# Cross-layer performance control of wireless channels using active local profiles

D. Moltchanov

Original scientific paper

***Abstract*— To optimize performance of applications running over wireless channels state-of-the-art wireless access technologies incorporate a number of channel adaptation mechanisms. While these mechanisms are expected to operate jointly providing the best possible performance for current wireless channel and traffic conditions, their joint effect is often difficult to predict. To control functionality of various channel adaptation mechanisms a new cross-layer performance optimization system is sought. This system should be responsible for exchange of control information between different layers and further optimization of wireless channel performance. In this paper design of the cross-layer performance control system for wireless access technologies with dynamic adaptation of protocol parameters at different layers of the protocol stack is proposed. Functionalities of components of the system are isolated and described in detail. To determine the range of protocol parameters providing the best possible performance for a wide range of channel and arrival statistics the proposed system is analytically analyzed. Particularly, probability distribution functions of the number of lost frames and delay of a frame as functions of first- and second-order wireless channel and arrival statistics, automatic repeat request, forward error correction functionality, protocol data unit size at different layers are derived. Numerical examples illustrating performance of the whole system and its elements are provided. Obtained results demonstrate that the proposed system provide significant performance gains compared to static configuration of protocols.**

***Index Terms*— wireless channels, non-stationarity, cross-layer performance control, system design, performance evaluation.**

## I. INTRODUCTION

TO optimize performance of applications in wired networks it is often sufficient to control performance degradation caused by packet forwarding procedures. Even though this is not a trivial task, dealing with wireless networks we also have to take into account performance degradation caused by incorrect reception of channel symbols at the air interface. These errors propagate to higher layers often contributing a lot to end-to-end performance degradation. As a result, the air interface could be a 'weak point' in any end-to-end performance assurance model would ever be proposed for IP-based wireless networks.

To improve performance of wireless channels state-of-the-art wireless access technologies incorporate a number of advanced features including multiple-in multiple-out (MIMO) antenna design, adaptive modulation and coding (AMC) scheme, different automatic repeat request (ARQ) and forward error correction (FEC) procedures, transport layer error concealment functionality, etc. Although being implemented at different layers of the protocol stack all these features aim at performance improvement of information transmission over wireless channels. To decide which protocol parameters provide the best possible performance for a given traffic and channel conditions, wireless access technologies call for novel design of the protocol stack that should now include cross-layer performance optimization capabilities.

To optimize performance of applications running over wireless channels different layers of the protocol stack should be allowed to communicate between each other exchanging control information. This information should be used by a certain performance control entity to determine the set of protocol parameters providing optimized performance at any given instant of time. Depending on the state of the wireless channel and traffic characteristics of an applications, it should be possible to dynamically change protocol parameters at different layers to obtain best possible performance for given wireless channel and traffic at any given instant of time. To achieve this aim a performance control system is needed.

In this paper we propose the reactive performance control system responsible for dynamic adaptation of protocol parameters at different layers. Functionality of each component of the system is discussed in detail. Protocol parameters include error concealment capability of FEC code at the physical layer, ARQ scheme, size of frames, and buffer space at the data-link layer, rate of the active application. Depending on the current state of the wireless channel and application in terms of first- and second-order bit error and frame arrival statistics, the proposed system determines the set of protocol parameters that result in best possible performance. To determine the range of protocol parameters providing optimized performance for a wide range of channel and arrival statistics the proposed system is analytically analyzed for probability distribution functions of the number of lost frames and delay of a frame. Numerical examples illustrating performance of the system are provided. Our results demonstrate that dynamic error concealment techniques may provide significant performance gains compared to static configuration of protocols. Using the proposed system quality of the wireless channel can be dynamically regulated providing truly best effort service over the air interface.

The rest of the paper is organized as follows. The need for cross-layer interactions between protocols at the air interface is explained in Section II. The structure of the cross-layer

Manuscript received May 5, 2007; revised August 6, 2007.
This research was supported by Academy of Finland.
Dmitri Moltchanov is with Network and Protocols group, Institute of Communication Engineering, Tampere University of Technology, Tampere, Finland. E-mail: moltchan@cs.tut.fi

1845-6421/07/7022 

performance control system for wireless channels is proposed in Section III. Elements of the system are described in detail. Section IV provides mathematical foundation of the proposed performance control system. Numerical examples illustrating performance of the proposed system are provided in Section V. Conclusions are given in the last section.

## II. The need for cross-layer design

### A. Cross-layer interactions in the protocol stack

Both ITU-T OSI abstract protocol model and TCP/IP protocol model separate and isolate functionalities of each layer of the protocol stack. In these models each layer is responsible for a certain set of functions, communicates directly with the same layer of a peer communication entity and is usually unaware of specific functions of other layers. Both architectures do not allow direct communication of any kind between non-adjacent layers. Communications within the protocol stack are only allowed between adjacent layers using the so-called request-response primitives defined for service access points (SAP). Higher layers use functions provided by adjacent lower layers.

Although the layered design of the protocol stack has proven itself to be efficient in wired networks, it is often inappropriate for wireless networks. To optimize performance of applications running over wireless channels state-of-the-art wireless access technologies require novel organization of the protocol stack at the air interface. Although interfaces between adjacent layers are still preferable, there is the need for direct interactions between non-adjacent layers. In fact, the network layer and above layers often need direct interfaces to the data-link layer for handover support. Another example concerns transmission parameters including transmission mode, channel coding and data-link layer retransmissions which must be related to application characteristics (e.g. type of information, source coding, etc.), network characteristics, user preferences and context of use. In order to take decisions on traffic management, data-link layer protocols should be aware of higher layers including network and transport layers' parameters and vice versa. In future wireless access technologies we can refer to the air interface protocol architecture with interactions among different layers.

