Credit-Based Flow Control for Multihop Wireless Networks and Stochastic Petri Nets Analysis

Rainer Schoenen\textsuperscript{1,2}

\textsuperscript{1}Communication Networks (ComNets), Faculty 6, RWTH Aachen University, Germany
\textsuperscript{2}Department of Systems and Computer Engineering, Carleton University, Canada
rs@sce.carleton.ca

Abstract—As the increase of data traffic is expected to grow faster than wireless capacity, dealing with congestion is unavoidable in the future. The capacity imbalance between wired and wireless links leaves a bottleneck on the wireless link while traffic keeps coming in through the bigger pipe. In current systems, the dynamics of a wireless channel and the variations in the higher priority real time traffic both make full link utilization impossible. Either the buffers are full, giving a full link but resulting in packet losses in the queues or the link is underutilized.

Closed loop credit based flow control can solve this problem by avoiding any buffer overflow completely and providing enough packets in the buffers to ramp up immediately when the channel has remaining capacity. In this paper a model using generalized stochastic Petri nets (SPN) is introduced. By structural analysis the boundedness of memory is proven while a Markov state analysis of the SPN provides numeric performance results. The studied scenario includes two wireless relayed hops, a wired backhaul and traffic class separation.

Keywords—stochastic Petri nets, wireless flow control, WFC, credit based, congestion, relay, multihop

I. INTRODUCTION

Wireless systems (cellular, ad-hoc and mesh) all have capacity limits which are significantly lower than what the fixed network backhaul can deliver. In the future the traffic generated and consumed by wireless terminals will increase faster than the capacity can grow due to new generations of radio technologies. Inevitably this will lead to congestion of the wireless links, at least during the busy hours. Real-time traffic will be saved from suffering by having a higher priority and limited load due to call admission control (CAC). So the remaining task needs to be solved: How is the non-real-time traffic protected from loss by buffer overflow, how is fairness among the congested flows guaranteed, how to avoid starvation of queues and how to best cooperate with the end-to-end TCP protocol for flow control and rate adaptation.

In cellular wireless systems we already have the notion of a flow [1] which is set up during connection establishment and whose QoS data is known to the MAC layer by means of crosslayer control plane signaling.

In this paper a flow control protocol is proposed for the traffic classes that are subject to congestion, especially the best effort (BE) class. This wireless flow control (WFC) is credit based [2], as this provides precision, stability and deterministic behavior [3] [4]. Known from the ATM ABR service, it is a new concept for wireless systems. However, it allows to safely drive the BE traffic into congestion and therefore cope with the future traffic load situation. Its structure is modeled using the Petri net approach, and performance results come from its stochastic analysis.

Petri nets [5] are an expressive and versatile tool for modeling and analysis of stochastic systems. They combine the flexibility of a Turing-complete automaton with the power of stochastic Markov chain analysis. Stochastic Petri nets (SPN) and generalized SPN (GSPN) [6] have become a useful tool for adept researches in computer science. Useful tools for GSPN analysis exist [7]. In recent years, GSPN have been used occasionally to model communications systems [8], [9], but a widespread use is not observed. Some examples are IEEE 802.16 [10] and IEEE 802.11 [11]. Links to higher layers exist in works for TCP models [12]. And there is still a high demand for recent technologies like the IMT-Advanced systems [13]. Especially ISO/OSI layer two (medium access control) can benefit from GSPN analysis methods. Most works study the maximum throughput only with a full buffer assumption. However, this is already in overload condition and either doesn’t provide finite delays or drops lots of packets (finite buffer).

The paper is organized as follows: Section II introduces the basics of stochastic Petri nets. The following sections introduce the flow control protocol model, a scheduler and a channel model. A wireless system model using this flow control on two hops is treated in section VI. Performance results are presented in section VII before ending with the conclusion.

II. PETRI NETS

Petri nets (PN) is a graphical and mathematical tool suitable to model complex systems with a state. Systems can be be described and studied when they are concurrent, asynchronous, distributed, deterministic or stochastic. Sub-classes of PN are finite state machines and marked graphs used for DSP algorithms [14]. Many aspects of flow charts
Elements of stochastic Petri nets.

