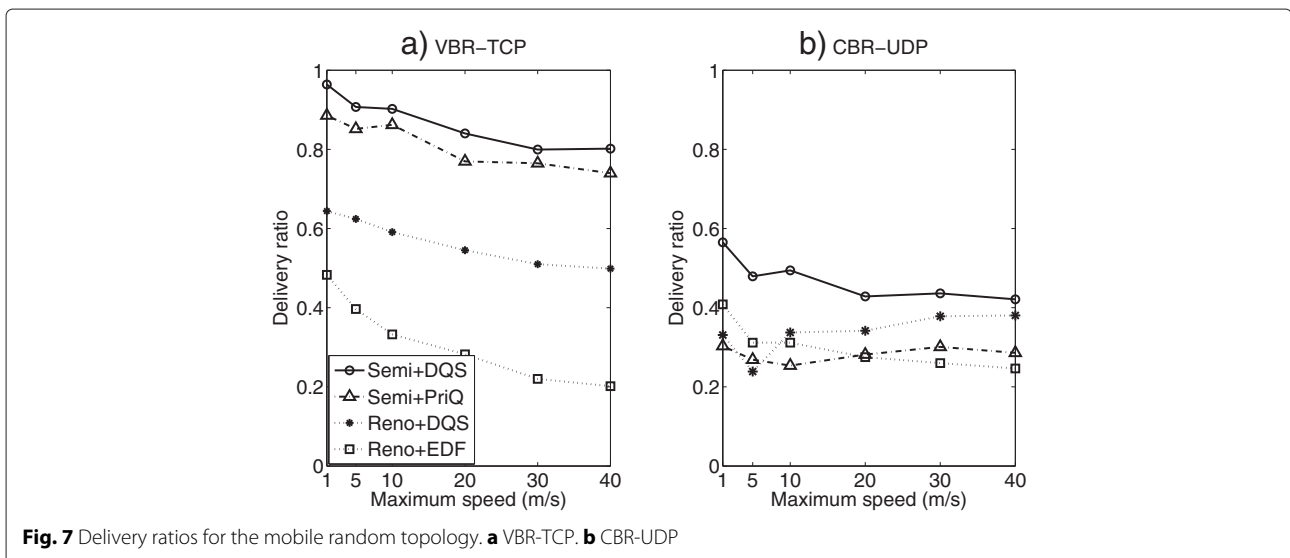**Fig. 6** Goodput for the mobile random topology. **a** VBR-TCP. **b** CBR-UDP

intermittent, which leads to more congestion misjudgment in the original TCP. Therefore, schemes adopting Semi-TCP have a distinct advantage in the dynamic environment, where the end-to-end routes are often changed. Our joint proposal can compress ACK packets through a delayed ACK-like mechanism and benefit from DQS that can schedule packets according to its remaining lifetime and path situations. Thus, more packets stratifying the delay requirements can be received by destination nodes, achieving higher throughput in both TCP and UDP traffics.

Semi-TCP and DQS also show better performance in terms of delivery ratio in the mobile random topology. Figure 7a shows that Semi-TCP with hop-by-hop congestion control achieves higher delivery ratio for VBR

traffic than Reno-TCP. In the meanwhile, DQS achieves higher delivery ratio for CBR traffic with QoS requirements than PriQ and EDF in Fig. 7b. The higher delivery ratio can explain the reason our proposed joint scheme achieves the best goodput in Fig. 6.

