A study of TCP performance in wireless environment using fixed-point approximation

Dmitri Moltchanov
Department of Communication Engineering, Tampere University of Technology, P.O.Box 553, Tampere, Finland
E-mail: moltchan@cs.tut.fi

Abstract
Loss of IP packets occurring as a result of imperfect local error correction is one of the major reasons for TCP performance degradation in wireless networks. These losses are misinterpreted by TCP senders as congestion indications and subsequently lead to the sharp decrease of the sending rate. In addition to wireless losses, packets can be lost as a result of buffer overflows at the IP layer. To study performance experienced by TCP sessions sharing a wireless channel, we adopt the fixed-point approximation. To do so, we distinguish between two modes of operation: (i) packet losses are mainly caused by imperfect error correction, (ii) the packet loss process is dominated by the buffer overflow. Applying different TCP models for these two regimes we approximate TCP throughput as a function of underlying layers’ parameters. Using this approach we study the effect of various protocol parameters on TCP throughput. We also investigate the effect of the queuing system involved in the FPA analysis and demonstrate that usage of complex models with correlated arrival processes and generally distributed service time does not qualitatively affect the estimated TCP throughput. This observation allows to use simple queuing models having closed form solutions for performance metrics of interest, e.g. M/M/1/K or Geo/Geo/1/K.

1 Introduction
Nowadays, TCP accounts for 80—90% percent of all the traffic in the Internet and there is no indication that its share will significantly decline in the future.
Predicting performance of TCP in various environments is crucial for better understanding, dimensioning and deployment of modern networks.

Taking into account negligible error rate of modern wired transmission media, performance degradation of TCP connections in wired networks mainly stems from buffer overflows. At the same time wireless channels are prone to incorrect reception of channel symbols occurring as a result of insufficient received signal strength. Although data-link error correction techniques such as forward error correction (FEC) and automatic repeat request (ARQ) partially correct these errors locally, some of them still propagate to higher layers resulting in loss of protocol data units (PDU). Thus, analyzing performance of TCP connections running over a wireless channel, we need to take into account packet losses occurring as a result of non-negligible bit error rate and those caused by buffer overflows.

The rationale behind this study is to provide a practical methodology to evaluate the effect of wireless channel characteristics, channel adaptation mechanisms and buffering process at the IP layer on the throughput of TCP connections sharing a wireless bottleneck. The proposed model is based on cross-layer modeling principles and consists of three basic steps: (i) wireless channel modeling, (ii) cross-layer extension of the wireless channel model to the layer of interest, and (iii) performance evaluation at the layer of interest. It is important to note that all these three steps are independent of each other meaning that a specific mathematical model used at each step of the modeling procedure can be replaced by a better one (e.g. simpler or more accurate) whenever needed. This adds to versatility of the proposed approach allowing to relax those assumptions taken in this work. The wireless channel is considered as a bottleneck implying that losses are induced by both buffer overflow and error-prone nature of the wireless medium. The proposed model allows to highlight many interesting features of TCP performance in wireless environment including influence of first- and second-order wireless channel statistics and configuration of the protocol stack at the air interface. We show that this influence is often too complicated to predict without detailed performance evaluation model. As wireless channel characteristics are expected to vary in time, the proposed model forms the basis for performance control system that needs to be implemented to provide the best possible throughput to TCP connections sharing the same wireless bottleneck.

We specifically stress that the model proposed in this paper was intentionally built to qualitatively study the effect of wireless channel statistics and various channel adaptation procedures on the throughput of TCP sources. For this reason we do not resort to a specific wireless access technology. Instead, we assume a certain set of properties inherent for wireless channels. However, as one will see in what follows, the modular structure of the pro-
posed framework makes our model versatile enough to use in quantitative performance evaluation studies of modern wireless access technologies.

The rest of the paper is organized as follows. Related work is reviewed in Section 2. The system model is introduced in Section 3. Service process of a wireless channel is derived in Section 4. The fixed-point approximation for wireless channels is elaborated in Section 5. Numerical results are presented in Section 6. Conclusions are drawn in the last section.

2 TCP modeling

TCP congestion and flow control procedures pose significant challenges for a performance analyst. Given the complexity of TCP window dynamics, exact models of its operation are often too complicated to be suitable for analytical analysis. Depending on the amount of details taken into account more or less accurate models have been formulated so far. Usually, these models approximate throughput of a TCP connection as it is the most important parameter for non-real-time applications. Researchers found out that it mainly depends on the round-trip time (RTT) and packet loss probability. Despite differences in approaches, all of those lead to the special cases of $1/\sqrt{T_R\sqrt{p}}$ law stating that TCP throughput is inverse proportional to the mean RTT, $T_R$, and square root of the packet loss probability, $p$, (see e.g. [29, 22]).

TCP models can be divided into three broad categories. These are models based on the renewal theory, processor sharing models, and those, based on the fixed-point approximation. The first group of models assumes that the network characteristics in terms of the packet loss probability and the end-to-end packet delay are known in advance and the connection of interest does not affect them. These assumptions often hold for high-speed backbone links/routers with a high degree of statistical multiplexing. The throughput of a TCP connection is then expressed as a function of these two parameters. Examples of such models include Mathis model [22], PFTK model [29], and various refinements of these two e.g. [10]. The former assumes that the packet losses are completely random. As a result, this model is suitable for those networks where routers implement active queue management (AQM) including random early detection (RED) and its variants. PFTK model assumes completely correlated environment where the loss of a single packet in the sending window leads to the loss of all subsequent packets in this window. This model is best suited for droptail queue management. Studies of the loss process in the Internet demonstrated that its characteristics are somewhere in-between two abovementioned extreme cases [35]. In [2], assuming a weakly stationary packet loss process the authors derived one of the most general re-
sults for TCP throughput. They further demonstrated that when the packet loss characteristics tend to be completely random their model degenerates to the Mathis one. One the other hand, for a highly correlated loss process it takes on properties of the PFTK model.

The major shortcoming of renewal TCP models is that they require exact knowledge of the packet loss probability and RTT. These values are rarely known in advance. Another important limitation of this class of models is that performance of a number of concurrent TCP connections sharing the same bottleneck is almost impossible to evaluate. The study of two competing TCP connections is the only available result in this category [3].

Another approach to TCP performance modeling is to use processor sharing models. These models take a macroscopic view of TCP sources neglecting their inherent features including the feedback-based congestion avoidance and control algorithms, TCP timeouts, etc. They assume that the link of interest is fairly divided between a number of TCP connections and each connection is able to fully utilize its fair share of resources. Note that these models do not require the packet loss probability and RTT as a part of the model’s input. In addition, performance of more than a single connection can be studied.

Models belonging to the last group assume no apriori knowledge of loss and delay characteristics. Given a certain amount of the bottleneck’s buffer space, the outgoing link rate share and the number of active TCP connections these models estimate loss and delay characteristics of the network inducing a certain rate of TCP sources. The solution according to such models is called fixed-point approximation (FPA). At the glance FPA is a combination of the TCP source model providing TCP throughput as a function of delay and loss parameters and the queuing system describing delay and loss performance provided by the network. Such combination allows to determine the so-called stationary regime (fixed-point) of a network describing the rate at which TCP sources transmit and performance parameters provided to them by the network. This stationary point is the bottleneck (network) operating regime. From the modeling point of view, FPA brings a number of independent equations that bind together the congestion window dynamics and the network performance parameters, i.e.

\[
\begin{align*}
E[W] &= f(p, T_R) \\
(p, T_R) &= g(E[W])
\end{align*}
\]  

(1)

where \(E[W]\) is the mean size of the congestion window, \(p\) is the packet loss probability, \(T_R\) is the mean RTT, \(f(\cdot)\) and \(g(\cdot)\) denote some functional relationships.
In (1), the first equation provides the average TCP window size as a function of the constant packet loss probability and the mean RTT. Examples of these models include the Mathis and PFTK models. The second equation describes the relationship between $p$, $T_R$, and the average window size $E[W]$. For example, this relationship can be obtained by solving an appropriate queuing system. The mean value of RTT includes the mean queuing delay at the bottleneck, $E[Q]$. The stationary fixed-point is found by solving these equations.

The FPA approach was introduced in [24]. The authors considered throughput of $N$ persistent TCP connections operating in congestion avoidance regime and sharing a single bottleneck with a simplified RED queuing discipline. Firstly, they obtained a simple square-root expression to relate the mean window size of TCP connections to the packet dropping probability and RTT. Then, they noticed that for a fully utilized router the sum of sources’ rates should be equal to the outgoing link rate. Using this reasoning they derived another independent expression for the mean window size of TCP connections sharing a bottleneck. The obtained system of equations was then solved using the Newton-Rhapson gradient technique. Similar setup has been considered in [12]. In that paper, the authors considered the so-called gentle variant of RED [30] and used a more comprehensive PFTK model to estimate TCP throughput [29]. To obtain independent expression for the throughput of a single TCP connection, they inverted the throughput expression for $N$ flows sharing a single bottleneck with a certain buffer size. The system is then solved for a fixed-point providing individual rates of TCP connections. Note that due to the usage of the more accurate expression for TCP throughput, the approach used in [12] provides better results compared to those of [24]. However, application of this approach is rather limited as PFTK model is mainly suitable for a droptail queue management discipline. In [8] and [9], the authors considered performance of TCP Tahoe and TCP Reno connections sharing a single bottleneck with a droptail queuing management discipline. In those studies the congestion avoidance phase of TCP sources has been modeled using a Markov chain model while the bottleneck link was represented using a $M/M/1/K$ queue in [9] and $M/D/1/K$ queue in [8]. Both network and source models were then allowed to numerically tune each other approaching the fixed-point solution. Markov-based TCP source models have been found to provide extremely accurate results. However, the authors did not provide any insights on the choice of the queuing system used to obtain parameters of the packet loss process. This question has been further studied by Wierman et al. in [34], where the authors considered the effect of the network model on the accuracy of TCP throughput approximation. They tested $M/M/1/K$, $M/D/1/K$, and $M^{[X]}/M/1/K$ queuing systems.
and found that the choice of the network model depends on the nature of TCP connections. For short-lived TCP connections, M/D/1/K model was found to be appropriate. When persistent TCP sources are considered, the batch arrival model seems to be more suitable. It is important to note that the authors did not provide any recommendations on the choice of the batch size. Also they did not test appropriateness of the service time distribution in those queuing systems. The model for short-lived TCP sources sharing a bottleneck link was proposed in [7]. In that work, sources have been represented using the TCP models proposed in [29] and [22]. While the PFTK model was found to provide accurate results, the Mathis one was shown to perform unacceptably bad even for low packet loss probabilities.