Stochastic Petri nets (SPNs) extend the paradigm to model time, essentially by assigning each transition a firing rate (in the continuous time case (CT)) or a firing probability (in the discrete time case (DT)). Here we will focus on the CT case only. The firing rate is basically: A transition can be fired if all places connected to input arcs contain enough \( w_{ij} \) tokens and all places connected via disabling arcs are empty. For further details of firing rules, liveness and reachability see [5].

A. Stochastic Petri Nets

Stochastic PN (SPN) extend the paradigm to model time, essentially by assigning each transition a firing rate (in the continuous time case (CT)) or a firing probability (in the discrete time case (DT)). Here we will focus on the CT case only. Then the firing rates are given by \( \Lambda = \lambda_1, ..., \lambda_m \) which can be marking-dependent. \( \lambda_j \) is the inverse of the average firing time \( t_j \). For the state change. The matrix \( R \) of all arc rates can then be used to calculate the steady-state solution of the MC and from that all other performance metrics can be obtained.

Queueing models and networks are a subset of the modeling power of SPN, and results for queue length and waiting times can easily be derived [15], that is why SPNs are becoming increasingly popular for modeling communication networks [8]. SPN can be used as well for modeling packet level and flow level, where a constant fluid of data in \( \text{bits}/s \) is assumed while being in a state.

III. CREDIT-BASED FLOW CONTROL

In this paper a credit-based flow control protocol is proposed as known from the ATM domain [2] [3]. For the basic operation each link is considered to be a closed loop between adjacent nodes (BS, RN or UT). Peers are called S for
sender and R for receiver here. For each loop, the receiving node R has an available total buffer of size \( l_l \) (limit_{link}). This buffer is partitioned into individual memory \( l_v \) (limit_{v}) for each flow or connection \( v \) out of \( C \) BE connections through that node. This partitioning is only logical and can be overlapping, i.e. buffer sharing is possible with \( \sum l_v > l_l \). Both buffers are protected by separate flow control loops, one per flow and one total. The sender S knows about each limit \( l_v \) (\( l_l \)) and installs this as the initial credit \( c_v \) (\( c_l \)).

Packets (or e.g. LTE-A resource elements) may now be sent arbitrarily (after the scheduler has served all higher-priority packets) as long as the current credits \( c_v \) (\( c_l \)) are positive (non-zero). For each sent packet a counter of transmitted packets \( t_v \) is incremented (from zero initially). R increments a counter \( r_v \) of packets that are received and forwarded to the next hop. In regular intervals (each \( N2_{VC} \) values) this counter is sent back to the source S which updates its state for all included connections \( v \) by replacing the previous values of forwarded packets, \( f_v \), with the new contents \( f_{new}^v \) of each record. The current credit for connection \( v \) is given by eq. 2 and similar for the link \( l \) in eq. 3:

\[
\text{credit}_v = c_v = l_v - t_v + f_v \geq 0 \quad (2)
\]

\[
\text{credit}_l = c_l = l_l - t_l + f_l \geq 0 \quad (3)
\]

By using absolute counters instead of tokens in the real implementation, a loss of credit messages on the wireless link is not a serious problem, as the next counter update is enough to continue.

In [3] a Petri net model was introduced which models all necessary details of one flow control loop. Here it is applied for wireless links with fading channels and in a multihop environment. Figure 2 shows the SPN model of one flow controlled link for two flows. Additional hops seamlessly integrate to the right of this model, as shown in section VI.

The main components in the forward direction are the packet queue memory \( QT_v \) (the instantaneous contents denoted by \( m_v := \#QT_v \) packets) and the \( k_v \) packets in flight on the link in place \( Ch_v \) (\( k_v := \#Ch_v \)). A packet of flow \( v \) is only allowed to be sent by the scheduler (see section IV) if there are tokens in the credit place \( \text{credit}_v \) (\( c_v := \#\text{credit}_v \)). In backward (feedback) direction \( f_v \) forwarded tokens queue in place \( ACK_v \) (\( f_v := \#ACK_v \)) until a record in \( PCPS \) is complete (\( pr := \#PCPS \)). \( N4 \) of such records build a credit update message packet in \( PCP \) (\( cp := \#PCP \)), which is again decomposed in the upstream node into records in \( PCPR \) (\( pr := \#PCPR \)). \( PFB_v \) exists for distinguishing the flows and holds the number of credit update records for this flow \( v \). It holds \( r_v := \#PFB_v \) tokens.