Figure 8 shows the average end-to-end delay of TCP and UDP flows in the random mobile topology. In this scenario, it is more difficult to guarantee the delay of packets. It is particularly necessary to carry out a suitable scheduling algorithm when packets of multiple flows intersect in one node. The average remaining lifetime estimation in our proposal without periodical probing discussed in Section 4.1 decreases the node contention and makes the our joint scheme perform better in terms of delay than the other schemes. Additionally, the modified Semi-TCP
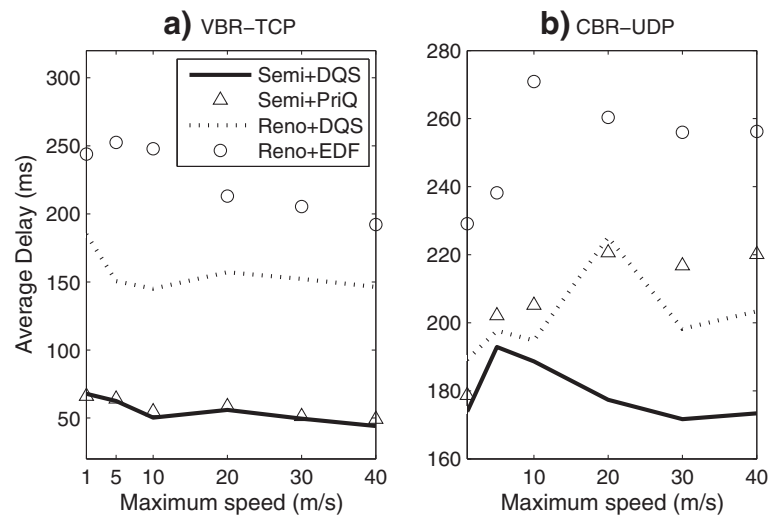

Fig. 7 Delivery ratios for the mobile random topology. **a** VBR-TCP. **b** CBR-UDP

Chen *et al. EURASIP Journal on Wireless Communications and Networking* (2016) 2016:19

Page 10 of 11

Fig. 8 Average end-to-end delay for the mobile random topology. **a** VBR-TCP. **b** CBR-UDP

benefits from using the SACK mechanism discussed in Section 4.3, which helps source nodes to capture packet losses faster. As a result, the source node can retransmit multiple packets timely, which also decreases the average end-to-end delay. The simulation results in the random mobile topology confirm our discussion conclusion that the proposed joint scheme in this paper is flexible and adaptive to dynamic multi-hop wireless networks.

6 Conclusions

A joint QoS provisioning and congestion control scheme based on our previous works of DQS and Semi-TCP has been proposed for multi-hop wireless network in this paper. To address the arising issues in our proposal when DQS and Semi-TCP are coexisted in the same protocol stack, we propose a simple and efficient approach to fast estimate the latest departure time for packets, a scheme to handle overdue packets, and an adaptive ACK mechanism to improve the overall performance of delivering non-realtime and real-time applications. We have also implemented our proposal in a shared database cross-layer protocol architecture. The simulation results showed that the joint scheme performs better with improved goodput and delivery ratio, as well as decreased end-to-end delay for multi-hop wireless networks.

Competing interests
The authors declare that they have no competing interests.

Author details
[1] School of Electronic and Information Engineering, South China University of Technology, Guangzhou, People's Republic of China. [2] College of Information Engineering, Shanghai Maritime University, Shanghai, People's Republic of China.

References
1. M Yang, Y Li, D Jin, L Zeng, X Wu, AV Vasilakos, Software-defined and virtualized future mobile and wireless networks: a survey. Mobile Netw. Appl. **20**(1), 4–18 (2014)
2. Z Sheng, S Yang, Y Yu, A Vasilakos, J Mccann, K Leung, A survey on the IETF protocol suite for the internet of things: standards, challenges, and opportunities. IEEE Wirel. Commun. **20**(6), 91–98 (2013)
3. IF Akyidiz, X Wang, A survey on wireless mesh networks. IEEE Comm. Mag. **43**(9), 23–30 (2005)
4. AV Vasilakos, Y Zhang, T Spyropoulos, *Delay tolerant networks: protocols and applications*. (CRC press, Florida, US, 2011)
5. A Dvir, AV Vasilakos, Backpressure-based routing protocol for DTNs. ACM SIGCOMM Comput. Commun. Rev. **41**(4), 3264–3273 (2015)
6. PHJ Chong, F Adachi, S Hamalainen, V Leung, Technologies in multihop cellular network. IEEE Comm. Mag. **45**(9), 64–65 (2007)
7. A Gupta, RK Jha, A survey of 5G network: architecture and emerging technologies. IEEE Access. **3**, 1206–1232 (2015)
8. SW Peters, RW Heath, The future of WiMAX: multihop relaying with IEEE 802.16 j. IEEE Commun. Mag. **47**(1), 104–111 (2009)
9. C Busch, R Kannan, AV Vasilakos, Approximating congestion + dilation in networks via "quality of routing" games. IEEE Trans. Comput. **61**(9), 1270–1283 (2012)
10. Y Zeng, K Xiang, D Li, AV Vasilakos, Directional routing and scheduling for green vehicular delay tolerant networks. Wireless networks. Wireless Netw. **19**(2), 161–173 (2013)
11. P Li, S Guo, S Yu, AV Vasilakos, Reliable multicast with pipelined network coding using opportunistic feeding and routing. IEEE Trans. Parallel Distributed Syst. **25**(12), 3264–3273 (2014)
12. Y Yao, Q Cao, AV Vasilakos, Edal: An energy-efficient, delay-aware, and lifetime-balancing data collection protocol for heterogeneous wireless sensor networks. IEEE/ACM Trans. Networking. **23**(3), 810–823 (2015)
13. Y Niu, C Gao, Y Li, L Su, D Jin, A Vasilakos, Exploiting device-to-device communications in joint scheduling of access and backhaul for mmwave small cells. IEEE J. Sel. Areas Commun. **33**(10), 2052–2069 (2015)