We define the cross-layer design of the protocol stack as a design that violates the respective layered structure of communication protocols. To date there were a number of proposals for cross-layers communication. We usually distinguish between the following types of cross-layer interactions: creation of new interfaces, merging of adjacent layers, design coupling, and vertical calibration of parameters across layers [1]. In order to directly exchange information between non-adjacent layers at the runtime new interfaces can be created. The information can be exchanged in the upward and downward directions. Merging of adjacent layers refers to joint definition and implementation of two or more adjacent protocols in the protocol stack. This technique allows to avoid new interfaces at the expense of more complicated implementation. Note that this approach is not inherently cross-layer but still violates the layered architecture of the protocol stack. According to the design coupling no information is exchanged between non-adjacent layers at the runtime. Instead, two protocols are just made aware of operational parameters of each other at the design phase. Vertical calibration of parameters across layers refers to the case when parameters of protocols at different layers are adjacent at the runtime such that a certain performance metric is controlled and optimized. This approach also requires new interfaces between non-adjacent layers.

The common aim of all abovementioned cross-layer communication schemes is to explicitly or implicitly exchange information between layers of the protocol stack whether at the runtime or at the design phase. Detailed review of cross-layer design approaches can be found in [1]. Several examples of the cross-layer design methodologies are discussed in [2].

### B. Cross-layer signalling schemes

In order for non-adjacent layers to communicate between each other the cross-layer signalling scheme is needed. This scheme should be responsible for exchange of control information between different layers using appropriate interfaces. Up to date a number of signalling schemes for cross-layer signaling in the protocol stack were proposed. We distinguish between in-band and out-of-band signaling. In order to communicate between TCP and radio link protocol (RLP) in wireless IP-enabled networks authors in [3] proposed to use the wireless extension header (WEH) of IPv6 protocol. The advantage of this method is that it makes use of IP data packets as in-band signalling for information exchange between the transport and the data-link layer. Another method was proposed in [4]. In order to provide communication between different layers of the protocol stack authors proposed to use ICMP messages. According to it a new message is generated whenever a certain parameter changes. The common shortcoming of two abovementioned approaches is that only few layers can actually exchange information. A different 'network' approach was proposed in [5]. According to the proposal a special network service that gathers, stores, manages and distributes information about current parameters used at mobile hosts is introduced. Those protocols that are interested in a certain parameter can access this network service. This approach provides the cross-layer functionality via a 'third party' service. Usage of local profiles instead of remote network profiles was proposed in [6]. The concept is similar to that one proposed in [5]. The only difference is that information is stored locally and there is no need to access it via the network. This results in low overhead and low delay associated with this approach. In [7] the concept of active local profiles is proposed. The principal difference compared to [6] is that active local profiles do not only store protocol parameters but implement control procedures to optimize performance of application running over wireless channels. In [8] authors proposed a dedicated cross-layer signalling protocol for communication between layers in the protocol stack. The major advantage of this protocol is that non-neighboring layers can exchange control information directly without processing at intermediate layers. This approach, however, requires additional complexity to be introduced directly to the protocol stack.

Authors in [8] provided a comparison between abovementioned signalling schemes and advocated out-of-band signalling. They argue that the signaling propagation path across the protocol stack is not efficient due to unnecessary processing of messages at intermediate layers. Additionally, the signaling message formats provided by in-band signalling schemes are not either flexible enough for signaling in both upward and downward directions or not optimized for wireless environment where the need for a new parameter to be exchanged between non-adjacent layer may occur. Finally, we note that the signalling scheme itself does not provide any advantages for a communicating entity. The ultimate goal of all cross-layer signalling schemes is optimization of protocol parameters at different layers to provide the best possible performance at any instant of time for any wireless channel and traffic conditions.

Considering the cross-layer signalling and optimization of application performance jointly one has to choose whether to use distributed or centralized control of wireless channel performance. The in-band cross-layer signalling proposals [3], [4], [8] imply the distributed performance control strategy. According to those proposals, layers exchange their information and this information should then be used by the performance control entities implemented at each layer that participates in information exchange and allows its parameters to be dynamically controlled. Such approach requires modifications to be introduced *to each layer* of the protocol stack. It was pointed out in [9] that there are a number of problems associated with distributed control strategy. Indeed, when the decision regarding changes of parameters is taken independently at each layer the resulting effect may not be straightforward. Secondly, delay associated with information exchange between non-adjacent layers can be unacceptable. Finally, to take appropriate decisions on changes of protocol parameters performance optimization subsystem must be implemented at each layer of the protocol stack that participate in performance control.

According to out-of-band signalling proposals [5], [6], layers export their current operational parameters to a certain external performance control entity via a predefined set of interfaces. This external entity not only saves information of all layers but optimize performance using controllable parameters of various protocols and then distribute information regarding what kind of protocol parameters should be used at the air interface. Thus, it makes an external intelligent cross-layer performance optimization system that incorporates features of the out-of-band cross-layer signalling system. This method is centralized in nature and generally in accordance with the signalling schemes proposed in [6], [7]. Note that distributed control is also possible with out-of-band signalling scheme.

### C. The layered architecture and the cross-layer design

Any cross-layer design of the protocol stack violates the modular design concept. Dealing with cross-layer design we have to remind the problems it may bring. As it was pointed out in [9] the layered structure of communication protocols already proved itself to be easily manageable in wired networks. Particularly, it provides a modular system design that is important to understand the operation of the whole system. At the development phase system designers must specify a number of layers, what functionalities each layer should provide and interfaces to adjacent layers. Moreover, the layered design of the system significantly simplifies implementation and further manufacturing of the system allowing, for example, reuse of components (e.g. protocols, interfaces). Indeed, protocols of the system can be developed in isolation assuming a certain set of services that a given protocol receives from the lower layer and provides to the higher layer.