The fixed-point approach can also be extended to multiple bottleneck scenarios as was firstly demonstrated in [7]. We note that the complexity of the approach grows as we proceed from a single bottleneck to the more complex network scenario. Following [7], single bottleneck models have been extended to multiple bottleneck scenarios in [15, 31]. Particularly, in [15] an arbitrary network topology with two traffic classes has been analyzed. The network model was represented using a M/M/1/K queuing system while the source models included both Mathis and PFTK. The authors, however, did not explain the solution of their fixed-point equations. A little bit simpler approach has been adopted in [31], where Roughan et al. used the buffer overflow model to represent the network part. The fixed-point solution for a network scenario is then given in the closed form.

The fixed-point approximation provides an excellent framework to jointly study source/network interactions. It is versatile enough to include a wide variety of possible TCP source models including Markovian, fluid-flow, and renewal theory models. These models can be coupled with quite general models of the bottleneck link. Fixed-point models can be solved either analytically or numerically. In order for analytical solution to exist, both TCP and queuing models must have closed form expressions for parameters of interest. FPA can be used even in those cases when only few connections share the bottleneck and the congestion window size of these connections affect the packet loss probability.

To the best of our knowledge the work of Le et al. [20] was the only attempt so far to apply the fixed-point approach to wireless channels. To capture properties of wireless channels implementing adaptive modulation and coding (AMC), the authors used the Markov model originally proposed in [21]. Then, they introduced the queuing model that implicitly captures ARQ functionality at the data-link layer. The maximum number of retransmission attempts allowed for a single frame at the data-link layer was limited,
i.e. non-persistent ARQ was assumed. The second independent equation is obtained applying the PFTK model. Although the proposed framework is quite general, it does not incorporate important features of wireless channels. First of all, packet losses were assumed to occur as a result of imperfect wireless channel conditions only implying that the amount of the buffering space at the IP layer is virtually unlimited. Secondly, only one TCP connection was assumed to use resources of the wireless channel. This assumption is inappropriate for modern wireless access technologies where a number of TCP connections may compete for the link’s resources. Furthermore, packet losses at the wireless channel with AMC and non-persistent ARQ is likely to be random rather than correlated meaning that PFTK model does not provide an accurate approximation. Finally, the effective channel rate was assumed to vary according to homogenous Markov chain modeling the AMC. Since the choice of a particular modulation and coding scheme at any given instant of time depends on many factors including the distance between the transmitter and the receiver, direction and velocity of the user, properties of the landscape, it is not possible to know in advance which scheme will be invoked and for how long. This observation questions appropriateness of the Markov model to study performance of AMC-enabled wireless channels. We also note that the modulation scheme and coding rate are not the only parameters that can be changed in attempt to optimize performance metrics provided by wireless channels. Our feeling is that a real-time performance control system should be used instead to decide on protocol parameters providing the best possible performance to applications at any given instant of time (see [26] for more detailed discussion). These parameters should include the size of PDUs at different layers, rate of the FEC code, maximum number of retransmission attempts allowed for ARQ, modulation scheme, traffic rate, etc.

3 The system model

The system of interest is shown in Fig. 1. We consider performance of $N$ TCP connections between a server host (sender) in a wired domain and a client host (receiver) connected via a wireless last-hop link. Our system best corresponds (but not limited to) to the downlink of a cellular broadband wireless access system, where a number of sources competes for resources of the constant bit rate (CBR) wireless channel. The wireless channel is considered to be a bottleneck implying that losses may occur as a result of imperfect error correction at the data-link layer and insufficient buffer space at the IP layer. This is not restrictive assumption considering that rates of
wireless technologies are still below than those of wired links. Since such scenario is common in today’s networks, we do not resort to a particular wireless technology. Instead, we assume a number of features a wireless channel incorporates and capture their effect.

Throughout this paper the maximum segment size (MSS) of TCP is assumed to be 1460 bytes. Coupled with 40 bytes of TCP and IP headers this value corresponds to the maximum transmission unit (MTU) of the Ethernet protocol. Each MSS is packetized into a single IP packet. That is why we further refer to the PDUs at the transport layer as packets. We consider so-called greedy applications that always have data to send. In other words, the sender always sends full-sized TCP segments (i.e. containing MSS bits of data) whenever congestion windows of TCP sources allow. We assume that the receivers’ buffers are sufficiently large, so that TCP sending rates are not limited by the receive windows. We also assume that the time needed to send a window of segments is smaller than the RTT. These assumptions do hold in modern networks. We do not consider competing UDP traffic as its effect is straightforward [13]. We also assume that data are transmitted in one direction only, from the servers to the client hosts.

Data packets arriving at the intermediate system are buffered at the IP layer and dispatched one after another to the data-link layer. Between the IP and data-link layers packets are segmented to a number of frames or code-words. In this paper, we assume that the terms ”frame” and ”codeword” refer to the same entity and each frame consists of exactly one codeword. Extension to multiple codewords within a single frame is straightforward. Both IP packets sizes and frames sizes are assumed to be constant. Then, a FEC code of Reed-Solomon (RS) type is applied and frames start to be transmitted. We consider two various HARQ protocols including Type I HARQ, Type II HARQ with incremental redundancy (Type II IR-HARQ). We concentrate on non-persistent versions limiting the number of retransmission attempts allowed for each frame in a packet. When a certain packet is successfully transmitted or dropped due to an excessive amount of retransmission attempts made for a certain frame the channel is made free for another IP
packet that is queued at the IP layer. The wireless bottleneck link has a service rate of $\mu$ bits per second and the number of waiting positions in the buffer is limited to $K$ full-sized packets. The queue management at the IP layer is assumed to be droptail, so any packet arriving when the buffer is full will be lost. Thus, losses in our system can occur as a result of excessive amount of retransmission attempts at the data-link layer and buffer overflow at the IP layer. We refer to the former as wireless losses and to the latter as congestion losses.

We also take the following assumption about operation of the HARQ systems: (i) feedback frames (negative and positive acknowledgements) are always correctly received, (ii) feedback delay is negligible, (iii) the probability of an undetected error is negligibly small. These assumptions have been used in many studies and found to be suitable for high-speed wireless channels with small propagation delay (see e.g. [4] among others). Observe that under the first two assumptions functionality of stop-and-wait, go-back-n, and selective repeat ARQ implementations become identical.

In order to analyze the system described above, we take the so-called cross-layer approach. We start defining a wireless channel model at the physical layer in terms of the bit error process. Then, we extend it to the transport layer explicitly taking into account the channel adaptation techniques including FEC, non-persistent ARQ schemes, and segmentation and reassembly between different layers. The performance evaluation model at the transport layer is based on the FPA approach, where a queuing model of the bottleneck and TCP model are used to find the stationary regime of the system. Taking into account that packet losses in droptail buffers are strongly correlated while those caused by imperfect error correction are rather independent we distinguish between two different operational regimes of the wireless bottleneck link (i) packets losses are primarily caused by buffers overflow and (ii) packet losses are mainly caused by imperfect error correction. For each case we use different model to estimate TCP throughput.

4 Service process of the wireless channel

Performance of applications in IP networks is evaluated at the IP or higher layers. Thus, conventional wireless channel models such as signal-to-noise ratio (SNR) or bit error models cannot be directly used for performance evaluation purposes and must be extended to the layer at which performance is evaluated. For this extension to be accurate, we have to take into account specific features of underlying layers including data-link error correction techniques, segmentation and reassembly between adjacent layers, etc.
We model the packet service process using the cross-layer approach we previously developed in [28]. According to it, we represent wireless channel characteristics at the physical layer using the bit error process and then extend it probabilistically to the IP layer. The proposed procedure allows to preserve autocorrelational properties of the bit error process and takes into account error correction mechanisms of the data-link layer including both FEC and ARQ. We sketch only basic steps here.

4.1 Bit error process

Let us denote the bit error process by \( \{ W_E(l), l = 0, 1, \ldots \} \), \( W_E(l) \in \{0, 1\} \). We model it using the discrete-time Markov modulated process with irreducible Markov chain \( \{ S_E(l), l = 0, 1, \ldots \} \), \( S_E(l) \in \{0, 1\} \), where 1 and 0 denote incorrect and correct bit reception, respectively. When at most single event is allowed to occur in a slot this process is known as a switched Bernoulli process (SBP). In order to parameterize SBP, only the mean value and lag-1 autocorrelation coefficient have to be provided. Parameters of SBP are estimated as follows [27]

\[
\begin{align*}
\alpha_E &= (1 - K_E(1))E[W_E] \\
\beta_E &= (1 - K_E(1))(1 - E[W_E]) \\
f_{1,E}(1) &= 0 \\
f_{2,E}(1) &= 1
\end{align*}
\]

where \( f_{1,E}(1) \) and \( f_{2,E}(1) \) are probabilities of bit error in states 1 and 2, respectively, \( \alpha_E \) and \( \beta_E \) are transition probabilities from state 1 to state 2 and from state 2 to state 1, respectively, \( K_E(1) \) is the lag-1 autocorrelation of bit error observations, \( E[W_E] \) is the mean of bit error observations.

