Wireless Hop-by-hop Credit-Based Flow Control Extended to Source for Stable Best Effort Traffic

Rainer Schoenen, Halim Yanikomeroglu

Department of Systems and Computer Engineering, Carleton University, Canada

***Abstract*—Data traffic is expected to grow faster than capacity in future wireless networks. Therefore it will become unavoidable to deal with congestion. Bottlenecks are located on the wireless links because back-haul and Internet are overprovisioned. Traffic routed towards the user terminal (UT) in down-link direction keeps coming in through a big pipe until it reaches the base station (BS). The following wireless links can only carry a limited data rate due to congestion. In a multi-hop situation buffers before the bottlenecks ramp up and become unstable, leading to packet loss. While real-time traffic is safe due to CAC, highest static priority and over-provisioning, best effort data traffic experiences congestion and therefore packet losses.**

A wireless flow control based on a credit-based hop-by-hop concept can solve this problem by avoiding any buffer overflow completely. This paper proposes extending the closed flow control loops to the source, either by a genuine credit-based flow control or by TCP rate control with deep packet inspection and ACK modification.

This paper analyses the queueing behavior with stochastic Petri nets models. Markov state analysis provides numeric performance results. The example scenario consists of two wireless relayed hops and a wired back-haul with different control approaches for the hop between source and bottleneck.

***Index Terms*—wireless flow control, WFC, credit based, congestion, relay, multihop, stochastic Petri nets**

I. Introduction

In the future the traffic generated and consumed by wireless terminals will increase faster than the capacity can grow due to improvements of radio technologies. Matching demand and supply [1] will become harder and harder, the more smartphone "killer" applications take over. All wireless systems have capacity limits which are significantly lower than what the fixed network back-haul can deliver per user. This will inevitably lead to traffic congestion on the wireless links, especially during the busy hours. Real-time (RT) traffic will not be affected because of a higher priority and strictly limited load due to call admission control (CAC). The remaining non-real-time (NRT) traffic must be protected from loss by buffer overflow. Loss by channel errors can be assumed negligible due to ARQ retransmissions on layer two. Additional tasks are to provide fairness among the congested flows and avoid starvation of queues On transport layer, the end-to-end TCP protocol provides flow control and rate adaptation [2]. However TCP is known to perform insufficient with the characteristics of wireless links in congestion and heavy load. With traditional TCP congestion control schemes still being suboptimal [3], fundamental limitations make it hard to further extend TCP the traditional way [4]. Approaches to break down end-to-end control into smaller loops were just recently proposed [5], [6].

In this paper a credit-based flow control (CBFC) protocol is applied in each node (hop-by-hop) of a multihop (relay) system [7]. This credit based wireless flow control (WFC) is an adaptation of an ATM flow control scheme [8]. In [9] it has been proven deterministic behavior. and its advantages are precision of control and stability of queue contents, i.e., no buffer overflow and packet loss. Best effort (BE) traffic can safely be operated into congestion and stay in congestion for a long time, if there would be no TCP on top, which considers delayed (queued) packets as lost and regulates the window size (thus the sender rate) down in an ineffective way. WFC can cope with the future congested traffic load situation if the transport layer control is also adapted.

In wireless systems CBFC is a rather new concept, and related work [10], [11] has not studied yet how to close the gap between the closed loop wireless links and the feeder links towards the source. This paper proposes extending the closed loops beyond the wireless (bottleneck) links, so that buffer overflows in the base station (BS) before the bottleneck are deterministically impossible by construction. For this first flow control segment three different options are studied, the uncontrolled first hop, a pure credit-based solution and the rate control of sources. In this paper the structure is modeled using stochastic Petri nets (SPN), and numeric performance results come from its stochastic Markov chain analysis without the shortcomings of simulations.

Petri nets [12], stochastic Petri nets (SPN) and generalized SPN (GSPN) [13] are powerful tools for modeling and analysis of stochastic systems with decent tool support [14]. While mainly used in computer science, the use for communication systems [15], [16] is very promising. There are modeling approaches for IEEE 802.16 [17], IEEE 802.11 [18] and TCP [19]. IMT-Advanced cellular systems [20] are also a current topic [21]. One big advantage of SPN models is the queue equivalent of a place, of which the state probabilities are determined easily. Therefore queueing and buffering can be studied in complex situations like in flow control where the ingress and egress processes are not regular. In overload condition (congestion) this is also a situation which cannot be treated with classical queueing theory. In addition, it is easy to obtain more than just average values by having the full buffer statistics.