Cross-layer protocol design may result in significant increase of complexity at the development and implementation phases. Functionality of the whole system may not be clearly understood due to a number of multi-layer loops. Additionally, since the modular design is no longer feasible, implementation and manufacturing costs can be high. Modification to any component of the system does not only change its own behavior but may also affect the performance of the whole system. Results of this influence are often difficult to predict. Additional efforts are required to ensure stability of the system.

Summarizing, we conclude that there should be a rational trade-off between the layered architecture and the performance optimization of wireless channel performance using cross-layer design. The cross-layer performance optimization tries to fulfill the short-term goals in terms of better performance for a given wireless access technology [9]. Clear and easily understandable layered architecture leads to long-term benefits. Among others, the low per-unit cost for a certain performance is one of the most important driving factors [9]. Thus, dealing with cross-layer performance control we have to take into account architectural considerations making as less cross-layer interactions as possible and implementing the performance control system as well isolated from the protocol stack as feasible.

## III. The performance control system

### A. Related work

To optimize performance of applications running over wireless channels state-of-the-art wireless access technologies incorporate a number of advanced features including multiple-in multiple-out (MIMO) antenna design, adaptive modulation and coding (AMC) scheme, automatic repeat request (ARQ) procedures, dynamic forward error correction (FEC), transport layer error concealment functionality, adaptive compression and coding for real-time application, etc. These mechanisms affect performance provided to applications differently and their joint effect is often difficult to predict. Recently, to evaluate joint operation of various channel adaptation techniques, cross-layer performance models started to appear. These frameworks provide a starting point in cross-layer design of wireless channels describing joint performance of two or more channel adaptation mechanisms.

Joint operation of AMC and ARQ was studied in [10]. Authors used finite-state Markov chain (FSMC) to capture changes in modulation and coding schemes. Performance of MIMO and AMC systems was studied in [11], [12], where authors introduced the notion of effective capacity of wireless channels. Joint operation of TCP congestion control and ARQ protocol was considered in [13], [14]. Performance gain provided to real-time and non-real-time applications by MIMO system was analytically shown in [15]. Liu *et. al* considered performance of TCP with AMC implemented at the physical layer, finite queue length and truncated ARQ at the data-link layer [16]. Note that implementation of AMC and MIMO system is rather complex and mainly available for wide/metropolitan area networks only.

Up to date, there were no contributions that tried to evaluate the effect of dynamic error-concealment procedures at the data-link layer assuming non-stationary wireless channel and traffic characteristics. In this paper we consider real-time applications with adaptive compression and coding. The size of protocol data units (PDU) at different layers is allowed to be dynamically changed. Using the cross-layer approach we evaluate performance that an application receives running over wireless channels at the IP layer where it is standardized. Contrarily to those studies cited above we also propose the associated performance control system.

### *B. The structure of the system*

The structure of the proposed performance control system is shown in Fig. 1, where CPOS stands for cross-layer performance optimization subsystem. The protocol stack is logically divided into three groups of protocols. The first group consists of an application itself that falls into a certain traffic class. We assume that a network is intended to deliver four traffic classes. These are conversational, streaming, interactive and background classes. Applications that fall in conversational class require to preserve time relation between information entities of the traffic stream and require stringent guarantees of end-to-end delivery of the traffic entities. Examples of these applications include real-time two-way voice communications, audio multicasting, etc. Applications that are classified to the streaming class require the network to preserve time relation between information entities of the traffic stream and do not require strict guarantees of end-to-end delivery. The most common example of these applications is the streaming video. Both interactive and background classes expect a network to guarantee the reliable delivery of information units. The difference between these two classes is that interactive class includes applications operating in request-response mode, thus, posing additional requirements on end-to-end delay. Applications that fall in background class are usually characterized by the so-called bulk transfers and do not require bounded delay in the network. Although the functionality of the performance control system for conversational and steaming applications is only considered here, the system can also be used for interactive and background applications.

Considering the defined traffic classes one may observe that there is strict correspondence between the traffic class and protocols at the transport and network layers. Those applications that require strict delay requirements usually use (RTP)UDP/IP as the combination of the transport and network layer protocols. Those applications that require a network to preserve the content of the transmission use TCP and IP at the transport and network layers, respectively. Nowadays, TCP, UDP and IP protocols are well standardized. For the sake of interoperability with existing implementations, no modifications should be made to these protocols. Indeed, changes introduced to any of these protocols may require network wide modifications. For this reason, we require that protocols of the transport and network layers and their parameters should not be controlled.

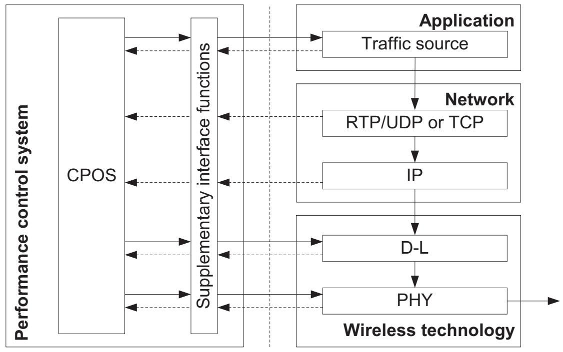



Fig. 1. The cross-layer performance optimization system.

Wireless access technology determines how the traffic is treated at the wireless channel. It defines protocols of the data-link and physical layers. These protocols are usually specific for a given wireless access technology and may incorporate advanced features such as dynamic choice of the parameters to achieve the best possible performance for given wireless channel conditions. Since the main performance degradation in wireless networks stems from the stochastic nature of wireless channel characteristics, these features of state-of-the-art wireless technologies provides a feasible option for performance control.