In what follows we will use matrix notation for \( \{ W_E(l), l = 0, 1, \ldots \} \). In matrix notation SBP is defined using two matrices \( D_E(0) \) and \( D_E(1) \) containing transition probabilities between of \( \{ S_E(l), l = 0, 1, \ldots \} \) with correct of incorrect bit reception. Note that \( D_E(0) + D_E(1) = D_E \), where \( D_E \) is the transition probability matrix of \( \{ S_E(l), l = 0, 1, \ldots \} \).

The proposed model captures first- and second-order statistical characteristics in terms of the bit error rate (BER) and normalized autocorrelation function (NACF). Note that the extension to the case of general finite-state Markov chain (FSMC, [36]) is straightforward. If wireless channels conditions exhibit piecewise stationary behavior as was reported in a number of recent studies, this model may represent statistical characteristics of covariance stationary parts with geometrically decaying autocorrelations. In this case, (2) is interpreted as a model for limited duration of time during which the mean value and NACF of bit error observations remain constant. See [19, 25] for more discussion on non-stationarity of wireless channel statistics.
4.2 Symbol error process

For an RS decoder, a symbol is considered to be lost if at least one bit in a symbol is received incorrectly. As a result, we firstly need to characterize the process of correct and incorrect reception of RS symbols.

Consider the process \( \{ W_N(n), n = 0, 1, \ldots \} \), \( W_N(n) \in \{ 0, 1, \ldots, m_S \} \), describing the number of incorrectly received bits in consecutive bit patterns of length \( m_S \). Here, index of the process denotes successive time intervals of length \( m_S \Delta \), where \( \Delta \) is the time required to transmit a single bit. This process is doubly stochastic, modulated by the underlying Markov chain \( \{ S_N(n), n = 0, 1, \ldots \} \), \( S_N(n) = S_E(l) \in \{ 0, 1 \} \), and can be completely parameterized via parameters of the bit error process. To do so, we have to determine \( m_S \)-step transition probabilities of the modulating Markov chain \( \{ S_E(l), l = 0, 1, \ldots \} \) with exactly \( k \), \( k = 0, 1, \ldots, m_S \), incorrectly received bits. Let us denote the probability of transition from state \( i \) to state \( j \) for the Markov chain \( \{ S_N(n), n = 0, 1, \ldots \} \) with exactly \( k \), \( k = 0, 1, \ldots, m_S \), incorrectly received bits in a bit pattern of length \( m_S \) by \( d_{Nij}(k, m_S) \). Let the set of matrices \( D_N(k, m_S), k = 0, 1, \ldots, m_S \), contain these transition probabilities. Since at most \( i \) bit errors may occur in \( i \) consecutive bit transmissions, we have the following recursion for \( D_N(i, k) \), \( i = 0, 1, \ldots, k \)

\[
D_N(i, 2) = \sum_{k=0}^{i} D_E(k)D_E(i-k), \quad i = 0, 1, 2, \\
D_N(i, 3) = \sum_{k=0}^{i} D_N(k, 2)D_E(i-k), \quad i = 0, 1, \ldots, 3, \quad (3)
\]

\ldots

where \( D_E(k), k \geq 2, \) and \( D_N(i, m_S), i > m_S, \) are all zero matrices.

Consider now the symbol error process \( \{ W_S(n), n = 0, 1, \ldots \} \), \( W_S(n) \in \{ 0, 1 \} \), where '0' indicates the correct reception of a symbol, '1' denotes the incorrect symbol reception. This process is modulated by the underlying Markov chain \( \{ S_S(n), n = 0, 1, \ldots \} \), \( S_S(n) = S_N(n) \in \{ 0, 1 \} \). Let us denote the transition probability from state \( i \) to state \( j \) for the Markov chain \( \{ S_S(n), n = 0, 1, \ldots \} \) with correct \( (k = 0) \) and incorrect \( (k = 1) \) symbol reception by \( d_{Sij}(k), k = 0, 1 \). These probabilities are then combined in matrices \( D_S(0) \) and \( D_S(1) \) describing the symbol error process. The process describing the number of bit errors in consecutive symbols is related to the symbol error process as

\[
D_S(0) = D_N(0, m_S), \quad D_S(1) = \sum_{k=1}^{m_S} D_N(k, m_S). \quad (4)
\]
4.3 Frame error process

Recall that the length of a frame is $m_F$ symbols including those used for error correction. The frame error process $\{W_F(t), t = 0, 1, \ldots\}$, $W_F(t) \in \{0, 1\}$, can be obtained similarly to the symbol error process taking into account that up to $l$ erroneously received symbols can be corrected by the FEC code. Note that index of the process refers to the consecutive time intervals of length $m_F m_S \Delta$. Particularly, the process describing the number of symbol errors in consecutive frames is related to the frame error process as follows

$$D_F(0) = \sum_{k=0}^{l-1} D_M(k, m_F), \quad D_F(1) = \sum_{k=l}^{m_F} D_M(k, m_F), \quad (5)$$

where $D_M(k, m_F), k = 0, 1, \ldots, m_F$, are transition probability matrices of the process $\{W_M(t), t = 0, 1, \ldots\}$, $W_M(t) \in \{0, 1, \ldots, m_F\}$, describing the number of incorrectly received symbols in symbol patterns of length $m_F$, $D_F(k), k = 0, 1$, are matrices describing $\{W_F(t), t = 0, 1, \ldots\}$. Matrices $D_M(k, m_F)$ can be derived using $D_S(k)$ similarly to (3).

Expressions (5) are interpreted as follows: if the number of incorrectly received symbols in a frame is greater than a certain value of the frame error threshold ($k > l$), the frame is incorrectly received and $W_F(t) = 1$. Otherwise ($k \leq l$), it is correctly received and $W_F(t) = 0$.

We also note that the proposed extension limits memory of the bit error process at lag $m_F m_S v$, where $m_F m_S v$ is the length of a single frame in bits. In practice, this value is sufficiently large allowing to accurately capture memory of the bit error process. For example, if the NACF of the bit error process decays according to a single geometrical term, the threshold $m$ at which the model becomes valid can be computed using a theoretical NACF of the bit error process. Recall, that it can be expressed as $K_E(m) = \lambda^m$, where $\lambda$ is lag-1 NACF. Observing Fig. 2(a) one may notice that even for highly correlated bit error processes with $\lambda = 0.9$ the correlation becomes negligible for lags greater than 30 as $0.9^{30} \approx 0.042$.

To confirm our conclusion, we tested the residual autocorrelation properties of the frame error process. In our experiments, the frame size was set to 255 bits, the mean of the bit error process was kept constant, while lag-1 NACF values varied from 0.1 to 0.9. The obtained NACFs are shown in Fig. 2(b). As one may observe, all NACFs are within their confidence limits ($\pm 2/\sqrt{n}$, where $n$ is the number of observations used to computed NACFs) meaning that the frame error process is uncorrelated irrespective of the degree of autocorrelation in the bit error processes. This implies that the memory of the covariance stationary bit error process is destroyed by the
first segmentation procedure (between the physical and the data-link layers). This observation allows us to conclude that the frame error process can be described by the frame error probability, \( f(1) \), which can be obtained using \( D_F(1) \) as

\[
f(1) = \bar{\pi}_F D_F(1) e,
\]

where \( f(0) = 1 - f(1) \) is the probability of correct frame reception, \( \bar{\pi}_F \) is the stationary distribution of \( \{W_F(t), t = 0, 1, \ldots\} \), \( e \) is the vector of ones of an appropriate size.

The abovementioned property allows to get rid of matrix computations in what follows and use scalars \( f(i), i = 0, 1, \) instead. We note that one can provide a number of counter examples showing that memory of the bit error process can be significantly longer than that of the stationary Markov process. However, as discussed in [19, 25], processes with limited memory can be considered as candidate models for a limited duration of time during which the mean value and NACF of bit error observations remain constant.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{memory.png}
\caption{The memory of the bit and frame error process.}
\end{figure}

4.4 IP packet service process

4.4.1 Type I HARQ system

Let us now concentrate on a Type I HARQ system limiting the number of retransmissions for each frame in a packet. Let the random variable (RV) \( D, D \in \{\min(r,v), \min(r,v) + 1, \ldots, rv\} \) denote the packet transmission time
and let \( d(k) \) be its probability function (PF). Note that according to this scheme whenever \( r \) times a frame is failed to be correctly transmitted, the whole IP packet is dropped irrespective of the number of frames that have already been correctly transmitted. Observe that the minimum transmission time of a packet is \( \min(r, v) \) slots, while the maximum time is \( rv \). The former happens when all frames are successfully transmitted in their first transmission attempts. The latter occurs when the first \( (v - 1) \) frames required exactly \( r \) attempts to be transmitted correctly while the last frame is either correctly or incorrectly received in \( r \) attempts.

Let \( A \) be the event of successful packet transmission and \( I_P \) be its indicator, i.e. \( I_P = 1 \) when a packet is successfully transmitted and \( I_P = 0 \) otherwise. PF of the packet transmission time can be obtained applying the law of total probability as follows:

\[
d(k) = c(k) + b(k), \quad k = \min(r, v), \min(r, v) + 1, \ldots, rv,
\]

where \( c(k) = Pr\{D = k, I_P = 1\} \) is probability of delay on \( k \) slots induced by successful packet transmission, \( b(k) = Pr\{D = k, I_P = 0\} \) is the probability of delay on \( k \) slots induced by unsuccessful packet transmission.