The paper further consists of Section II which introduces the basics of stochastic Petri nets. Then the flow control protocol model is introduced with a scheduler and a channel model. The wireless extension in a two (three with back-haul) hop scenario is presented in section IV. Performance results in section V show the buffer behavior and advantages of the approach.

II. Generalized Stochastic Petri Nets

Petri nets [12] are a paradigm for formal analysis of systems that are described in a graphical manner. As a mathematical description (language), its main property is having a discrete multidimensional state space with arbitrary connectivity which allows it to model complex concurrent, asynchronous, distributed, deterministic or stochastic systems. State machines, marked graphs [22], flow charts and description languages are subclasses of PN. The literature on the underlying graph theory is vast, but [12] gives a good overview. Important questions to study are liveness analysis and the reachability set which corresponds to a Markov chain.

A Petri net is a directed, weighted, bipartite graph with two sets of nodes, namely *places* (P_i) and *transitions* (T_j). Transitions are drawn as boxes, places as circles. Arcs between these elements must be directed and are called input arcs if they connect from a place to a transition, else output arc. Additionally there is the convenience notation of a disabling arc, drawn with a small circle at the end of the arc.

Input arcs connect P_i to T_j and carry an integer weight w_{ij}. output arcs connect T_j to P_i with a similar multiplicity v_{ij}. The incidence matrix $\mathbf{D} = [v_{ij} - w_{ij}]$ consists of these integer weights. *Tokens* (denoted as dots) are an integer annotation of a place. The current number of tokens in place P_i is denoted with $\#P_i = m_i$. All the token counts in places form the state called *marking* $\vec{m}$. The start state is defined by the initial marking $\vec{m}_0$. PN change their marking state in a discrete manner by *firing* of a transition T_j. Here we define the firing vector $\vec{t}_{k-1}$ of all zeros with the exception of a single '1' at the j.th index. Firing a transition is subject to a basic rule: It is allowed to fire if all places connected to input arcs contain at least w_{ij} tokens (to consume) plus all places connected with disabling arcs must be empty.

$$\vec{m}_k = \vec{m}_{k-1} + \vec{t}_{k-1} \cdot \mathbf{D} \tag{1}$$

Ordinary PN are timeless, but stochastic PN (SPN) fire timed transitions (empty boxes) according to random events described by exponential inter-firing times λ. Due to the memoryless property, the resulting reachability graph (RG) forms a Markov chain (MC) [13]. By solving the steady-state solution of the underlying MC, all performance metrics derived from the probability mass function of the number of tokens in a place can be obtained.

Extensions known as Generalized SPN (GSPN) and Deterministic and Stochastic SPN (DSPN) allow to have other than exponential transition behavior. Immediate transitions (thick bars) can be annotated with priorities and weights to model the outcome likelihood of deterministic and random decisions.

III. The credit-based flow control model

The CBFC protocol assumed in this paper is derived from QFC [23], CT [24]) or FCVC [25] which was originally developed for the ATM ABR traffic class. Analysis in [8], [9] showed a great potential of this method. In the wireless domain the model was first proposed in [11] but was only assumed for the wireless (bottleneck) links. In this paper it is extended to include the backhaul and Internet links to control up to the source without gap, i.e., a real hop-by-hop protocol extending from end to end.

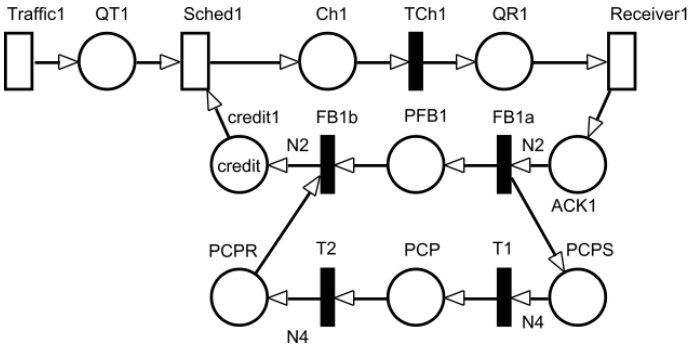


Fig. 1. credit-based flow control of one link and one flow. Independent flows have the same structure in parallel, but share the path from PCPS to PCPR. The tokens in PCP represent a real control information packet.