There is one per-flow flow control loop (here \( C = 2 \) connections), using the feedback quantization constants \( N2 \) discussed in [3] to decrease the feedback frequency. Each link is can also be controlled by a link flow control loop, quantized with \( N2_{link} \) (N2L). This is omitted here but described in [4]. The packing into a feedback message consisting of \( N4 \) records reduces the flow control bandwidth overhead, which is discussed into details in [3].

Let this system be implemented in each hop and the connection between them be made by arcs which have buffering places to hold tokens which are currently on the fly. The consistency of each loop implies [16] that the weighted sums of tokens (Eq. 4-5) on them are constant:

\[
\forall v: k_v + m_v + f_v + N2 \cdot r_v + c_v = l_v \quad (4)
\]

\[
\sum (k_v + m_v + f_v + c_v) + N2 \cdot (\sum ar_v + rr + pr + N4 \cdot cp) = \sum l_v \quad (5)
\]

Therefore there cannot be more tokens in any place of the loop than there are initial credits (\( c_v = l_v \)) in the credit places. Thus the buffer places \( m_v \) can also at most contain \( l_v \) packets each and the whole queue (buffer memory) usage \( l_{max} \) is bounded by both the sum of all relevant connection limits and the link limit.

\[
l_{max} := \max(\sum m_v) = \min(\sum l_v, l_l) \quad (6)
\]

The boundedness of buffer contents implies that no packet loss can occur due to buffer overflow. Packet losses on the link are accounted for by an underlying hop-by-hop ARQ mechanism. Actually ARQ and WFC can use and share the same sequence numbers. For the discussion of dimensioning and quantization effects see [3].

The complexity of the algorithm is not very high. It only requires three counters per flow for each intermediate node, and two of them are already present due to ARQ sequence numbers. The number of flows on wireless links is usually not high. Each UT might have 5 flows in parallel, which is the count on the last link. A link BS-RN might carry 500 flows in a cellular network, so the memory consumption is a few kilobytes which is small compared to the packet buffer itself. Each node should have a reasonable packet buffer of at least several times the bandwidth-delay product of the hop before and after.

The option to reduce \( l_{max} \) below the sum of the individual credits (“buffer sharing”) allows a tradeoff between memory and peer blocking probability [3].

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 [3].

Regarding the transition \( Sched1 \), we see the necessity separate the traffic into real-time (RT) and best effort (BE) by means of a static priority scheduler [17], which leads to the next section. Only BE traffic needs to be treated by flow...
Figure 3: SPN model of the packet scheduler separating RT (CBR/VBR) and NRT (BE) traffic by static priority. Fairness in BE is established by RR subscheduling and credits tokens enable flow control from downstream

Figure 4: SPN of the Gilbert-Elliot channel. The state #G1 can be used in state-dependent formulations of SPN properties.

control, since the load of RT is below congestion by the help of CAC.

IV. Hierarchical Scheduling

Figure 3 shows the SPN model of the packet scheduler assumed in each node. Note that packet scheduling is different and separable from resource scheduling on the wireless link [18]. Most important is the separation of RT and NRT (BE) traffic by a static priority scheduler (having \( O(1) \) complexity). Within the priority class there are subschedulers per class that treat all flows within the class according to their QoS requirements [17]. The RT traffic and scheduler is not treated here further, but there exist simple [19] and heavy solutions [20], FCFS being still the easiest with \( O(1) \) complexity and a good delay-fair behavior among flows. Within the BE class fairness is usually most important, especially in the congested region (full buffers). When no further weights are required, a simple round-robin (RR) scheduler does the job pretty well in \( O(1) \) complexity.

In Figure 3 also the interfaces to the flow control are shown by the credit places.