We obtain \( d(k) \) according to (7). However, we firstly need to describe the packet loss process at the data-link layer. Let \( f_L \) be the probability that a single frame is lost as a result of an excessive number of retransmission attempts. We find \( f_L \) using the frame error probability \( f(1) \) as follows

\[
f_L = 1 - \sum_{i=1}^{r} [f(1)]^{i-1} f(0).
\]

Let further \( f_{L,i}, \ i = 1, 2, \ldots, v, \) be the probability that the whole IP packet is dropped as a result of an excessive number of retransmissions made for the \( i \)th frame. Since successive frame transmission times are independent, \( f_{L,i}, \ i = 1, 2, \ldots, v, \) can be found using \( f_L \) as follows:

\[
f_{L,i} = (1 - f_L)^{i-1} f_L.
\]

Using (8) we get the IP packet loss probability due to an excessive number of retransmission attempts made for a single frame as

\[
p_L = Pr\{I_P = 0\} = 1 - (1 - f_L)^v.
\]

Let \( U_1, U_i \in \{1, 2, \ldots, r\} \) be the RV denoting the number of transmission attempts required to transmit a single frame given that is successfully transmitted and let \( u(1, k) \) be its PF, i.e. \( u(1, k) = Pr\{U = i|I_P = 1\} \), where \( I_P \)}
is the indicator of successful frame transmission. Since probabilities of incorrect frame reception in successive slots are independent from each other, the amount of time to successfully transmit a single frame is given by

\[ u(1, k) = \frac{[f(1)]^{k-1}f(0)}{\sum_{r=1}^{r}[f(1)]^{r-1}f(0)}, \quad k = 1, 2, \ldots, r. \]  

(11)

In what follows, we determine delays caused by both successful and unsuccessful packet transmission recursively assuming that a packet consists of \(i, i = 1, 2, \ldots, v\), frames. The delay for a given number of frames in a packet is then obtained by setting \(i = v\). Let us extend the definition of the RV \(U_1\) as follows. Let \(U_i, i = 1, 2, \ldots, v, U_i \in \{\min(r, i), \min(r, i) + 1, \ldots, ri\}\), be the set of RVs describing delay of a single packet given that it consists of exactly \(i\) frames and all of them have been successfully transmitted and let \(u(i, k)\) be their PFs. The first PF in the sequence, \(u(1, k)\), has already been found in (11). We get \(u(2, k)\) using convolution of two PFs \(u(1, k)\) as follows:

\[ u(2, k) = \sum_{j=0}^{k} u(1,k)u(1,k-j), \quad k = \min(r, 2), \min(r, 2) + 1, \ldots. \]  

(12)

Extending (12) to the case \(i = 3, 4, \ldots, v\), we get:

\[ u(i, k) = \sum_{j=0}^{k} u(i-1,k)u(1,k-j), \quad k = \min(r, i), \ldots, \]  

(13)

where \(u(i, k) = 0, \; i = 1, 2, \ldots, v, \; k > i(r - 1)\).

To compute delay induced by successful packet transmission, we need \(u(v, k)\) \(k = \min(r, v), \min(r, v) + 1, \ldots, rv\), in (13). Observe that this PF describes the delay of the packet conditioned on the event of successful packet transmission, i.e. \(Pr\{D = k|I_P = 1\}\). Removing conditioning, we get:

\[ c(k) = u(v,k)(1-p_L), \quad k = \min(r, v), \min(r, v) + 1, \ldots, vr. \]  

(14)

Let now \(U^*_i, i = 1, 2, \ldots, v, U^*_i \in \{r, r + 1, \ldots, ri\}\) be the set of RV describing delay induced by unsuccessful packet transmission given that the \(i\)th frame failed to be received correctly as a result of an excessive number of retransmission attempts and let \(u^*(i, k)\) be their PFs. The easiest way to get these PFs is to use \(u(i, k)\), \(i = 1, 2, \ldots, v - 1, k = \min(r, i), \min(r, i) + 1, \ldots, ri\), already obtained in (13). For \(i = 1\) we have all weight located at \(r\). Consider that the \(i\)th frame, \(i = 2, 3, \ldots, v\) failed to be transmitted. It implies that all preceding \((i - 1)\) frames have been transmitted correctly and
the transmission time induced by them is given by \( u(i - 1, k) \). The only possible number of slots induced by the failed attempt to transmit \( i \)th frame is \( r \). Thus, we have:

\[
    u^*(i, k) = u(i - 1, k - r), \quad k = r + i, r + i + 1, \ldots, ir. \tag{15}
\]

Observe that these PFs are conditioned on the event of unsuccessful packet transmission, i.e. \( \text{Pr}\{D = k | I_P = 0\} \). Thus, in order to obtain \( b(k) \), \( k = r, r + 1, \ldots, rv \), we have to remove conditioning on unsuccessful packet transmission as a result of the failed attempt to transmit \( i \)th frame and sum all those functions contributing to a particular value of the packet transmission time. We have

\[
    b(k) = \sum_{i=1}^{v} u^*(i, k)(1 - f_{L})^{i-1}f_{L}, \quad k = r, r + 1, \ldots, rv. \tag{16}
\]

Substituting (14) and (16) into (7), we get the final result.

### 4.4.2 Type II HARQ system

Let us now consider a Type II HARQ system with incremental redundancy (IR-HARQ). According to IR-HARQ, in the first transmission attempt only data bits are transmitted. Retransmissions carry redundancy bits only which are added to the original codeword at the receiver increasing the probability of successful frame reception. Although properties of some coding techniques used in Type II HARQ systems (e.g. convolutional codes) are able to generate infinitely long error correction sequence and potentially allow for persistent operation, most Type II IR-HARQ schemes limit the number of retransmissions allowed for a single frame to some value \( r \).

Assume that the cumulative FEC-encoded frame received at the \( i \)th retransmission attempt is given by \( (n, in) \), \( i = 1, 2, \ldots, r \). This implies that exactly \( n \) symbols are sent at each retransmission attempt. Let \( D, D \in \{\min(r, v), \min(r, v) + 1, \ldots, rv\} \), denote the packet transmission time and let \( d(k) \) be its PF. Observe that similarly to the Type I HARQ the minimum delay is \( \min(r, v) \) slots, while the maximum delay is \( rv \). The delay and loss performance of Type II IR-HARQ system can be also be analyzed similarly to the non-persistent Type I HARQ system. The only required modification is to recompute the frame and packet loss probabilities.

Let \( f_{L,i}, i = 1, 2, \ldots, r \), be the probability of incorrect frame reception after \( i \)th retransmission attempt. Note that \( f_{L,i} \geq f_{L,j} \) for \( i > j \). These probabilities can be found computing the cumulative number of incorrectly
received symbols in patterns of $i$ consecutive symbol transmissions as explained previously. Then, applying (5) one can get $f_{L,i}$, $i = 1, 2, \ldots, r$, using the cumulative error correction threshold, $l$. When the Reed-Solomon codes are used, $l$ at the $i$th retransmission attempt is $(in - n)/2$, $i = 1, 2, \ldots, r$.

Once $f_{L,i}$, $i = 1, 2, \ldots, r$, are obtained the frame loss probability is readily given by $f_L = f_{L,r}$. Thus, the packet loss probability is

$$p_L = 1 - (1 - f_{L,r})^v. \quad (17)$$

Finally, observing that the probability of IP packet loss caused by an excessive amount of retransmission attempts made for the $i$th frame is geometrically distributed with parameter $f_{L,r}$ the packet transmission time can be found using conditional PFs as explained previously for the non-persistent Type I HARQ system.

5 Fixed-point approximation

5.1 TCP models

In Section 4, we have shown that for covariance stationary Markovian bit error statistics the packet loss process caused by imperfect error correction tends to be purely stochastic (uncorrelated) with geometrically distributed interloss times. Given this loss process, the throughput of a TCP connection can be approximated by the well-known Mathis formula:

$$\gamma_m(p, T_R) = \frac{\sqrt{3L}}{2T_R \sqrt{p}}, \quad (18)$$

where $L$ is the packet size, $T_R$ is the mean RTT value, $p$ is the packet loss probability.

This result was published in [22] and since then it is known as Mathis formula. It is based on geometrical interpretation of the sawtooth behavior of TCP in the congestion avoidance phase and was shown to provide a fair approximation of TCP throughput in a completely random loss environment. It is important to note that in (18) the effect of timeouts is not taken into account. The reason is that when successive packet losses are independent and rare the timeout probability is negligibly small. We use (18) when losses caused by imperfect error control is the major source of performance degradation.

Another reason for IP packet losses in a wireless environment is buffer overflows. To model persistent TCP connections whose packets can be
dropped as a result of buffer overflow at the wireless bottleneck we use PFTK model providing TCP throughput in the following form [29]:

\[
\gamma_p(p, T_R) = \frac{L \left( \frac{1-p}{p} + \frac{W(p)}{2} + Q(p, W(p)) \right)}{T_R(W(p) + 1) + \frac{Q(p, W(p))G(p)T_0}{1-p}}, \quad W(p) < W_M,
\]

\[
\gamma_p(p, T_R) = \frac{L \left( \frac{1-p}{p} + \frac{W_M}{2} + Q(p, W_M) \right)}{T_R \left( \frac{W_M}{4} + \frac{1-p}{pW_M} + 2 \right) + \frac{Q(p, W_M)G(p)T_0}{1-p}}, \quad W(p) \geq W_M,
\]  

(19)

where \(W_M\) is the maximum receive window size, \(T_0\) is the mean value of the TCP retransmission timeout and

\[
Q(p, w) = \min \left( 1, \frac{(1 - (1 - p)^3)(1 + (1 - p)^3(1 - (1 - p)^{w-3}))}{1 - (1 - p)^w} \right),
\]

\[
G(p) = 1 + p + 2p^2 + 4p^3 + 8p^4 + 16p^5 + 32p^6,
\]

\[
W(p) = \frac{2}{3} + \sqrt{\frac{4(1-p)}{3p} + \frac{4}{9}}.
\]  

(20)

This result was derived by Padhye et al. in [29] and is actually generalization of the previous efforts in application of the renewal theory to TCP modeling. Authors in [29] validated their model against real traces and found it to be fairly accurate for a wide range of loss statistics. This model was also extensively used in fixed-point analysis in wired networks and found to be relatively accurate. The reason is that it is quite versatile and includes many specific features of the TCP implementation including the receive window size, loss detection via timer expiration, and reception of three duplicate acknowledgements (DUPACK). The first expression in (19) describes the throughput of TCP connection when the congestion window is less than or equal to \(W_M\). The last term in the nominators of (19) accounts for throughput limitation imposed by the receive window size. The last term in the denominators accounts for losses occurring as a result of timeout expiration. One may observe that this result is in agreement with the \(1/\sqrt{p}\)-law. Indeed, setting \(T_0 = 0\) and \(W_M \to \infty\), we approximate Mathis expression for throughput of a single TCP connection [22].