Figure 1 shows the DSPN model of CBFC per flow for one hop. Here only one flow is displayed, but all of them are treated the same way. Only NRT traffic needs to be treated by flow control, since the load of RT is below congestion by the help of CAC. BE traffic is generated in transition $Traffic1$ and queued in $QT1$. The scheduler $Sched1$ multiplexes the packets (tokens) onto the link (channel) $Ch1$, which is later modeled according to wireless link properties. Scheduling happens hierarchically and RT traffic has priority over NRT traffic [26]. The remaining link capacity is shared by all NRT flows and the subscheduler distributes them in a round robin (RR) manner. Note that this scheduler has a complexity of O(1). For simplicity the RT traffic is not shown here, as it is also not subject to CBFC. Instead, the timing behavior of $Sched1$ captures the baseline load situation. Also on the sender side **S**, there must be at least one token in $credit1$ to allow the sending of a packet. There are no further limitations and when there are lots of credits available, the instantaneous rate can be up to the link rate, without any slow start known from TCP. On the receiver side **R**, packet tokens are processed by $Receiver1$. A firing of this transition means that the packet has been *forwarded* to the next hop, i.e. left the buffer $QR1$ in the receiving node. Once a packet has been forwarded, a new token is created in place $ACK1$ and sent back to the source in order to fill up the credit place again. The circuitry between $ACK1$ and $credit1$ cares for aggregation of feedback, so that the control overhead of sending feedback can be reduced.

Figure 1 shows only one hop, but all the hops have the same behavior, i.e. a closed loop between adjacent nodes (BS, RN or UT). Lost packets must still be handled by an underlying ARQ mechanism, which can cooperate with CBFC by using the same sequence numbers. Lost credit messages are dealt with by not sending incremental credits, but rather absolute counters. So the next arriving credit message will always backup (and obsolete) an older one.

For the following discussion the symbols are explained in Table I. The receiving node **R** has a total available total buffer of size l_l (l for total link). Each flow v has an individual logical buffer size l_v (v for virtual connection). The total buffer memory is partitioned into pieces for each l_v, but this can be overlapping to achieve buffer sharing ($\sum l_v > l_l$). There is a separate flow control loop for the total buffer as well in this case. The limits l_v and l_l are known at setup time and installed

TABLE I
TABLE OF SYMBOLS

v	flow index, connection number
l_l	total buffer of size in receiver side
l_v	logical buffer size per flow v
c_l	initial credit for total link
c_v	initial credit per flow: $c_v := \#credit_v$
$N2$	decimation parameter per flow
$N4$	packing parameter, records per packet
t_v	counter of transmitted packets (flow)
t_l	counter of transmitted packets (link)
f_v	counter of forwarded packets (flow) $vf_v := \#ACK_v$
f_l	counter of forwarded packets (link)
k_v	packets in flight on the link in place Ch_v: $k_v := \#Ch_v$
pr	prepared credit records: $pr := \#PCPS$
cp	credit update message packet in PCP: $cp = \#PCP$
rr	received records in $PCPR$: $rr := \#PCPR$
r_v	credit update records in PFB_v: $r_v := \#PFB_v$

as initial credit c_v (c_l). In a real implementation, **S** increments a counter of transmitted packets t_v and **R** increments a counter r_v of packets that forwarded to the next hop. This counter is sent back to **S** in regular intervals (each $N2_v$ values), where it is used to update the state for all included connections v. During this update the new contents f_v^{new} of each record replace the previois value f_v.

Eq. 2 determines the current credit for connection v, and respectively in Eq. 3 for the link.

$$credit_v = c_v = l_v - t_v + f_v \geq 0 \tag{2}$$

$$credit_l = c_l = l_l - t_l + f_l \geq 0 \tag{3}$$

This system is implemented in each hop and the connection between the nodes are constructed by arcs which contain buffering places to store tokens which are currently on the fly. For consistent PN the weighted sums of tokens on each loop is constant:

$$\begin{aligned} \forall v : k_v + m_v + f_v + N2 \cdot r_v + c_v &= l_v & (4)\\ \sum (k_v + m_v + f_v + c_v) + & & (5)\\ N2 \cdot (\sum cr_v + rr + pr + N4 \cdot cp) &= \sum l_v \end{aligned}$$

This guarantees that no place in the loop can have more tokens than there are initial credits ($c_v = l_v$) in the credit places.

The buffer buffer places m_v thus cannot overflow, because they contain at most l_v packets and the whole queue usage is bounded by connection limits and the link limit.

$$l_{max} := \max(\sum m_v) = \min(\sum_{\forall v} l_v, l_l) \tag{6}$$

Bounded buffer contents means overflow protection and therefore no packet loss here.

The algorithm is not very complex. There are three integer counters required per flow on each node, and two are already there as ARQ sequence numbers. To calculate the memory usage, this must be multiplied with the number of flows. On wireless links there are usually not that many. For now five flows per UT or 500 flows per radio cell are realistic. This only adds up to a few kilobytes needed in a BS, which is small compared to the packet buffer itself. The packet buffer should be dimensioned to be able to store several times the bandwidth·delay product of the hop before and after.