Chen *et al. EURASIP Journal on Wireless Communications and Networking*   (2016) 2016:19

Page 11 of 11

14. X Zhang, Y Zhang, F Yan, AV Vasilakos, Interference-based topology control algorithm for delay-constrained mobile ad hoc networks. IEEE Trans. Mobile Comput. **14**(4), 742–754 (2015)

15. T Meng, F Wu, Z Yang, G Chen, A Vasilakos, Spatial reusability-aware routing in multi-hop wireless networks. IEEE Trans. Comput., 244–255 (2015)

16. L Liu, Y Song, H Zhang, H Ma, AV Vasilakos, Physarum optimization: a biology-inspired algorithm for the steiner tree problem in networks. IEEE Trans. Comput. **64**(3), 819–832 (2015)

17. 4G Americas, 4G Mobile Broadband Evolution: 3GPP Release 10 and Beyond – HSPA+, SAE/LTE and LTE-Advanced (2011). http://www.4gamericas.org/

18. P-K Huang, X Lin, C-C Wang, in *Proc. IEEE INFOCOM 2011*. A low-complexity congestion control and scheduling algorithm for multihop wireless networks with order-optimal per-flow delay, (Shanghai, China, 2011), pp. 2588–2596

19. S Jiang, Granular differentiated queueing services for QoS: structure and cost model. ACM SIGCOMM Comput. Commun. Rev. **35**(2), 13–22 (2005)

20. X Teng, S Jiang, G Wei, G Liu, in *IEEE Vehicular Technology Conference (VTC Spring 2008)*. A cross-layer implementation of differentiated queueing service (DQS) for wireless mesh networks, (Singapore, 2008), pp. 2233–2237

21. L Tang, Q Guan, S Jiang, B Guo, A deadline-aware and distance-aware packet scheduling algorithm for wireless multimedia sensor networks. Int. J. Distrib. Sens. Netw. **2015** (2015)

22. S Jiang, Q Zuo, G Wei, in *Proceedings of the 4th ACM Workshop on Challenged Networks*. Decoupling congestion control from TCP for multi-hop wireless networks: semi-TCP, (Beijing China, 2009), pp. 27–34

23. Y Cai, S Jiang, Q Guan, FR Yu, Decoupling congestion control from TCP (semi-TCP) for multi-hop wireless networks. EURASIP J. Wireless Commun. Netw. **2013**(1), 1–14 (2013)

24. M Mathis, J Mahdavi, S Floyd, A Romanow, TCP selective acknowledgement options. Internet Engineering Task Force RFC 2018 (1996)

25. R Braden, Requirements for internet hosts-communication layers. RFC 1122 (1989)

26. C Fragouli, V Sivaraman, MB Srivastava, in *Proc. IEEE INFOCOM'98*. Controlled multimedia wireless link sharing via enhanced class-based queuing with channel-state-dependent packet scheduling, (San Francisco, CA, 1998), pp. 572–580

27. D Stiliadis, A Varma, Latency-rate servers: a general model for analysis of traffic scheduling algorithms. IEEE/ACM Trans. Netw. (ToN). **6**(5), 611–624 (1998)

28. Y Cao, VOK Li, Scheduling algorithms in broadband wireless networks. Proc. IEEE. **89**(1), 76–87 (2001)

29. P Jayachandran, M Andrews, in *Proc. IEEE INFOCOM 2010*. Minimizing end-to-end delay in wireless networks using a coordinated edf schedule, (San Francisco, CA, 2010), pp. 1–9