Note that in (18) and (19) both the packet loss probability and the mean RTT are taken into account. They provide the first relationship in the form of \(\gamma = f(T_R, p)\). These two models are appropriate for different operational regimes of the wireless channel. We discuss why it is possible to clearly differentiate between these two regimes further in Section 6.
5.2 Queuing model

To obtain the other independent expression binding \((p, T_R)\) and the average arrival rate of TCP connections, consider the packet service process at the IP layer. Provided that the network is in its stationary regime the wireless channel bottleneck can be modeled using a queuing system with appropriate arrival and service processes. We use D-BMAP/G/1/K queue, where \(K\) is the capacity of the system measured in packets. We allow service time distribution to be general and follow \(d(k)\) derived in Section 4. The reason to use D-BMAP as the arrival process is that this process is quite versatile and includes many simpler processes as its special cases. This would allow us to study applicability of various queuing systems for the FPA approach.

A time diagram of the D-BMAP/G/1/K queuing system is shown in Fig. 3. According to this system, packets arrive in batches, and batches of packets arrive just before the end of slots. Arrivals are not allowed to seize the server immediately and the service of any arrival starts at the beginning of a slot. Arrivals depart from the system at the slot boundaries, just after batch arrivals (if any). The state of the system is observed just after the departure (if any) and these points are imbedded Markov points. This system is known as “late arrival model with delayed access”. The sojourn time is counted as the number of slots spent by a packet in the system. The system can accommodate at most \(K\) packets. We assume partial batch acceptance strategy. According to this strategy, if a batch of \(R\) packets arrives when \(k\) packets are in the system and \(R > (K - k)\), only \((K - k)\) packets are accommodated.

![Figure 3: Time diagram of the D-BMAP/G/1/K queuing system.](image)

We define D-BMAP \(\{W_A(n), n = 0, 1, \ldots \}\), \(W_A(n) \in \{0, 1, \ldots \}\), using an infinite sequence of matrices \(D_A(k), k = 0, 1, \ldots \), each of which contains transition probabilities of the underlying Markov chain \(\{S_A(n), n = 0, 1, \ldots \}\), \(S_A(n) \in \{0, 1, \ldots , M - 1\}\), of the process with exactly \(k\) packet arrivals.
Complete description of the system requires a two-dimensional Markov chain \( \{ S_Q(n), S_A(n), n = 0, 1, \ldots \} \) imbedded at the moments of packet departures from the system, where \( S_A(n) \) is the state of the packet arrival process, and \( S_Q(n) \in \{ 0, 1, \ldots, K - 1 \} \) is the number of packets in the system just after packet departures. To parameterize this process we have to determine transition probabilities between imbedded Markov points.

Let \( D_A(k, m), k = 0, 1, \ldots, m = 0, 1, \ldots \), be the set of matrices describing transitions from state \( i \) to state \( j \), \( i, j \in \{ 0, 1, \ldots, M - 1 \} \), of the modulating Markov chain of the packet arrival process in \( m \) steps with exactly \( k \) packet arrivals. Setting \( D_A(1, k) = D_A(k), k = 0, 1, \ldots \), we find them recursively as

\[
D_A(k, m) = \sum_{l=0}^{m} D_A(k - 1, l)D_A(m - l).
\]  

(21)

Let \( D_Q(k), k = 0, 1, \ldots \), be the set of matrices describing transitions from state \( i \) to state \( j \), \( i, j \in \{ 0, 1, \ldots, M - 1 \} \), of the modulating Markov chain of the arrival process with exactly \( k \) arrivals in a service time of a single packet. We determine them using the service time distribution \( d(k) \) and (21):

\[
D_Q(k) = \sum_{i=0}^{\infty} D_A(k, i)d(i), \quad k = 0, 1, \ldots
\]  

(22)

Let \( C(i, j), i, j \in \{ 0, 1, \ldots, K - 1 \} \) be the transition probability matrices describing time evolution of \( \{ S_Q(n), S_A(n), n = 0, 1, \ldots \} \). These matrices can be completely defined using \( D_Q(k), k = 0, 1, \ldots \) as follows:

\[
C(0, j) = \begin{cases} 
D_Q(k), & j \neq K, \\
\sum_{m=K}^{\infty} D_Q(m), & j = K,
\end{cases}
\]

\[
C(i, j) = \begin{cases} 
D_Q(j - i + 1), & j \neq K, \quad j > i - 2, \\
\sum_{m=K-i}^{\infty} D_Q(m), & j = K, \quad j > i - 2.
\end{cases}
\]

Let now \( T \) be the transition probability matrix containing \( C(i, j), i, j \in \{ 0, 1, \ldots, M - 1 \} \) as its \((i, j)\) elements and let \( \vec{x}_D = (x_{D,01}, \ldots, x_{D,K-1M}) \) be its steady-state distribution. Solving \( \vec{x}_D T = \vec{x}_D, \vec{x}_D \vec{e} = 1 \), we get \( x_{kj} = \lim_{n \to \infty} Pr\{ S_Q(n) = k, S_A(n) = j \} \). There are a number of algorithms to compute these probabilities, e.g. [14, 23, 1]. To obtain performance metrics of interest, we need steady-state probabilities of the number of packets in the system in the arbitrary slot, \( \vec{x} \), and just before arrival of a batch, \( \vec{x}_A \). These quantities can be found using results reported in [18]. Once \( \vec{x} \) and \( \vec{x}_A \) are obtained, one can use results of [16] to get all the required performance metrics.
5.3 FPA solution

Before we proceed with the solution of our fixed-point equations, we need to make two important observations. First of all, as we observed previously the throughput of a TCP connection is a function of the packet loss probability and RTT. The mean RTT of the connection $i$, $T_{R,i}$, can be written as

$$T_{R,i} = \tau_i + E[Q^*_i]/\mu, \quad (23)$$

where $\mu$ is the raw rate of the wireless channel, $\tau_i$ accounts for the two-way propagation delay in the wired part of the network, $E[Q^*_i]$ is the mean waiting time of packets of connection $i$ at the bottleneck given that these packets are not lost at the data-link layer.

Following [24, 12, 9, 7] we assume that all TCP connections experience the same average queuing delay and the packet loss probability as a result of buffer overflows. Thus we have $p_{B,i} = p_B$, $E[Q^*_i] = E[Q^*]$, $i = 0, 1, \ldots, N - 1$, where $N$ is the number of TCP connections, $E[Q^*]$ is the mean waiting time in the system given that the packet of interest has not been lost at the data-link layer. Note that $E[Q^*]$ can be obtained as a sum of the mean waiting time in the buffer and the time required to successfully transmit a packet over the wireless channel. One can get the latter term using $c(k) = Pr\{D = k, I_P = 1\}$ derived in Section 4. Since the outcome of the packet transmission is independent of the mean waiting time in the buffer, PF of the waiting time in the system given that the packet of interest has not been lost at the data-link layer can be obtained using convolution.

Now RTTs $T_{R,i}$ can be rewritten as follows:

$$T_{R,i} = \tau_i + E[Q^*]/\mu. \quad (24)$$

Secondly, observe that in addition to the packet loss probability for PFTK model we will also need the packet loss indication. Observe that for queuing systems without batch arrivals the packet loss probability and the packet loss indication are the same. The reason is that for these models at most one arrival can be lost in a slot. Once batch arrivals are allowed, these metrics are different. However, the packet loss indication, $p_I$, can be computed using PF of the number of lost packet in a slot, $f_P(i)$, $i = 0, 1, \ldots$ as $p_I = 1 - f_P(0)$, where $f_P(0)$ is the probability that there are no lost packets in the slot.

Let us now discuss the solution of the fixed point equations derived in previous subsections. Consider the queuing system first. For a given set of arrival rates $\lambda_{A,i}$ chosen with step $\Delta \lambda_A$ we look for the average waiting time of packets in the system $E[Q^*]$. Note that each $\lambda_{A,i}$ also induce a certain packet loss probability $p_B$ as a result of buffer overflows. This allows $p_B$ and $E[Q^*]$ to be expressed as a function of $\lambda_A$. 

21
Note that packet loss probability as a result of buffer overflows, \( p_B \), is not the ultimate measure of the loss performance. The packet loss probability as a result of an excessive number of retransmission attempts, \( p_L \), also contributes to the overall loss performance. Also note that \( p_L \) implies that a packet is accepted to the system, i.e. not lost due to buffer overflow. Using these observations we get the overall packet loss probability \( p \) as

\[
p = 1 - (1 - p_L)(1 - p_B).
\]

Now observe that for a fully utilized wireless channel shared by \( N \) TCP sources the following holds

\[
\sum_{i=0}^{N-1} \gamma_i(p, T_{R,i}) = \mu, \tag{26}
\]

where \( \gamma_i(p, T_{R,i}), i = 0, 1, \ldots, N - 1 \) are the throughputs of individual TCP connections computed according to either Mathis of PFTK models, \( \mu \) is the raw rate of the wireless channel. Recall that the choice of the TCP model depends on whether \( p_L > p_B \).