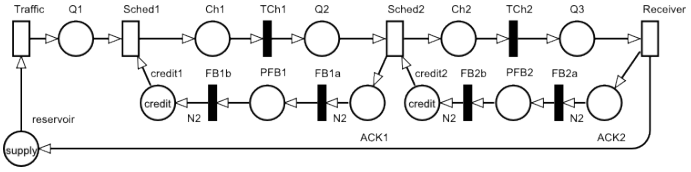


Fig. 2. A two hop wireless system, representative for a relayed or mesh network using SPN elements to represent the credit-based flow control

The transition types in Figure 1 determine the timing behavior. Except the "schedulers" all of them are immediate and do not contribute to the latency. The sources of latency are the link propagation delay and the waiting for multiple-weighted input arcs to gather enough tokens for the adjacent transition to fire [9].

IV. WIRELESS FLOW CONTROL ON MULTIPLE HOPS

In this paper a wireless relay scenario with two wireless hops is assumed, but it is representative even for more hops. The SPN in Figure 2 shows the system for one flow. The first wireless node (the base station) with buffer $Q1$ and scheduler $Sched1$ is connected to the source using a wired link which is assumed to have a much higher capacity than the wireless links. This is one of the reasons why the buffer in $Q1$ can quickly overflow.

The exponential transition $Sched1$ models the fluctuations of the channel, i.e., varying remaining capacity due to channel variations and higher priority RT traffic that is not modeled here explicitly. The next step consists of modeling the wireless channel, shown in Figure 3(a). The channel state is modeled by the two-state Gilbert-Elliot (GE) channel model, also known as on-off Markov channel model [27]. Its SPN model [28] using two places $Ch1off$ and $Ch1on$, and the state changes from 'good' to 'bad' and reverse by the transitions $Ch1dn$ and $Ch1up$ with transition firing rates λ_{10} (into fading) and λ_{01} (recovery). The rates λ_{10} and λ_{01} can be calculated from the usual parameters P_{error} and T_{period} by Eq. 7-8. Both channels are independently fading.

$$\lambda_{Ch1dn} = (T_{period} \cdot (1 - P_{error}))^{-1} \tag{7}$$

$$\lambda_{Ch1up} = (T_{period} \cdot P_{error})^{-1} \tag{8}$$

The packet interarrival time is exponentially distributed. Tokens (packets) generated in transition $Traffic$ come into queue $Q1$. $Q1$ is located in the BS, before the first bottleneck hop. From the scheduler $Sched1$ on packets are transmitted on the channel (place $Ch1$) and upon reception are queued in $Q2$, which is located on the intermediate hop (relay, RN). The access to the second hop is controlled by the scheduler $Sched2$. When $sched2$ grants access, the packet in transmitted on the channel (place $Ch2$) and queued in $Q3$ upon arrival. The application response ($Receiver$) consumes the packet and triggers the credits to flow towards the sender, hop by hop. Both scheduler transitions are only allowed to fire so many times as there are credit tokens available. By this way a packet loss by buffer overflow (due to a slower link downstream) is impossible.

For numeric analysis, the total state space must be bounded. Therefore the supply place P_{sup} is introduced without limiting the original system. The supply token count s should be chosen $10x$ larger than the expected queue occupancy in Q.

The timing behavior is determined by the type of the transitions. Except the "schedulers" all of them are immediate and do not contribute to the latency. The sources of latency are the link propagation delay and the waiting for multiple-weighted input arcs to gather enough tokens for the adjacent transition to fire [9]. The total delay is influenced by all queues on the path. By using the *p-invariant* of the main loop, the expression can be simplified this way:

$$\#Q1 + \#Q2 + \#Q3 + \#Ch1 + \#Ch2 + \#P_{sup} = s \Rightarrow \quad (9)$$

$$E[\#Q1 + \#Q2 + \#Q3 + \xi] = s - E[\#P_{supply}] \quad (10)$$

Figure 3(a) also shows that the buffer Q1 in the base station is not protected from overflow. However, as the wireless links are naturally the bottleneck, Q1 is where large packet backlogs arise. In the congestion situation they overflow and lead to non-tolerable packet loss. In case Q1 is dimensioned very large, the loss is only postponed and the resulting packet delays would make TCP assume there has been a loss anyway. Thus, simple oversizing the buffers is not the solution [29].

Clearly the flow control loops should be extended towards the source node (traffic source, application server). This avoids any buffer overflow in any node of the network and provides a reliable communication for BE traffic in a congested network. This will become an important aspect in the future, when traffic grows beyond capacity. In this case, with fully extended flow control, buffer sizes do really matter, and only in this case without TCP ACK timeouts and rate control, the rule is that more buffer gives more benefit.