30. G Holland, N Vaidya, Analysis of TCP performance over mobile ad hoc networks. Wireless Netw. **8**(2/3), 275–288 (2002)

31. G Xylomenos, GC Polyzos, P Mahonen, M Saaranen, TCP performance issues over wireless links. IEEE Commun. Mag. **39**(4), 52–58 (2002)

32. Z Fu, H Luo, P Zerfos, S Lu, L Zhang, M Gerla, The impact of multihop wireless channel on TCP performance. IEEE Trans. Mobile Comput. **4**(2), 209–221 (2005)

33. S Rangwala, Congestion control in multi-hop wireless networks. PhD thesis, (University Of Southern California, 2010)

34. H Balakrishnan, V Padmanabhan, S Seshan, R Katz, A comparison of mechanisms for improving TCP performance over wireless links. IEEE/ACM Trans. Netw. **5**(6), 756–769 (1997)

35. H Elaarag, Improving TCP performance over mobile networks. ACM Comput. Surv. **34**(3), 357–374 (2002)

36. J Liu, S Singh, ATCP: TCP for mobile ad hoc networks. IEEE J. Sel. Areas Commun. **19**(7), 1300–1315 (2002)

37. S Kopparty, SV Krishnamurthy, M Faloutsos, SK Tripathi, in *IEEE Global Telecommunications Conference GLOBECOM'02*. Split TCP for mobile ad hoc networks, (Taipei, Taiwan, 2002), pp. 138–142

38. F Sun, VOK Li, SC Liew, in *Proc. IEEE WCNC 2004*. Design of snack mechanism for wireless TCP with new snoop, vol. 2, (Atlanta, US, 2004), pp. 1051–1056

39. S Floyd, TCP and explicit congestion notification. ACM SIGCOMM Comput. Commun. Rev. **24**(5), 8–23 (1994)

40. S Mascolo, C Casetti, M Gerla, MY Sanadidi, R Wang, in *Proc. 7th Annual Int. Conf. Mobile Computing and Networking*. TCP Westwood: bandwidth estimation for enhanced transport over wireless links, (2001), pp. 287–297

41. PP Mishra, H Kanakia, A hop by hop rate-based congestion control scheme. ACM SIGCOMM Comput. Commun. Rev. **22**(4), 112–123 (1992)

42. Y Yi, S Shakkottai, Hop-by-hop congestion control over a wireless multi-hop network. IEEE/ACM Trans. Netw. **15**(1), 133–144 (2007)

43. AP Silva, S Burleigh, CM Hirata, K Obraczka, A survey on congestion control for delay and disruption tolerant networks. Ad Hoc Netw. **25**, 480–494 (2015)

44. A Zhou, M Liu, Z Li, E Dutkiewicz, Cross-layer design for proportional delay differentiation and network utility maximization in multi-hop wireless networks. IEEE Trans. Wireless Commun. **11**(4), 1446–1455 (2012)

45. Z Ding, D Wu, Sliding mode based congestion control and scheduling for multi-class traffic over per-link queueing wireless networks. IEEE Trans. Veh. Tech. **62**(3), 1276–1288 (2013)

46. YC Tseng, SY Ni, YS Chen, JP Sheu, The broadcast storm problem in a mobile ad hoc network. Wireless Netw. **8**(2/3), 153–167 (2002)

47. Q Zhang, YQ Zhang, Cross-layer design for QoS support in multihop wireless networks. Proc. IEEE. **96**(1), 64–76 (2008)

48. L Zhou, X Wang, W Tu, GM Muntean, B Geller, Distributed scheduling scheme for video streaming over multi-channel multi-radio multi-hop wireless networks. IEEE J. Selected Areas Commun. **28**(3), 409–419 (2010)

49. M Conti, G Maselli, G Turi, S Giordano, Cross-layering in mobile ad hoc network design. Computer. **37**(2), 48–51 (2004)

50. V Srivastava, M Motani, Cross-layer design: a survey and the road ahead. IEEE Comm. Mag. **42**(12), 112–119 (2005)

51. NS2. http://www.isi.edu/nsnam

52. F Fluckiger, *Understanding networked multimedia: applications and technology*. (Prentice Hall International Ltd., UK, 1995)