Using (26) we estimate individual rates \( \gamma_i(p, T_{R,i}) \) corresponding to the previously obtained values of \( E[Q^*] \) and \( p \). The intersection point between \( \sum_{i=0}^{N-1} \gamma_i(p, T_{R,i}) = \mu \) and \( \lambda_A = f(E[Q^*], p) \) gives the fixed-point solution in terms of \( \gamma_i(p, T_{R,i}), i = 0, 1, \ldots, N - 1 \). Note that since \( E[Q^*] \) is discrete function with step \( \Delta E[Q^*] \), the fixed-point is obtained by approximating \( \sum_{i=0}^{N-1} \gamma_i(p, T_{R,i}) = \mu \) and \( \lambda_A = f(E[Q^*], p) \) around the intersection point by piecewise linear function. In order to obtain rates of individual connections \( \gamma_i(p, T_{R,i}), i = 0, 1, \ldots, N - 1 \), we need to normalize them with respect to their RTTs, \( T_{R,i} \). Example of the solution is shown in Fig. 4.

6 Numerical results

Here, we demonstrate the effect of various model’s parameters on the throughput of a TCP source. We also study the effect of queuing and TCP models on the accuracy of FPA technique. For Type I HARQ systems we assume an RS code of size 20 symbols containing exactly 10 data symbols. For Type II HARQ the size of each transmission is exactly 10 symbols. Assuming packet size of 1500 bytes, the minimum packet transmission time at the data-link layer is 150 time slots. The mean RTT of the wired network is set to 0.06s and the capacity of the queuing system is chosen to be 30 packets. Finally, the raw rate of the wireless channel is assumed to be 384Kbps. We would
like to note that those insights of the system behavior we discuss further remain qualitatively similar for other choices of FEC codes, RTTs in the wired network, and raw rates of the wireless channel.

### 6.1 Preliminary notes

Solution of the D-BMAP/G/1/K system involved in the FPA approach requires significant computational efforts as no closed form expressions for the packet loss probability and the mean packet delay exist for this system. As a result, to get each TCP throughout corresponding to a certain set of input parameters multiples Markov chains associated with D-BMAP/G/1/K queuing systems need to be solved numerically. In what follows, we will try to understand whether it is possible to decrease these efforts replacing the D-BMAP/G/1/K queuing system by simpler models.

In addition to the D-BMAP/G/1/K queuing model we also consider here Geo/Geo/1/K, Geo/G/1/K, Geo[^X]/Geo/1/K, Geo[^X]/G/1/K, SBP/Geo/1/K, and SBP/G/1/K. Here, we do not provide analysis of these systems as it can be found elsewhere, e.g. in [6, 32]. Before we proceed with numerical results few additional notes need to be made. Observe that the Geo/Geo/1/K system is the simplest case of finite-buffer discrete-time queuing systems. It is also the only one having closed form expression for both mean packet delay and packet loss probability [6]. For example, there is no closed-from expression for the packet loss probability for the Geo/G/1/K queuing system. However, observe that numerical analysis is still much simpler compared to
the D-BMAP/G/1/K queuing system, especially, when the number of states of the modulating Markov chain of D-BMAP is more than a few digits. In order to parameterize $\text{Geo}^{[X]}/\text{Geo}/1/K$ and $\text{Geo}^{[X]}/G/1/K$ we need to provide the mean batch size, $b$, and the average batch arrival intensity $\lambda_{A,b}$. Here, we assume that the batch size is constant. Once $b$ and the average arrival intensity of packets, $\lambda_A$, are known the average batch intensity can be found as $\lambda_{A,b} = \lambda_A/b$. Observe that $\lambda_A$ is included as a part of the FPA approach.

We also need to specify the SBP arrival process. When D-BMAP $\{W_A(n), n = 0, 1, \ldots \}$ is allowed to have two states only, each state of D-BMAP is associated with a Bernoulli distributed number of arrivals in a single slot, and these distributions depend on the current state only it reduces to the SBP. Let $\{G_A(n), n = 0, 1, \ldots \}$ be the rate process of SBP $\{W_A(n), n = 0, 1, \ldots \}$ (see [5] for more details). Note that ACFs of SBP and of its rate processes are generally different due to the difference in $R_W(0)$ and $R_G(0)$, where $R_W(0)$ and $R_G(0)$ are variances of the SBP and of its rate process, respectively. However, there is a special case of a SBP that is of particular interest. This process is obtained by setting the probability of arrival in state 1 to 1 and in state 2 to 0. Due to the specific structure of this process it is characterized by the following properties:

$$E[W] = E[G], \quad R_G(i) = R_W(i) = \theta^{i-1}, \quad i = 0, 1, \ldots, \quad (27)$$

where $R_W(i)$ and $R_G(i)$ are NACFs of the SBP and of its rate process, respectively, $\theta$ is the non-unit eigenvalue of the transition probability matrix of the modulating Markov chain. The former property holds for any D-BMAP, the latter – for this process only.

Although (27) reduces generality of D-BMAP, the usage of SBP in illustrative studies is much easier compared to the general D-BMAP. We already used it in this paper to model the bit error process in Section 4. Recall, that in order to parameterize SBP we need to provide the mean and the lag-1 NACF values only. Thus, changing the lag-1 NACF value for each value of the mean arrival rate we may get packet arrival processes with different NACFs. Note that this process can also be viewed as a batch arrival one. For large values of lag-1 NACF arriving packets tend to organize in so-called trains.

### 6.2 Choosing queuing and TCP models

To choose an appropriate queuing system and a TCP model to use in FPA analysis we need to identify important properties of the packet arrival and loss processes in the droptail router. TCP uses feedback from the network...
to decide upon traffic injection to the network. As a result, the packet arrival process is not random but somewhat deterministic. For example, assuming loss-free environment, behavior of the TCP sender is completely deterministic and dictated by receiver’s window and the network capacity. Losses occurring randomly introduce stochastic components into TCP traffic behavior. Assuming that a single TCP source is in the congestion avoidance state the window evolution is described by the following system of equations:

\[
\begin{align*}
W_{i+1} &= W_i + 1, & I_L &= 0 \\
W_{i+1} &= W_i / 2, & I_L &= 1
\end{align*}
\]

where \( I_L \) is the indicator of a packet loss event.

Observing (28) one may notice that the number of packets that are allowed to be sent depends on whether there was a loss in the previous round. At the same time, the packet packet loss probability depends on the amount of packets generated by the source. Thus, the packet loss process cannot be considered independently of the packet arrival process.

6.2.1 Packet loss process

The choice of the TCP model depends on the packet loss process experienced by individual TCP sources. According to our system there are two packet loss processes affecting TCP throughput. These are the loss process caused by imperfect local error correction and the loss process resulting from the buffer overflow. Thus, the aggregated loss process is a superposition of the alternative loss/no-loss cycles and a renewal Bernoulli process. Observe that parameters of the packet loss process caused by imperfect local error correction are independent of the packet arrival and buffering processes. However, those losses occurring as a result of buffer overflows do depend on the state of TCP connections and the size of the buffer.

The question we need to answer is can we clearly differentiate between the two operational regimes of a wireless channel? Observe that the modeling framework introduced in Sections 4 and 5 allows to estimate performance of applications that do not use network/source interactions, e.g. real-time services using RTP/UDP at the transport layer. Fig. 5 illustrates two loss processes associated with the system for two different offered traffic loads \( \rho \). These figures were made assuming a Type I HARQ system with \( r = 9 \), Poisson arrival process, the packet size of 1500 bytes, and \( K_E(1) = 0.0 \).

The buffer size was limited to \( K = 30 \) packets. Observing Fig. 5 one may notice that the packet loss probability caused by buffer overflow dominates the packet loss process when BER values are small. When BER increases
packets start to be lost at the data-link layer. However, losses caused by buffer overflow do not diminish immediately. The reason is that the mean packet service time is still high. Recall, that the packet service time includes time to transmit packets that are lost at the data-link layer. When BER increases even further losses caused by imperfect error correction become abundant and packet losses caused by buffer overflow decrease. The reason is that the mean packet service time decreases when most packets are lost as a result of inability to transmit first few frames in a packet. Note that the described dependencies are also inherent for Type II HARQ systems with incremental redundancy. As a result, for both HARQ systems we can use Mathis model when losses are primarily caused by wireless channel conditions and PFTK model when losses caused by buffer overflow dominate.

6.2.2 Packet arrival process

Another issue we need to address is which arrival process is best suited for our FPA model. In the following discussion, we distinguish between two cases: (i) there is only one TCP source using the wireless channel and (ii) there are multiple TCP sources. As we will see in what follows, in order to decide upon traffic characteristics we need to take into account the interplay between values of $E[D]$ and $\tau_i$, $i = 0, 1, \ldots, N - 1$. Recall that the packet transmission time is the same for all TCP connections while $\tau_i$ are allowed to be different. To get more insights into traffic injection process we firstly consider how packets are generated by a single TCP source.

![Figure 5: Individual and aggregated loss processes.](image-url)
When short-distance wired network and/or low-speed wireless technology are considered $\tau < E[D]$. In this case packet arrivals are more or less evenly spread in time as shown in Fig. 6(a), where IS stands for intermediate system. Since the time required to transmit a single packet is a RV with mean $E[D]$, the packet arrival process is mainly affected by the packet service process of the wireless channel. The distance between packet arrivals is random with mean $E[D]$. Thus, the system of interest is $G/G/1/K$. When variance of the packet transmission time is small fairly accurate approximation can be obtained applying the $D/G/1/K$ queuing system. When long-distance wired network and/or high-speed wireless technology are of interest $\tau > E[D]$. In this case the packet arrival process is bursty in nature as shown in Fig. 6(b). More specifically, interarrival times between batches of packets depend of the value of $\tau$, while packets in a batch are separated by $E[D]$. The system of interest is then $G^{[X]}/G/1/K$. Similarly to the previous case when variance of $\tau$ is small accurate approximation is obtained using deterministic batch arrival process and the system of interest is $D^{[X]}/G/1/K$.