Extending the flow control to the source by means of another credit-based hop appears natural, but it happens outside the scope of equipment of newly defined wireless communications. Therefore either a new IETF protocol must be defined that is implemented by all vendors, or an interface is implemented in the BS which translates the flow control behavior into existing (TCP) protocol messages.

UDP cannot be controlled like this. But this is alright, because UDP is preferable for RT traffic which is preferred by priorities, limited by CAC and cannot make use of overly delayed packets anyway.

The classical end-to-end TCP control is not very suitable for the wireless domain [30]. However, as long as a new protocol is not established, an interface can be built in the BS by generating artificial TCP ACK messages that are able to throttle down the source (deep packet inspection and injection). This works by extending the BS (eNB in LTE-A) functionality to ISO/OSI layer 4, where TCP ACK messages can be created by the BS (instead of the end point in the UT). These TCP ACK messages contain acknowledgment number and window size field which can be used to throttle the incoming traffic flow or even exert a finer grain rate control. ACKs from downstream will then be gracefully ignored.

The implementation complexity and computational cost is not high when compared to what progressive algorithms in PHY layer require, e.g., for MIMO, CoMP or MAC-layer scheduling with NP-hard optimization. Here the number of flows is proportional to the number of UTs in a radio cell. For each flow there is only need for 3 integer counters which consume much less memory than the packet buffers. Obviously, incrementing one counter per transaction (transition firing) is no significant effort.

V. Performance Results

The models in Figure 3(a), 3(c) and 3(e) have been analyzed using tool-supported Markov chain analysis of the SPN. The scenarios are representative and different parameters from the ones chosen here do not influence the principles of flow control (like deterministic boundedness of queues). For simplicity of the model, the parameters used here are fixed as $credit = 10$, $N2 = 5$. The channel capacity is fixed by the inverse of $T_{Sched1} = T_{Sched2} = T$. The channel coherence time is $T_{Ch1up} = T_{Ch2up} = 2 \cdot T$, $T_{Ch1dn} = T_{Ch2dn} = 2 \cdot T$. All results in Figure 3 are shown by the complementary cumulative distribution function (CCDF) of the buffer contents in all relevant queues at a load of $\rho = 0.6$, which causes temporary congestion when the channel is in off state. The value of T is not relevant because all transition rates are defined relative to each other. Thus $T = 1\mu s$ is a reasonable assumption.

The result graphs on the right of Figure 3 show the CCDF of the packet buffer occupancies. Results with this accuracy (10^{-7}) can only be obtained by analytic means and not by simulation. This is one reason for the SPN model and Markov analysis approach chosen here.

Figure 3(b) shows the CCDF for the closed loop wireless flow control, but open loop wired link, where packets queue (potentially unbounded) in the BS, i.e. before the bottleneck. This is the the reference situation and the motivation for this paper. As it can be seen, the buffer Q1 is potentially unbounded. With increasing load this quickly builds up to a significant backlog of packets and, of course, overflow at a certain point.

The credit based extension and complete closed hop-by-hop control in shown in Figure 3(c). The first hop credit has been chosen as 20 in order to model the larger capacity of the back-haul link. As Figure 3(d) shows, there is deterministically no buffer overflow. This behavior can be observed at any load and congestion situation. Therefore is is well suited for the reliable transport of connection-oriented protocol data. Backpressure is fed back to the source, and in the source up to the session and application layer, when all OS internal buffers are full. This way the tolerable rate can be calculated and even be used for application servers to adjust application-specific settings like, e.g., source coding rate and video frame rate. Also, the result can potentially be used for download proxy selection, P2P peer selection or max-capacity routing.

With TCP in place and without this CBFC first hop, an interface is needed between TCP and CBFC. The SPN in Figure 3(e) was constructed for this purpose, where the TCP source can be rate controlled to four rate levels, depending on the measured level in $\#Q1$. T0 sends at full rate, T1 at half, T2 at quarter, and $rate = 0$ when $\#Q1$ approaches limit 20. Figure 3(f) shows its performance and the buffer protection of Q1 can be observed. However, this approach is more complex to implement and requires rate and window size calculations on the path. Also, it is a hard way to achieve agreement to an RFC document in the IETF.

When comparing solutions (c) and (e), both can guarantee the buffer protection However, the control in (e) tends to stay on a reduced rate level, whereas credit based control allows full

ramp-up and rate as soon as the downlink buffer announced space for further packets. Thus CBFC allows fast and precise ramp-up, whereas the traditional TCP slow-start may even be counter-productive.