V. Channel Models for SPN Analysis

Wireless channels possess a number of properties and are influenced by many effects. Usually they are decomposed into antenna gain, pathloss, slow fading, fast fading, shadowing, multipath propagation effects. The simplest model suitable in a packet level SPN approach is the two-state Gilbert-Elliot (GE) channel, also seen as on-off MC. Its SPN equivalent is shown in Figure 4 using places \( G_1 \) and \( G_0 \), the token position in which marks the state as 'good' or 'bad'. The transition firing rates \( \lambda_{10} \) (into fading) and \( \lambda_{01} \) (recovery) model the state changes from 'good' to 'bad' and opposite. They are translated from the common parameters \( P_{\text{error}} \) and \( T_{\text{period}} \) by Eq. 7-8.

\[
\lambda_{10} = \left( T_{\text{period}} \cdot (1 - P_{\text{error}}) \right)^{-1} \tag{7}
\]

\[
\lambda_{01} = \left( T_{\text{period}} \cdot P_{\text{error}} \right)^{-1} \tag{8}
\]

The state itself can be used in another part of the SPN to influence a scheduler, e.g. switch between transmitting or deleting packets which are currently transmitted on the channel or by switching the available short-term capacity.

Other SPN models like the finite-state Markov channel (FSMC) [21] can be found in [22].

VI. Wireless Multihop Links Using Flow Control

In this section a scenario is analyzed which consists of two wireless hops, i.e. a relayed link. Figure 5 shows the basic SPN for one flow. Here we assume a system which can have wired links towards the source and destination, where the wired links are assumed to have a capacity much larger than the wireless links.

\( Sched_1 \) is an exponential transition which models the fluctuations in remaining capacity due to channel variations and higher priority RT traffic that is not modeled here explicitly. The traffic (packet) interarrival time is exponential, putting packets (tokens) into queue \( Q_1 \) before the first hop (imagine \( Q_1 \) in the BS). After admitted to the channel by the scheduler, packets traverse the channel place \( Ch_1 \) and are queued in \( Q_2 \) on the intermediate hop (relay, RN). The scheduler \( Sched_2 \) there controls access to the second hop. If admitted, the packet flows through \( Ch_2 \) to be queued in \( Q_3 \) shortly before the next link (wired) or application can make use of it.

The flow control works by sending feedback (backpressure) messages back to the sender. Only after a packed is further forwarded onto the next link or the application has consumed it, an ACK token is created and put into place \( ACK_1 \) (or \( ACK_2 \) on the second hop). As mentioned in section III, there is a quantization operation with \( N_2 \) in order to reduce the feedback overhead. But basically all ACK messages assemble again in the credit place after some time. The scheduler transition may only fire if there
is at least one credit token left. By this way a packet loss by buffer overflow (due to a slower link downstream) is impossible.

The supply place $P_{sup}$ is necessary to bound the state space. As a rule-of-thumb, the supply $s$ should be ten times larger than the expected queue occupancy in $Q$.

Figure 6 shows the system model including two independent wireless fading channels. The channels are modeled as two separate subnets of type as in Figure 4, where the marking #Ch1off controls the disabling arc to the scheduler place, effectively stopping all transmissions. Both channels are independently fading.

For measurements the following equation derived from the $p$-invariant of the main loop is useful in order to simply determine the total delay:

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

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

VII. NUMERIC PERFORMANCE RESULTS

The system model has been analyzed by Markov chain analysis of the underlying MC (defined by the RG) of the SPN. The parameters used for this numeric example are: $\text{credit} = 10$, $N2 = 5$; the channel coherence time is specified by $T_{Ch1up} = T_{Ch2up} = 2\cdot T$, $T_{Ch1dn} = T_{Ch2dn} = 2\cdot T$; The channel capacity is defined by $T_{Sched1} = T_{Sched2} = T$. The traffic is varied between 'no traffic' ($\rho = 0$) and 'congestion' ($\rho = 1$) by $T_{Traffic} = T/\rho$. $T$ itself does not play a role since all transition rates are defined relative to each other, so $T = 1/\mu s$ is a good choice.