Consider now the case of multiple TCP connections. When $\tau_i > E[D]$, $i = 0, 1, \ldots, N - 1$, the packet arrival process is bursty in nature as shown
in Fig. 7(b). One can also observe that interarrival times between batches of packets depend on many factors including time instants when TCP connections originate and probabilistic characteristics of $\tau_i$, $i = 0, 1, \ldots, N - 1$. Assuming independent originating times of connections, interarrival times between batches can be approximated by a memoryless geometric distribution. The resulting queuing system is Geo$^{[X]}$/G/1/K. Also note that the amount of time between packets in a batch is random with mean $E[D]$. When $\tau_i < E[D]$, $i = 0, 1, \ldots, N - 1$, and the number of connections is significant starting times of connections dictate behavior of the packet arrival process as shown in Fig. 7(a). Once again, assuming that all connections starts at arbitrary time instants the packet interarrival times can be approximated by a geometric distribution. Thus, the system of interest is Geo/G/1/K.

The situation becomes more complicated when some of $\tau_i$ are less than $E[D]$ while some are greater. In this case the fraction of TCP connections having $\tau_i > E[D]$ adds to the burstiness of the arrival process. Those connections with $\tau_i < E[D]$ operate more or less randomly adding arrivals according to connections arrival times. If the number of connections with $\tau_i > E[D]$ is significant compared to those with $\tau_i < E[D]$ we suggest to use a Geo$^{[X]}$ arrival process. When TCP connections with $\tau_i < E[D]$ dominate we use a Geo arrival process.

Figure 7: Packet arrival processes for different $\tau_i$ and $E[D]$.

Note that packet losses may significantly disrupt the identified arrivals
processes of interest. For example, when multiple TCP sources share a single bottleneck link with a droptail queuing discipline they may become synchronized. The reason is that packets from a number of flows can be dropped simultaneously eventually leading to synchronization among TCP connections. In this case packets tend to arrive in batches. While synchronization effect was shown to occur frequently in wired networks with droptail routers, it may not happen in wireless environment. Indeed, when packet losses occurring as a result of imperfect error correction is at least of the same order of magnitude as those caused by buffer overflow and they are not severely correlated synchronization between TCP connections is less likely to occur. Essentially, these losses emulate behavior of untuned RED queue management which was shown to effectively cope with synchronization problem. When wireless losses are small compared to those caused by buffer overflow the wireless channel behaves similarly to the wired one.

To support our conclusions we carried out measurements of the packet arrival process from a number of TCP connections running over HSDPA interface. Since we tested the case when \( \tau_i > \mathbb{E}[D] \), \( i = 0, 1, \ldots, N - 1 \), we expect that the packet arrival process is bursty in nature. The number of TCP connections was set to 5 and the starting times of all flows were chosen randomly. These connections were used to download the same file from the same ftp server meaning that \( \tau_i \approx \tau_j \), \( \forall i, j \). The file size was sufficient to remove the effect of the transient slow-start phase of TCP connections. Since we were not able to physically access the base station, we measured the packet sending times at the TCP senders. Observing Fig. 8 one may notice that these times approximately resembles packets arriving times at the intermediate system.

Let us tag a certain connection and let \( j_{k,l}, k = 0, 1, \ldots \) denote the number of packets in-between \((l - 1)\)th and \(l\)th packets of the tagged connection. Let \( J, J \in \{0, 1, \ldots\} \), be the RV denoting the number of packets from other connections in-between two successive packets of the tagged one. PFs of \( J \) for two randomly chosen TCP connections are demonstrated in Fig. 8(a). First of all, notice that the probability of having zero packets in-between two packets of the tagged connection is very high and approaches 0.8. This implies that packets from a single TCP connection tend to arrive in batches. Note that the structure of PFs of \( J \) for all TCP connections sharing a wireless channel is qualitatively similar. This behavior can be attributed to fairness of the protocol when all RTTs are approximately the same. \( \chi^2 \) statistical test for homogeneity of samples confirmed this observation under level of significance set to 0.9. Finally, notice the special periodic structure of the PF of the number of in-between packets. In our experiments we found out that this structure depends on the number of multiplexed TCP connections only. Fig. 29
8(b) provide more insights demonstrating successive in-between packets for a randomly chosen connection. Observe that bursts of non-zero in-between packets happen regularly implying that there is some sort of synchronization between TCP connections.

![Graphs of packet distribution](image)

(a) PFs of in-between packets  
(b) Number of in-between packets

Figure 8: Statistics of in-between packets for some TCP connections.

Fig. 9 demonstrates PFs and NACFs of the packet batch sizes of three randomly chosen connections. Observe that packets always arrives in batches. The mode of all distributions is located at 3. Other batch sizes are much less frequent. It is also important to observe that the tails of all distributions are relatively long implying that there are non-negligible probabilities of exceptionally big batch sizes. Practically, it implies that arrival models with fixed size of a batch may not provide accurate approximation of the packet arrival process. Taking into account droptail nature of the queue management process these observations allows us to expect that packet losses are also bursty in nature. Moreover, it is interesting to observe that there is no autocorrelation between batch sizes that belong to the same connection as illustrated in Fig. 9(b). Almost all NACF values stay within confidence limits computed according to the rule of thumb and given by $\pm 2/\sqrt{N}$, where $N$ is the number of samples used to compute the NACFs. While the presented results cannot be interpreted as quantitative measures of the aggregated arrival process from a number of TCP connections at the HSDPA wireless interface, they demonstrate that for $\tau_i > E[D]$, $i = 0, 1, \ldots, N - 1$, the packet arrival and loss processes tend to be bursty in nature.
6.3 TCP performance

Throughput obtained by a TCP source for both types of HARQ systems as a function of BER and lag-1 NACF value of the bit error process is shown in Fig. 10. The number of retransmission attempts was kept constant at 6. To emulate batch arrivals we used a SBP arrival process with $K_A(1)$ set to 0.5. There are a number of interesting phenomena to observe. First of all, observe that the throughput of the HARQ II scheme is almost always better than that of HARQ I. Also note that the presence of autocorrelation in the bit error process results in better TCP throughput for both types of HARQ schemes. Particularly, for a highly correlated process with lag-1 NACF value 0.9 TCP throughput is close to the optimal even for high values of BER. It is also important to note that for both HARQ schemes there is so-called "turning point" after which TCP throughput degrades very quickly. This behavior can be explained considering delay and loss characteristics provided to the transport layer.

The mean packet transmission time, $E[D]$, is shown in Fig. 11 while the representative behavior of the packet loss probability was shown previously in Fig. 5. As one may notice for both HARQ schemes there is significant drop-off in the mean packet transmission time after the turning point. Since TCP throughput is inverse proportional to the RTT value one may expect that this should positively affect TCP throughput. However, observing corresponding loss metrics we see that this improvement comes at the significant increase of the packet loss probability. Around the turning point the system switches its operational regimes. Up to this point packet losses are mainly caused by buffer-overflows while after it most packets are lost as a result of an insufficient number of retransmission attempts for a given FEC code.

Figure 9: Statistics of batch sizes for randomly chosen connections.
Figure 10: Delay and loss probabilities as a function of $E[W_E]$ and $K_E(1)$.

We also stress that the packet loss probability at the data-link layer grows exponentially fast and its growth depends on the autocorrelation of the bit error process. Finally, it is important to note that the packet loss probability caused by buffer overflow continues to remain at the same level even when the turning point is reached. Then it drops off to zero very quickly. The reason is the joint effect of increasing loss probability at the data-link layer and decreasing packet service time. Anyway the packet loss probability caused by buffer overflows is no longer dominant after the turning point as the packet loss probability at the data-link layer is significantly bigger.

Figure 11: Delay and loss probabilities as a function of $E[W_E]$ and $K_E(1)$. 
Depending on the dominating source of performance degradation, one may observe that the turning point divides the operational region of the wireless channel into the so-called overflow and wireless-dominated regimes. Observe that only overflow-dominated regime is application friendly. When the system enters the wireless-dominated regime TCP throughput degrades quickly calling for immediate changes in the channel adaptation parameters. These actions may include changes of the FEC code, increase of the number of retransmission attempts, change of the modulation scheme, etc.

The effect of the number of transmissions, $r$, is illustrated in Fig. 12. In both figures the value of $K_E(1)$ was set to 0.0. We used a SBP arrival process with $K_A(1)$ set to 0.5. As we expected, the increase in the number of retransmission attempts results in better throughput obtained by TCP sources for both types of HARQ systems. Practically, it allows to move the turning point of the system to the right direction along the $X$ axis. It is also important to note that for small values of $r$ (i.e. $r \leq 3$), performance of Type I HARQ is significantly better. The reason is that according to our assumption the first transmission attempt in the Type II HARQ system does not carry any redundancy symbols. Therefore, for a non-negligible BER and small values of lag-1 NACF of the bit error process packets are likely to be lost as a result of insufficient correction capability of the FEC code. Observe that modifying the Type II HARQ scheme to carry redundancy symbols in each transmission attempt (including the first one) will make its performance better than that of the Type I HARQ system across the whole range of BER and lag-1 NACF values.

![TCP throughput as a function of $r$.](image_url)
6.4 Effect of the queuing model

6.4.1 Numerical results

Let us take a look at the general effect of the queuing model involved in the FPA analysis (Fig. 13). To provide these illustrations the following parameters were used: \( r = 6 \), the lag-1 NACF value of the SBP packet arrival process was set to 0.5, batch size for Geo\(^X\) was set to 2. For easiness of comparison the Geo/Geo/1/K queuing system is shown in all figures. One of the most important conclusions is that for all queuing models behavior of TCP throughput curves remains qualitatively similar. However, as one may notice the queuing model may provide some quantitative impact.