According to operation of the performance control system, the application firstly determines the network protocol suit (TCP/IP or (RTP)UDP/IP) to be used during the active session. This decision is taken independently of the performance control system and the mapping is strict for a given application. The application implicitly notify the CPOS about protocols that are used at the transport and network layers providing the traffic class on informational output. It should also provide information concerning the expected performance level that should be provided at the local wireless channel. Alternatively, this information can be stored in CPOS. During the whole duration of the session CPOS monitors states of wireless channel and application in terms of covariance-stationary stochastic process. The current state of the application (traffic model), the current state of the wireless channel (wireless channel model) and protocols parameters at the data-link and physical layers are used to determine performance parameters that are important for a given application. These parameters may include frame loss rate, frame delay, delay variation, etc. Then, CPOS should determine which actions should be taken to provide the best possible performance for a given application at the current instant of time, that is, whether current protocol parameters should be changed and, if yes, what changes are required. The list of actions should include change of protocol parameters at the data-link and physical layers. This capability is already available in most state-of-the-art wireless access technologies. Additionally, it may also include change of the applications parameters (e.g. rate of the codec for video and audio applications), change of the buffer space at the data-link layer, change of PDU size at different layers. The former capability is usually available for real-time applications such as streaming video or two-way voice communications. Note that when a certain application does not allow to change the rate at which the traffic is fed to the network, the feedback regarding the current rate should still exist. However, when this capability is available, controlling inputs should be provided to the source.

At the beginning of the session, the CPOS should also be made aware of controllable protocols in the protocol stack. It can be made statically at the development phase. It is important to note that protocols can be initialized with default parameters. In this case, these parameters are immediately communicated to the CPOS. Another approach is to setup a predefined set of initial parameters for each class of applications and allow CPOS to initialize protocol parameters. During the active session the CPOS controls the performance perceived by an application by setting protocol parameters in response to changing traffic and channel conditions.

### C. The cross-layer performance optimization subsystem

The core of the proposed performance control system is the CPOS. The structure of the CPOS is shown in Fig. 2. Three major components of this system are the real-time channel estimation module (rt-CEM), the real-time traffic estimation module (rt-TEM) and the performance evaluation and optimization module (PEOM). The rt-CEM is responsible for detecting changes in wireless channel statistics and estimation of the channel state in terms of the mathematical model. The rt-TEM performs the same functions for traffic observations. To enable these capabilities, wireless channel and traffic statistics are observed in real-time, pre-processed and then fed to the input of the respective change-point analyzer. Note that usage of rt-TEM is only mandatory for real-time applications that have unexpectable traffic patterns. The most common example of these applications is variable bit rate (VBR) streaming video. When the traffic pattern of an application is known in advance this block should be omitted and the predefined model should be used. Example of these applications include voice communications.

Change-point analyzers test incoming observations for changes in parameters that affect performance of applications running over wireless channels. Recently, emerged methods of measurement-based traffic modeling allowed to recognize major statistical characteristics of the traffic affecting its service performance in a network [17], [18]. According to Li and Hwang [17] the major impact on performance parameters of the service process is produced by the empirical distribution of the arrival process and the structure of its autocorrelation function (ACF). Hayek and He [18] highlighted importance of empirical distributions of the number of arrivals showing that

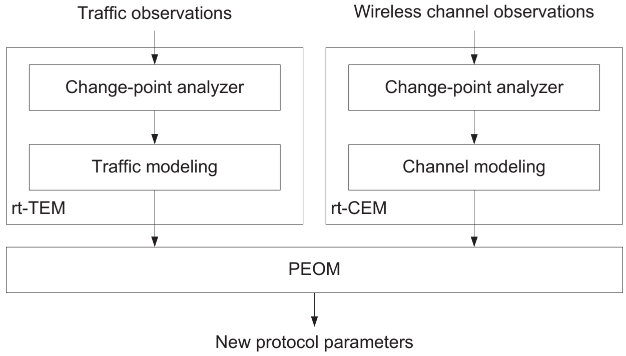



Fig. 2. The structure of the performance optimization subsystem.

the queuing response may vary for inputs with the same mean and ACF. It was also shown [17] that accurate approximation of empirical data can be achieved when both marginal distribution and ACF of the model match their empirical counterparts well. Recently, it was also shown that the wireless channel statistics including the mean frame error rate and the lag-1 autocorrelation of the frame error process significantly affect performance parameters of applications running over wireless channels at the data-link layer with hybrid ARQ/FEC [19]. In [20] authors considered the effect of bit error propagation to the IP layer with FEC procedures implemented at the data-link layer. It was also found that the mean bit error rate and the lag-1 autocorrelation of the bit error process affect the performance response at the IP layer in terms of the mean number of lost IP packet. Similar conclusions have been made in [21], where authors considered the effect of bit errors on performance of applications at the IP layer.

To monitor wireless channel statistics, SNR, bit error, or frame error processes can be used. The reason to use the bit error process is twofold. Firstly, it allows to abstract functionality of the physical layer of different wireless access technologies. As a result, a single cross-layer performance control system can be potentially applied to different wireless channels. Secondly, the bit error process is binary in nature. It allows to significantly decrease the complexity of the modeling algorithm as shown in [22]. It is also allowed to use the SNR process instead of the bit error process if the relationship between the bit error probability and the SNR value is known. Finally, the frame error process can also be used. Note, that frame error statistics can be directly obtained observing operation of ARQ protocols at the data-link layer.

Monitoring the frame error process introduces significant delays in detection of the channel state. In this case, the system may not react timely to changes in wireless channel conditions. The advantage of monitoring SNR or bit error processes is that the reaction time decreases significantly. When the relationship between the SNR value and bit error probability is already available (e.g. obtained via filed measurements) the proposed scheme can be used for SNR observations too. Direct monitoring of the bit error process of the wireless channel may provide a feasible alternative to this approach. However, in order to estimate statistics of the bit error process in real-time the source should periodically transmit a predefined information at the wireless channel such that the receiver is aware of the content of this transmission and its exact placement. This feature can be implemented using either channel equalization bits or synchronization information. However, it is still unclear how much information should be transmitted to provide a satisfactory estimator of the channel state. Authors in [23] provided some insights on this problem.