Figure 7 shows results for the study of different traffic arrival rates to the systems, normalized to the capacity of the system. The results reveal the queueing nature of random traffic with a server given by the wireless links. At a certain maximum load, the queue $Q1$ grows asymptotically to infinity, a condition usually called unstable overload. However, the flow control cares for the fact, that there is no overflow in the wireless queues $Q2$ and $Q3$.

Because all the MC state probabilities are known after analysis, it is easy to also obtain the probability density function (PDF) or CDF of the number of tokens in any place and therefore PDF and CDF of packet queue occupancy and packet delays.

Figure 8 shows results of the multihop model of Figure 6 with fading channels on both hops. Due to the parameter settings, both wireless channels can only carry 50% of the capacity of the example before. Therefore the offered load has been normalized to account for this fact ($\mu_{max} = 1/(2T)$).

The results show that only the first queue $Q1$ grows with increased traffic, because it is not within a flow control loop in this example. All other queues on the wireless nodes are bounded by construction (here: $\#Q2 \leq l_v = 10$). The total number of packets in flight $Qt$ is obviously dominated by $Q1$. The number of tokens in the credit places #credit (C1,C2 in the graph) do not deplete so soon and even have 5 on average in the congested situation. Even if the second link is completely blocked, the scheduler Sched2 would stop and soon the credits would drop to zero to also stop Sched1.

Figure 9 shows the complementary cumulative distribution functions (CCDF) of the token distribution in the queues $Q1$, $Q2$ and the credit place C1. More than the average results in Figure 6, this shows that the buffer memory in $Q2$ does never exceed the upper bound given by the initial credit.

At this point it becomes clear that an ideal wireless network should also extend the flow control loops to the links (and hops) delivering the packets to the first wireless node (base station). This would totally avoid any buffer overflow in any node of the network and provide a reliable communication for BE traffic in a congested network. This will become an important aspect in the future, when traffic grows beyond capacity.

As this credit-based flow control is rather new in this
context, it cannot be expected that this concept will be adopted also for the wired links so soon. The classical end-
to-end TCP control is not very suitable for the wireless
domain [23]. However, a solution exists 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
flags that can be used to throttle the incoming traffic flow
or even exert a finer grain rate control.

VIII. CONCLUSION

This paper addresses the problem of congested traffic of
the best effort category in future wireless networks. The
proposed solution consists of a traffic separation of real-time
(RT) and non-real-time (NRT) or best-effort (BE) traffic,
so that RT traffic can be treated with standard QoS-aware
scheduling algorithms and limits by CAC, and BE traffic is
allowed to use up the available capacity completely. This can
be achieved by fair schedulers, e.g. round robin, and a flow
control protocol that avoids buffer overflows completely. A
modeling approach using stochastic Petri nets (SPN) was
proposed for studying the performance of the protocol in
wireless systems. The main contribution is the proposal
to use this protocol know from the ATM domain also in
wireless networks and its stochastic Petri net (SPN) model.
More complex system components can be added easily in
the SPN paradigm. It is convenient to plug in this flow
control model into a bigger system model and concentrate
on other interesting tasks on the MAC layer like resource
allocation, scheduling, and the study of packet delays. Also
it is straightforward to extend the two hops analyzed here to
multiple hops and even towards mesh networks, as long as
a flow establishment [1] is used which exchanges the WFC
parameters in the initial handshake protocol. Future work
will make more use of this and can provide analytic/numeric
results where otherwise only simulation would have been
used. Also, the sustainability aspect of wireless communica-
tions requires a rethinking [24] towards coping with the
limitations instead of overprovisioning and flatrate tariffs.

REFERENCES

Networks,” in IEEE WiCom, Beijing, China, Sep 2009.
sioning of Credit-Based Flow Control for the ABR Service in
ATM Networks,” in Proceedings of the IEE GLOBECOM,
Flow Control with Internal Backpressure in Input and Output
tions,” Proceedings of the IEEE, vol. 77, no. 4, pp. 541–581,
April 1989.
[8] J. Billington et al., Application of Petri Nets to Communica-
Wireless Opportunistic Schedulers using Stochastic
Petri Nets,” IEEE Transactions on Wireless Communications,
vol. 8, no. 4, April 2009.