![Figure 13: TCP throughput as a function of the queuing model.](image)

To study the effect of the queuing system in detail let us consider the
effect of the arrival and service processes in isolation. Fig. 14 shows the effect of the service time distribution for various values of $E[W_E]$ and $K_E(1)$. As one may notice the effect is noticeable for the overflow-dominated regime. The reason is that in this regime packet losses are relatively frequent and mainly caused by buffer overloaded. Packet losses implies that the system is overflowed. In this case changes in probabilistic characteristics of the packet transmission time start to affect loss performance. One may also notice that the effect is more profound for the Type II HARQ system. To understand the underlying reasons behind the observed phenomena we once again need to refer to the performance metrics provided to the transport layer.

![TCP throughput as a function of the service time distribution.](image)

Figure 14: TCP throughput as a function of the service time distribution.

PFs of the packet transmission time measured in seconds for various values of BER are shown in Fig. 15. Although for these illustrations $K_E(1)$ was set to 0.0 and $r$ was chosen to be 9, the following conclusions hold for other choices of these parameters. As one may observe the distribution of the packet transmission time for Type I HARQ is almost geometrical for small values of BER. Since most frames are transmitted in their first transmission attempts, the probability mass is mostly concentrated on those values which are slightly bigger than the packet size. When BER increases and packets start to get lost the probability mass is spread over all possible values of the packet transmission time. When BER increases even further the packet transmission time distribution changes its structure again. However, this regime is not of practical interest as wireless losses already dominate the packet loss process. Notice that for the Type II HARQ scheme and small values of BER the structure of the distribution is also far from a geomet-
rical one. The reason is that according to this HARQ scheme, first frame transmissions do not carry any redundancy symbols. When first transmissions are required to include a certain level of redundancy PFs would be similar to those of the Type I HARQ scheme. For bigger values of BER the packet service time distribution is qualitatively similar to that of the Type I HARQ scheme. Similarly to Type I HARQ approximation by geometrical distribution is not very accurate for all possible values of BER.

Figure 15: PFs of the packet transmission delay at the data-link layer.

To identify those input parameters responsible for underestimation of the TCP throughput when Geo/Geo/1/K queuing model is used, let us take a look at Fig. 16, where the coefficient of variation (CV) of the packet transmission time is shown as a function of BER and $K_E(1)$. As one may notice for small values of BER all CVs are much less than 1 for both types of HARQ. When BER gets bigger CVs increase. These observations are in perfect agreement with those made for PFs. In the impractical wireless-dominated regime CVs decrease again.

Recall now the Pollazcek-Kninchine formula for the mean waiting time in the M/G/1 system:

$$E[Q] = E[D] \frac{\rho_A}{1 - \rho_A} \frac{1 + C_A^2}{2},$$  \hspace{1cm} (29)

where $E[D]$ is the mean packet transmission time, $\rho_A = \lambda_A E[D]$ is the offered traffic load, and $C_A^2$ is the squared CV (SCV) of the packet transmission time. The result in form of (29) holds for a wide class of M/G/1 queuing systems and is approximately true for their discrete-time counterparts. This result
also qualitatively holds for finite-buffer systems we consider here. As one may see SCV in (29) is one of the factors increasing the queuing delay in Geo/G/1 type of the systems. First of all, recall that the queuing delay affects TCP throughput. Secondly, it is known that high values of SCV negatively affects the packet loss probability. Indeed, Kim and Shroff in [17] observed that the curve representing the packet loss probability as a function of the buffer size $K$ always has a similar shape as the probability of exceeding the level $K$ in an infinite buffer. In fact, many authors use tail probabilities of the buffer content distribution of infinite capacity systems to approximate the packet loss probability (see e.g. [11] for M/G/1/K, [33] for G/G/1/K and the references therein). Thus, behavior of the second moment of the packet service time distribution is the reason for increased TCP throughput when the Geo/G/1/K queuing system is used.

We may conclude that application of the Geo/Geo/1/K queuing system leads to underestimation of the actual TCP throughput. Nevertheless, the absolute difference is never bigger than 10% compared to the TCP throughput obtained using the Geo/Geo/1/K queuing system. As a result, we see that usage of exact packet service time distribution brings only small advantages compared to the approximation by a memoryless geometrical distribution. This is very important fact, especially, taking into account that memoryless service time distributions often lead to less complicated queuing models that may have closed-form solutions for performance metrics of interest.

![Graphs showing CVs of the packet transmission delay at the data-link layer](image)

(a) Type I HARQ  
(b) Type II HARQ

Figure 16: CVs of the packet transmission delay at the data-link layer.

We would like to stress that the previous discussion is only partially true for bi-directional traffic. For simplicity, we discuss the effect of bi-directional...
traffic assuming perfect channel conditions and a simple data-link layer that does not incorporate HARQ system. However, the followings conclusions hold for non-negligible BER. In this case the packet service time distribution is often bi-modal with modes located at $40/\mu$ and $1500/\mu$ bytes/s., where $\mu$ is the link rate. Let $X, X \in \{40, 1500\}$ be the RV denoting the packet sizes with the corresponding PF $\{q_{40}, q_{1500}\}$. Different sources report different probabilities for these two packet sizes. For illustrative purposes we assume that $q_{40}$ ranges from 0.1 to 0.09. Fig. 17 shows CV of the packet size distribution for different values of $q_{40}$. As one may observe, CV varies drastically and depending on $q_{40}$ may lead to under- or overestimation of the actual TCP throughput when the Geo/Geo/1/K queuing system is used in FPA analysis. Notice that when one-way TCP connections are considered, the packet service time distribution degenerates to a single value and CV is simply zero. This is the worst possible scenario for $\cdot$/Geo/1/K approximation. Finally, we would like to note that two-way traffic is also expected to result in even bigger variance of the packet service time in wireless systems due to the presence of short ACK packets. This would increase CV of the packet transmission time and make application of queuing models with geometrically distributed service times more justified.

![Figure 17: Packet size distribution for bi-directional traffic in wired networks.](image)

Now let us consider the effect of the packet arrival process. Fig. 18 demonstrates the effect of the batch size when the Geo$^{[X]}$/Geo/1/K queuing system is used for both Type I and Type II HARQ schemes. To obtain these figures the rest of parameters were set as follows: $r = 6$, $K_E(1) = 0.0$. As one may see for bursty arrival process TCP throughput significantly drops in the practical overflow-dominated regime of the system for both HARQ schemes.
The reason is that for the same packet arrival rate the packet loss probability is significantly bigger when arrivals occur in batches. As a result, the Geo/Geo/1/K queuing system may provide rather optimistic results. Notice that the behavior of all throughput curves remains qualitatively similar.

![TCP throughput as a function of the batch size](image)

Figure 18: TCP throughput as a function of the batch size $b$.

Now consider the case of the SBP/Geo/1/K queuing system. Recall, that the difference between Geo$^{[X]}$ and SBP arrival models used in this paper is that in the former case packets arrive in batches and all batches are of the same size, while according to SBP packets arrive in successive time slots in trains of different length. The average length of a train depends on $K_A(1)$ and $E[W_A]$. Note that knowing the value of $K_A(1)$ and $E[W_E]$ and taking into account that there is always one arrival in the state 1 of SBP we may estimate the average train length as

$$\frac{1}{\alpha_A} = \frac{1}{(1 - \lambda_A)E[W_A]},$$

(30)

where $E[W_A]$ is the mean of SBP, $\alpha_A$ is the transition probability from state 1 to state 2, $\lambda_A$ is the non-unit eigenvalue of the packet arrival process. Fig. 19 shows the mean train size as a function of $K_A(1)$ and $E[W_A]$.

Fig. 20 shows TCP throughput as a function of BER when the SBP/Geo/1/K queuing system is used. Other parameters were set as: $K_E(1) = 0.0$, $r = 9$. As one may observe the difference is noticeable for highly correlated arrival processes only, e.g. those with $K_A(1) > 0.6$. Notice that the window size of a TCP connection varies during its lifetime. In addition, packets from a single connection are likely to arrive in trains equally separated in time rather than
Figure 19: Average train size of SBP for different $K_A(1)$ and $E[A]$.

in batches without no time in-between. As a result, the SBP model may provide better approximation of the packet arrival process.

Figure 20: TCP throughput as a function of $K_A(1)$.

7 Conclusions

To study performance of a wireless channel shared by a certain number of TCP connections, we adopted the fixed-point approximation. The through-
put of a single TCP connection is expressed as a function of underlying layers parameters including wireless channel statistics, strength of the FEC code, number of retransmission attempts allowed for ARQ, and size of PDUs. Packet losses are allowed to occur due to both an excessive number of retransmission attempts at the data-link layer and buffer overflow at the IP layer.

The provided results show that the performance response of the wireless channel in terms of delay and loss of IP packets varies substantially for different choice of wireless channel statistics and protocol configuration at the underlying layers. Moreover, for different wireless channel statistics, different sets of underlying layers parameters results in best possible performance in terms of TCP throughput. Based on these observations we stipulate the need for a cross-layer performance control system. The main responsibility of such a system would be tuning of protocol parameters resulting in best possible throughput of TCP connections.

We also studied the error of replacing accurate queuing systems of D-BMAP/G/1/K type by simpler ones having closed form solution for the packet loss probability and the mean queuing delay, e.g. Geo/Geo/1/K. The approximation error primarily stems from inability to identify accurate approximation of the packet arrival process from a number of multiplexed TCP connections. Surprisingly, the error of replacing the actual service time distribution by its memoryless approximation is rather small. Taking into account that the proposed model is approximate in nature, the usage of Geo/Geo/1/K or M/M/1/K queuing systems should not severely affect the accuracy of TCP throughput approximation. It is important to note that qualitative behavior of TCP throughput curves remains unchanged for all tested queuing models. Finally, we note that the error of approximation may significantly increase in the presence of bi-directional TCP traffic.

We warn our readers and potential users that the proposed model is approximate in nature and is not intended for accurate estimation of TCP throughput. The main application area of the model is to study trade-offs arising between various adaptation procedures implemented at the lower layers in modern and forthcoming wireless access technologies.

References