The change-point analyzers must signal those points when a change in either traffic or wireless channel statistics is detected. When a change is detected, the current wireless channel and traffic models are parameterized in the respective modeling blocks and then immediately fed to the input of PEOM. The current traffic and channel models are also stored in the respective modeling blocks for further usage. Note that it is allowed for PEOM to be activated in response to the change in either channel or traffic statistics only. Otherwise, no actions are taken except for continuous monitoring of the channel and traffic statistics.

The structure of PEOM is shown in Fig. 3. According to the system design the current traffic and channel models are fed to the input of the decision module. Taking the reference performance of a given application at appropriate layer (e.g. data-link or IP) as another input, this module decides whether the current performance is satisfactory. In order to take this decision the module containing the performance evaluation framework (PEOF) is activated. If the performance is satisfactory, no changes are required and current protocol parameters are further used. Otherwise, the current wireless channel and traffic models are used to decide whether performance can be improved and, if so, which parameters have to be changed and how. Depending on particular protocols of the protocol stack and type of the application, new protocol parameters resulting in best possible performance for a given wireless channel and traffic statistics are computed in the PEOF and then fed back to the decision module. These parameters are used till the next change in input wireless channel or traffic statistics. The PEOF may implement the performance evaluation framework or just contain a set of pre-computed performance curves corresponding to a wide range of wireless channel and traffic statistics and different configurations of the protocol stack. Due to the real-time nature of the system the latter approach is preferable.

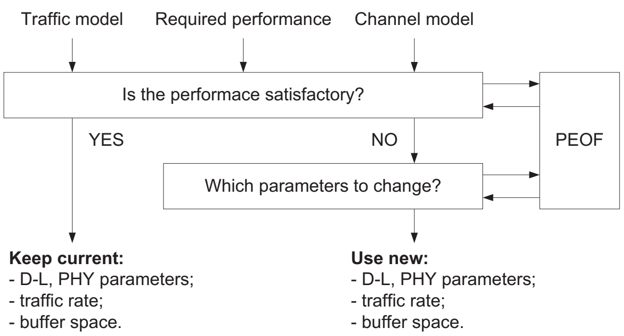



Fig. 3. The performance evaluation and optimization module.

To implement the system, the following has to be developed:

- test for detecting changes in channel and traffic statistics;
- model for channel and traffic observations;
- cross-layer extension for wireless channel model;
- performance evaluation model.

In the following sections we provide solutions to these tasks.

## IV. Components of the System

### A. Change detection in input statistics

*1) Ergodicity and stationarity:* To determine parameters of stochastic processes, such as mean and variance, based on only one, sufficiently large realization we usually assume ergodicity for stochastic observations. The sufficient condition for a stochastic process to be ergodic is $\lim_{i\to\infty} K(i) = 0$, where $K(i)$, $i = 0, 1, \dots$, is its ACF. The concept of stationarity is an advantageous property of ergodic processes. It is of paramount importance in context of modeling of stochastic observations. Practically, if certain observations are found to be non-stationary, stochastic modeling is rarely feasible. A process is said to be strictly stationary if its all $M$-dimensional distributions are the same. Only few real-life processes are strictly stationary. A process is said to be weakly (covariance) stationary if the mean value of its all sections is the same and ACF depends on the time shift only, i.e. $K(t_1, t_2) = K(i)$, $i = t_2 - t_1$. In what follows, we assume that wireless channel and traffic observations are either covariance stationary or can be segmented into a number of covariance stationary parts.

Unfortunately, there are no effective methods to statistically test whether a limited set of observations are stationary or not. Most previous studies implicitly assumed that wireless channel and traffic observations are realizations of covariance stationary stochastic processes. One of the common approaches to model wireless channel observations (e.g. SNR process) is to divide the range of observations into a number of discrete values and associate each state of the modulating Markov chain with a distinct value. In this case the number of states of the Markov model can be large leading to the so-called overfitting problem, i.e. obtained model can be used to represent a given trace but is not appropriate for other traces. The resulting process is covariance stationary. However, even if observations are truly covariance stationary, this approach is not suitable for performance control purposes. Indeed, it allows to determine parameters of the data-link layer resulting in best performance in along run. These parameters may not be optimal for any given instant of time.

*2) Wireless channel statistics:* Important observations of bit error statistics have been published in [24]. Authors found their GSM bit error traces to be non-stationary and proposed an algorithm to extract covariance stationary parts. They further used doubly-stochastic Markov process to model those parts separately. The modeled trace is finally obtained by concatenation. Among other conclusions, authors suggested that a given bit error trace can be divided into a number of concatenated covariance stationary traces. Note that the bit error probability is a function of the SNR value, and frame error probability is a function of bit error probability. As a result, we can expect the same properties for SNR and frame error observations too.

To illustrate the time-varying nature of wireless channel statistics we consider an arbitrary 11Mbps IEEE 802.11b bit error trace available from [25]. The whole number of bit error observations is $6.5E6$. We divided the whole trace into 65 non-overlapping segments each of which contains $1E5$ bit error observations. According to the setup of experiments this corresponds to approximately 24 transmitted frames where each frame was of 4096 bits in length. Point estimators of the mean, variance and lag-1 autocorrelation coefficient of these segments are shown in Fig. 4, where $E[Y]$ is the mean, $\sigma^2[Y]$ is the variance, $K_Y(1)$ is the lag-1 autocorrelation. One may see that the bit error rate changes in time significantly. Recalling the fundamental property of covariance stationary binary stochastic processes, $\sigma^2[Y] = E[Y](1-E[Y])$, we can state the same conclusion for variance. This implies that when the mean value changes so does the variance. In [20] we have shown that such ranges of mean and variance may correspond to completely different loss performance experienced at the data-link and IP layers in terms of the mean number of lost packets and mean delay experienced by a packet. Additionally, one may observe that the range of lag-1 autocorrelation is not significant ($\max K_B(1) - \min K_B(1) \approx 0.05$). In [20] we have shown that this range does not result in significant difference in the mean number of lost packets and mean delay of a packet. However, our statistical studies revealed that the range of lag-1 autocorrelation coefficient can be much larger and should be taken into account.