VI. Conclusion

In this paper the problem of congested links under heavy traffic and wireless links is addressed for the best effort traffic category in future wireless networks. The solution is a credit-based hop-by-hop flow control protocol, potentially with TCP coupling or without TCP congestion control at all. The system becomes stable regarding buffer usage, independently of traffic conditions and wireless link state. A traffic separation into real-time (RT) and non-real-time (NRT) or best-effort (BE) traffic is prerequisite. RT traffic is treated with standard QoS-aware scheduling algorithms and subject to CAC. BE traffic can now be allowed to use up the available capacity completely. Fair schedulers, e.g. round robin, are recommended.

The flow control protocol proposed in this paper avoids buffer overflows completely. The main contribution of this paper is the extension of this protocol from the wireless domain to the source. It provides a model for coupling with TCP and has the potential of even replacing TCP. Numeric results by stochastic Petri net analysis prove that buffers cannot overflow wherever there is a closed loop control around the corresponding hop. A proposed TCP interface is also viable, implemented in the base station. It can be applied for multiple hops and even mesh networks, as long as a flow establishment [31] is used which exchanges the WFC parameters in the initial handshake protocol.

Future work will make more use of SPN for all aspects of wireless systems and can provide analytic/numeric results where otherwise only simulation would be used.

The topic of sustainable networks is becoming more and more important, i.e. how to operate networks which can cope with any load and congestion situation. Future work should therefore also include the other aspects of demand-supply balancing [1] by end-to-end QoS, usage-based tariffs and green thinking before embracing all exponential growth.

References

[1] R. Schoenen, G. Bulu, A. Mirtaheri, and H. Yanikomeroglu, "Green communications by demand shaping and User-in-the-Loop tariff-based control," in *2011 IEEE Online Green Communications Conference (IEEE GreenCom'11)*, Online.

[2] Jon Postel, "RFC 793: Transmission Control Protocol," Internet Engineering Task Force, Tech. Rep., 1981. [Online]. Available: http://www.ietf.org/rfc/rfc793.txt

[3] M. Scharf, "Comparison of end-to-end and network-supported fast startup congestion control schemes," *Computer Networks*, vol. 55, no. 8, pp. 1921 – 1940, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128611000491

[4] M. Honda, Y. Nishida *et al.*, "Is it still possible to extend TCP?" in *Proceedings of the ACM Internet Measurement Conference (IMC)*, Nov 2011, p. 6.

[5] M. Proebster, M. Scharf, and S. Hauger, "Performance Comparison of Router Assisted Congestion Control Protocols: XCP vs. RCP," in *Proceedings of the 2nd International Workshop on the Evaluation of Quality of Service through Simulation in the Future Internet - Held in conjunction with SIMUTools 2009*, March 2009.

[6] S. Hauger, M. Scharf, J. Kögel, and C. Suriyajan, "Evaluation of Router Implementations for Explicit Congestion Control Schemes," *Journal of Communications*, pp. 197–204, March 2010.

[7] R. Pabst, B. Walke, D. C. Schultz, H. Yanikomeroglu *et al.*, "Relay-based deployment concepts for wireless and mobile broadband radio," *IEEE Communications Magazine*, pp. 80–89, Sep 2004.

[8] H. Kung and S. Wang, "Zero Queueing Flow Control and Applications," in *Proceedings of the IEEE INFOCOM*, 1998.

[9] R. Schoenen, G. Post, and A. Müller, "Analysis and dimensioning of credit-based flow control for the ABR service in ATM networks," in *Proceedings of the IEEE GLOBECOM*, 1998, vol.4 p.2399.

[10] T. Weerawardane, R. Perera, and C. Goerg, "A Markov model for HSDPA TNL flow control and congestion control performance analysis," in *Proceedings of IEEE VTC Spring*, Budapest, May 2011.

[11] R. Schoenen, "Credit-based flow control for multihop wireless networks and stochastic Petri nets analysis," in *Proceedings of the CNSR*, Ottawa, May 2011.

[12] T. Murata, "Petri Nets: Properties, Analysis and Applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–581, April 1989.

[13] M. Marsan, *Modelling with Generalized Stochastic Petri Nets*. Wiley, 1996, iSBN 0-471-93059-8.

[14] R. German, "A toolkit for evaluating non-markovian stochastic Petri nets," *Performance Evaluation*, vol. 24, pp. 69–87, 1995.

[15] J. Billington *et al.*, *Application of Petri Nets to Communication Networks*. Springer, 1999, ISBN 3-540-65870-X.