*3) Multimedia traffic statistics:* Video traffic statistics were often claimed to experience a high degree of variability (see [26], [27], [28] among others). Over the past decade some studies of video traffic patterns revealed that they may also experience changes in their statistical characteristics often manifesting non-stationary behavior [29], [30], [31]. Self-similarity, long-range dependence and non-stationarity are three major underlying reasons for any traffic pattern to experience high variability. Practically, high variability implies that there are large bursts in a traffic pattern. There are also long time spans during which the local average of the a traffic pattern stays well below the global average. Whatever the underlying reason for high variability, static resource allocation results in ineffective usage of resources when the load is below than expected or inappropriate performance when the load is higher.

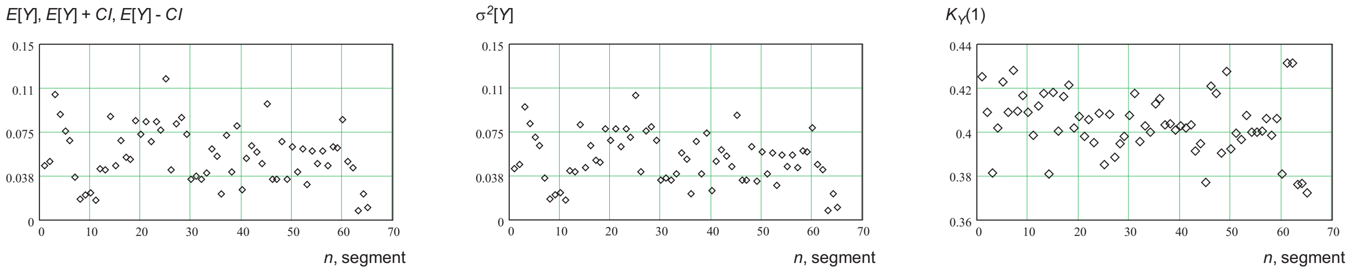
In this study we use video traffic traces available from the University of Berlin [32]. These traces are represented by sequences of frame sizes where size of the frame is measured in bits. Traces were captured for a number of coding schemes including H.263 and MPEG-4 with different quality levels. More information concerning those traces can be obtained from [33]. Although, we use H.263 variable bit rate (VBR) sequences, we have checked that the same conclusions remain valid for MPEG-4 sequences from the same traffic achieve and MPEG-1 sequences from [34].

To illustrate the time-varying nature of multimedia traffic observations we consider an arbitrary H.263 VBR trace. The whole trace contains $5E4$ observations. We divided the whole trace into 25 non-overlapping segments each of which contains 2000 observations. This corresponds to approximately 80 seconds of video data. Statistical characteristics of these segments are shown in Fig. 5, where $E[A]$ is the mean, $\sigma^2[A]$, is the variance, $K_A(1)$ is the lag-1 autocorrelation. One may see that the size of frames changes in time significantly. The range of the frame size, ($\max E[A] - \min E[A]$), is computed to be $1.15E3$, which is around 103% of the global average of all segments. The range of variance is $\max \sigma^2[A] - \min \sigma^2[A] = 1.73E6$, which is around 150% of global variance of all segments. Such range corresponds to completely different loss performance experienced at the data-link and IP layers in terms of the mean number of lost PDUs. Additionally, one may observe that the range of lag-1 autocorrelation is also significant and given by $\max K_A(1) - \min K_A(1) = 0.71$. This range may result in significant difference in the mean number of lost PDUs at the data-link and IP layers. Similar observations have been made for traces from [32] and [34].

*4) Change-point statistical tests:* Since the mean value of wireless channel and traffic observations may significantly change in time, for the proposed performance optimization and control system we suggest to monitor the mean values of respective observations. The whole task is then reduced to the so-called on-line 'change-point' statistical problem at the unknown point. Time instants at which changes occur must be detected using an on-line change-point statistical test.

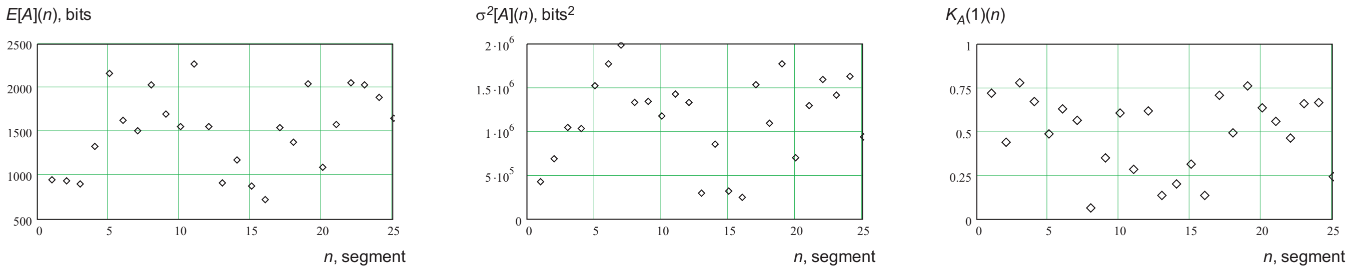
There are a number of change detection algorithms developed to date. The common approach to deal with this task is to use control charts including Shewhart charts, CUSUM charts, and exponentially weighted moving average (EWMA) charts [35], [36], [37], [38]. These charts originally came from statistical process control (SPC) where they are successfully used to monitor the quality of production. The underlying idea of control charts is that all causes of deviation of observations from the target process can be classified into two groups. These are common causes of deviation and special causes of deviation. The deviation due to common causes is the joint effect of numerous causes affecting the process. They are inherent part of the process. Special causes of deviation are not the part of the process, occur accidentally and affect the process significantly. Control charts signal the point at which special causes occur using two control limits. If observations are between them, a process is assumed to be 'in-control'. If some observations fall outside, the process is considered as 'out-of-control'.