[16] L. Lei, C. Lin, J. Cai, and X. Shen, "Performancs analysis of wireless opportunistic schedulers using stochastic Petri nets," *IEEE Transactions on Wireless Communications*, vol. 8, no. 4, April 2009.

[17] S. Geetha and R. Jayaparvathy, "Modeling and Analysis of bandwidth Allocation in IEEE 802.16 MAC: A Stochastic Reward net Approach," *Int. J. Communications, Network ans System Sciences*, vol. 3, no. 7, pp. 631–637, July 2010.

[18] R. Jayaparvathy, S. Anand, S. Dharmaraja, and S. Srikanth, "Performance Analysis of IEEE 802.11 DCF with Stochastic Reward Nets," in *International Journal of Communication Systems,*, vol. 20, no. 3, 2007.

[19] R. Gaeta, M. Gribaudo, D. Manini, and M. Sereno, "On the use of Petri nets for the computation of completion time distribution for short TCP transfers," *Proceedings of the 24th international conference on Applications and theory of Petri nets*, vol. LNCS, pp. 181–200, 2003.

[20] J. Monserrat, P. Sroka, G. Auer, J. Cabrejas, D. Martin, A. Mihovska, R. Rossi, A. Saul, and R. Schoenen, "Advanced radio resource management for IMT-Advanced in WINNER+ (II)," in *Proc. ICT-MobileSummit 2010*, Florence, Italy, Jun 2010.

[21] R. Schoenen, A. B. Sediq, H. Yanikomeroglu, G. Senarath, and Z. Chao, "Fairness analysis in cellular networks using stochastic petri nets," in *Preceedings of PIMRC'2011*, Toronto, Canada, Sep 2011.

[22] V. Živojnović, R. Schoenen, and H. Meyr, "On Retiming of Multirate DSP Algorithms," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. VI, Atlanta, May 1996.

[23] *Quantum Flow Control, Revision 2.0.5*, The Flow Control Consortium, Mar. 1997, http://www.qfc.org/.

[24] *Controlled Transfer*, ITU-T, 1999, iTU-T Q7/13 section 6.X.

[25] H. Kung and R. Morris, "Credit-Based Flow Control for ATM Networks," *IEEE Network Magazine*, vol. 9, no. 2, pp. 40–48, March/April 1995.

[26] R. Schoenen and A. Otyakmaz, "QoS and Flow Management for Future Multi-Hop Mobile Radio Networks," in *Proceedings of the IEEE VTC Fall*, Ottawa, Canada, Sep 2010.

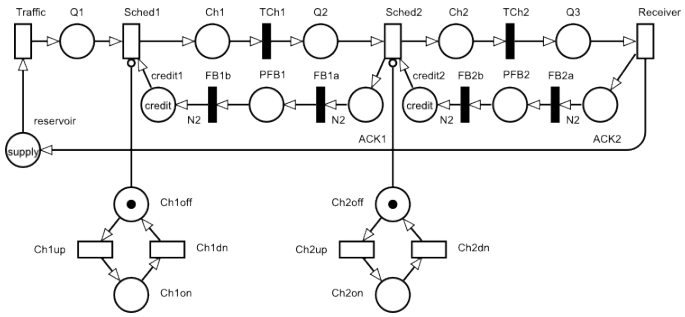
[27] H. Wang and N. Moayeri, "Finite-State Markov Channel – A Useful Model for Radio Communication Channels," *IEEE Transactions on Vehicular Technology*, vol. 44, no. 1, pp. 163–171, Feb 1995.

[28] R. Schoenen, M. Salem, A. Sediq, and H. Yanikomeroglu, "Multihop wireless channel models suitable for stochastic Petri nets and markov state analysis," in *Proceedings of IEEE VTC Spring*, Budapest, May 2011.

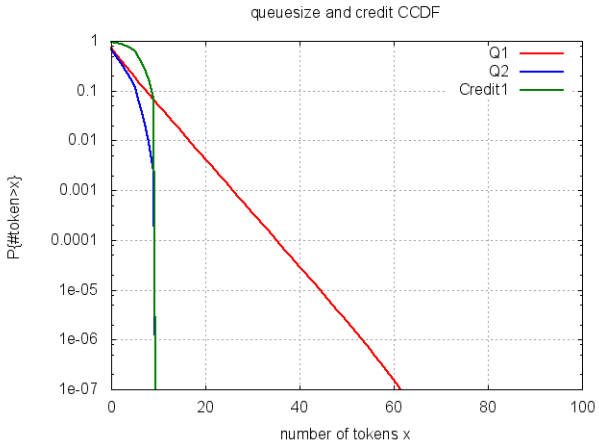
[29] J. Gettys, "Bufferbloat - Dark Buffers in the Internet," IEEE Internet Computing, May 2011, http://www.bufferbloat.net/. [Online]. Available: http://www.bufferbloat.net/

[30] M. Malkowski and S. Heier, "Interaction between UMTS MAC Scheduling and TCP Flow Control Mechanisms," in *Proceedings of 2003 International Conference on Communication Technology*, Beijing, China, Apr 2003, pp. 1373–1376.