For detecting changes in wireless channel observations the following interpretation of causes of deviation is taken. We assume that common causes of deviation are those resulting from multipath propagation environment at a certain separation distance from the transmitter. Special causes are those caused by movement of a user including changes of the distance between the transmitter and a receiver, possible shadowing of the signal by obstacles, change of the nomadic state of a user (e.g. stationary, pedestrian, vehicular). For traffic observations we assume that special causes of deviation are those resulting from specific long-term characteristics of the video including scene changes. Common causes of deviation are due to short-term stochastic nature of video data. For both traffic and wireless channel observations the whole procedure is as follows. Initially, a control chart is parameterized using statistical estimates of moments. When a change occurs, a new




(a) Mean (b) Variance (c) Lag-1 autocorrelation

Fig. 4. Mean, variance, lag-1 autocorrelation of segments of a bit error trace.




(a) Mean (b) Variance (c) Lag-1 autocorrelation

Fig. 5. Mean, variance, lag-1 autocorrelation of segments of a H.323 trace.

process is considered as 'in-control' and the control chart has to be re-parameterized according to this process.

*5) EWMA change-point detection:* Let $\{Y(n), n = 0, 1, \dots\}$ be a sequence of observations. The value of EWMA statistic at the time $n$, denoted by $L_Y(n)$, is given by

$$L_Y(n) = \gamma Y(n) + (1 - \gamma)L_Y(n - 1), \qquad (1)$$

where parameter $\gamma \in (0, 1)$ is constant.

In (1) $L_Y(n)$ extends its memory not only to the previous value but weights values of previous observations according to constant coefficient $\gamma$. In (1) this previous information is completely included in $L_Y(n-1)$. The first value of EWMA statistics, $L_Y(0)$, is usually set to the mean of $\{Y(n), n = 0, 1, \dots\}$ or, if unknown, to estimate of the mean. As a result, for on-line real-time test there should always be a certain warm-up period involving estimation of the mean.

The reason to use EWMA statistics is as follows. Although, according to (1), the most recent value always receives more weight in computation of $L_Y(n)$, the choice of $\gamma$ determines the effect of previous observations of the process on the current value of EWMA statistics. Indeed, when $\gamma \to 1$ all weight is placed on the current observation, $L_Y(n) \to Y(n)$, and EWMA statistics degenerate to initial observations. Contrarily, when $\gamma \to 0$ the current observation gets only a little weight, but most weight is assigned to previous observations. Non-zero value of $\gamma$ makes EWMA control chart more resistant to occasional outliers while reactive properties of the chart decrease. Summarizing, EWMA charts give flexibility at the expense of additional complexity in setting $\gamma$.

To parameterize the proposed EWMA control chart two parameters have to be provided. Firstly, parameter $\gamma$ determining the decline of the weights of past observations should be set. Secondly, control limits $(E[L_Y] \pm C_Y)$ must be provided. Unfortunately, when the form of distribution prior to a change is not known there is no theoretical results clarifying what the width of control limits should be. In our experimental work we found that the control limits for autocorrelated process with normal marginal distribution provide fairly accurate results

$$E[Y] \pm k\sigma[Y]\sqrt{\left(\frac{\gamma}{2-\gamma}\right)\left(\frac{1 + K_Y(1)(1-\gamma)}{1 - K_Y(1)(1-\gamma)}\right)}, \qquad (2)$$

where $\sigma[Y]$ and $K_Y(1)$ are the the standard deviation and lag-1 autocorrelation of $\{Y(n), n = 0, 1, \dots\}$, respectively, $k$ is a design parameter. Note that usage of (2) provides the trade-off between theoretical approach, where each time a change occurs probability distribution function of new in-control process must be estimated, and practical applications, where the warm-up period should be as small as possible.

The values of $k$ and $\gamma$ determine the wideness of control limits for a given process with certain $\sigma^2[Y]$ and $K_Y(1)$. These two parameters affect behavior of the so-called average run length (ARL) curve that is usually used to determine efficiency of a certain change detection procedure. ARL is defined as the average number of in-control observation up to the first out-of-control signal. Different parameters of $k$ and $\gamma$ for a given ARL, $\sigma^2[Y]$ and $K_Y(1)$ are provided in [39], [40]. Finally, $E[Y]$, $\sigma^2[Y]$ and $K_Y(1)$ are not usually known in practice and must be estimated from empirical data. Therefore, estimates of $E[Y]$, $\sigma^2[Y]$ and $K_Y(1)$ should be used in (2).

### *B. Arrival and error models*

After a shift in the mean value is detected, wireless channel and traffic observations should be represented using covariance stationary model. Note that the state of the wireless channel or application in terms of the model should be estimated using as little information as possible. As a result, there is a trade-off between the accuracy of the model and the time required to take decision regarding its parameters. Additionally, one of the fundamental requirements for a model is to be suitable for fast on-line adaptation and refinement of parameters resulting in another trade-off between the complexity of the fitting algorithm and accuracy of the model. In this paper covariance stationary segments of arrival and error processes are modeled by special cases of discrete-time batch Markovian process (D-BMP). It is known as discrete-time batch Markovian arrival process (D-BMAP, [41]) in traffic modeling and hidden Markov chain (HMM) in signal processing.