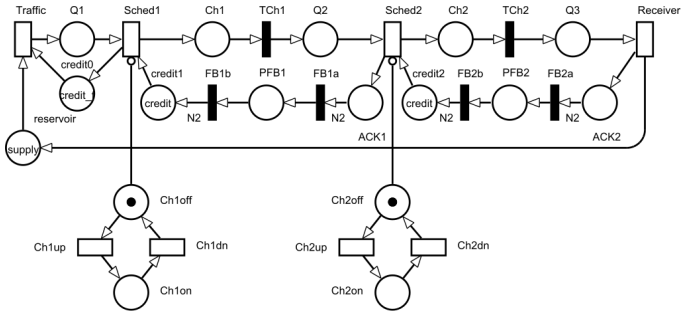
[31] A. Otyakmaz, D. Bültmann, R. Schoenen, and I. Durmaz, "On Flow Management for Future Multi-Hop Mobile Radio Networks," in *IEEE WiCom*, Beijing, China, Sep 2009.



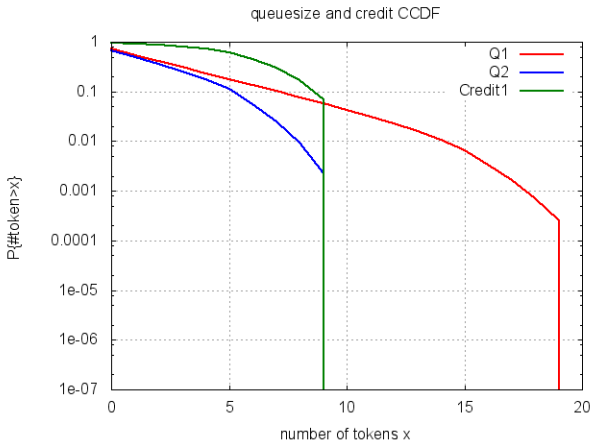
(a) SPN model for two wireless hops and open wired back-haul hop



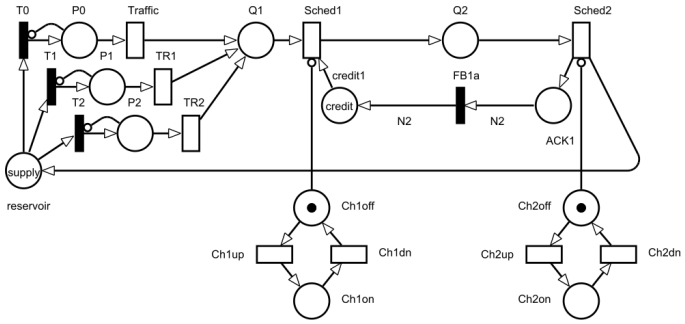
(b) results for open hop show regular behavior in Q1



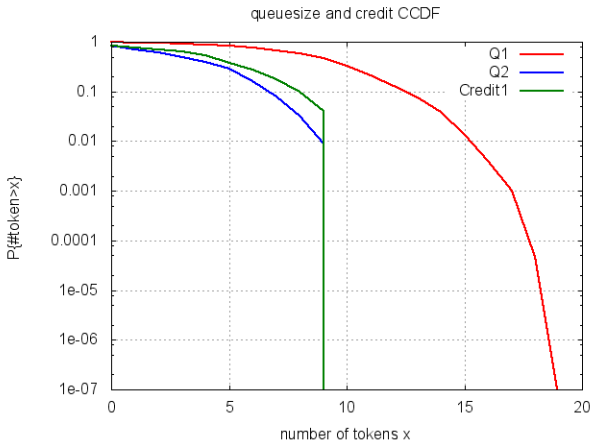
(c) Full hop-by-hop flow control including wired Internet link



(d) results show bounded buffer occupancy in all hops



(e) TCP rate control of source, controlled in BS to protect Q1



(f) full control is possible with TCP injection

Fig. 3. SPN models for the uncontrolled wired link (first hop) and the proposed extensions plus their results at load $\rho = 0.6$. The principle boundedness of buffers with CBFC is valid in all other load situations as well.