*1) Discrete-time batch Markovian process:* Assume a discrete-time environment, i.e. time axis is slotted, the slot duration is constant and given by $\Delta t = (t_{i+1} - t_i)$, $i = 0, 1, \dots$. Consider the discrete-time homogenous ergodic Markov chain $\{S(n), n = 0, 1, \dots\}$ defined at the state space $S(n) \in \{1, 2, \dots, M\}$. Let $D$ be its transition probability matrix and $\vec{\pi} = (\pi_1, \pi_2, .., \pi_M)$ be the row array containing equilibrium state probabilities of this Markov chain. Let then $\{W(n), n = 0, 1, ..\}$ be D-BMP whose underlying Markov chain is $\{S(n), n = 0, 1, \dots\}$. According to D-BMP, the value of the process is modulated by the discrete-time Markov chain $\{S(n), n = 0, 1, \dots\}$, $S(n) \in \{1, 2, \dots, M\}$. We define D-BMP as a sequence of matrices $D(k)$, $k = 0, 1, \dots$, each of which contains probabilities of transition from state to state with $k = 0, 1, \dots$, arrivals, respectively. For example, element $d_{ij}(0)$ defines transition from state $i$ to state $j$ without any arrivals, element $d_{ij}(k)$ defines transition from state $i$ to state $j$ with a batch arrival of size $k$. It is easy to see that for each pair of states $i, j \in \{1, 2, \dots, M\}$ the following

$$d_{ij}(k) = Pr\{W(n) = k, S(n) = j | S(n-1) = i\}, \quad (3)$$

are conditional probability functions of D-BMP.

Let the vector $\vec{G} = (G_1, G_2, \dots, G_M)$ be the mean vector of D-BMP, where $G_i = \sum_{j=1}^{M} \sum_{k=1}^{\infty} k d_{ij}(k)$, $i = 1, 2, \dots, M$. The mean process of D-BMP is denoted by $\{W_G(n), n = 0, 1, \dots\}$ with $W_G(n) = G_i$, when the Markov chain is in the state $i$ in the time slot $n$. The ACF of the mean process of D-BMP is [41]

$$R_G(i) = \sum_{l, l \neq 1} \phi_l \lambda_l^{i-1}, \qquad i = 1, 2, \dots, \quad (4)$$

where $\phi_l = \vec{\pi} \sum_{k=0}^{\infty} k D(k) \vec{g}_l \vec{h}_l \sum_{k=0}^{\infty} k D(k) \vec{e}$, $\lambda_l$ is the $l$s eigenvalue of $D$, $\vec{g}_l$ and $\vec{h}_l$ are $l$s left and right eigenvectors of $D$, respectively, and $\vec{e}$ is the vector of ones. Note that ACFs of D-BMP and its mean process are generally different [41].

The number of terms composing ACF of the mean process of D-BMP depends on the number of eigenvalues. The number of eigenvalues is the function of the number of states of the modulating Markov chain. Thus, varying the number of states of the modulating Markov chain we vary the number of terms composing the ACF. Recall that it is also allowed for D-BMP to have different probability functions for each different pair of states. These properties have been used in many studies to derive models of various traffic sources with sophisticated distributional and autocorrelational properties (see [42], [43], [44] among others).

In what follows we allow our D-BMP to have conditional probability functions that depend on the current state only. In this case, $D(k)$, $k = 0, 1, \dots$ have the same elements on each row. This process is known as Markov modulated batch process (MMBP). It is important that this process still has ACF distributed according to (4). We also use only two states of the modulating Markov chain. For this reason we refer to such processes as switched ones.

*2) Frame arrival process:* Let us denote the frame arrival process by $\{W_A(n), n = 0, 1, \dots\}$. In this paper, terms 'frame' and 'codeword' are used interchangeably assuming that a single frame consists of exactly one codeword. This requirement is not fundamental and can be relaxed when needed as explained below.

When MMBP $\{W_A(n), n = 0, 1, \dots\}$ is allowed to have only two states of the modulating Markov chain, $S_A(n) \in \{1, 2\}$, and each state is associated with Poissonally distributed number of arrivals in a single slot, it reduces to switched Poisson process (SPP). The marginal distribution of SPP is a weighed sum of two Poisson distributions, where weighting coefficients are given by elements of the stationary distribution of the modulating Markov chain as follows

$$\pi_{1,A} = \frac{\beta_A}{\alpha_A + \beta_A}, \qquad \pi_{2,A} = \frac{\alpha_A}{\alpha_A + \beta_A}, \quad (5)$$

where $\alpha_A$ and $\beta_A$ are transition probabilities from state 1 to state 2 and from state 2 to state 1, respectively.

The ACF of the mean process (4) of SPP reduces to

$$R_G(i) = \alpha_A \beta_A \left( \frac{G_{2,A} - G_{1,A}}{\alpha_A + \beta_A} \right)^2 (1 - \alpha_A - \beta_A)^i, \quad (6)$$

where $\alpha_A$ and $\beta_A$ are transition probabilities of the Markov modulating process from state 1 to state 2 and from state 2 to state 1, respectively. Note that $\lambda_A = (1 - \alpha_A - \beta_A)$ is the non-unit eigenvalue of the modulating Markov chain of SPP $\{W_A(n), n = 0, 1, \dots\}$. The values of $\lambda_A$ and $\lambda_G$, $\alpha_A$ and $\alpha_G$, $\beta_A$ and $\beta_G$ are the same since the modulating Markov chains of the mean process $\{W_G(n), n = 0, 1, \dots\}$ and SPP $\{W_A(n), n = 0, 1, \dots\}$ are the same.

The ACF of the SPP $\{W_A(n), n = 0, 1, \dots\}$ is [18]

$$R_A(i) = R_G(i) + E[W_A]\delta_n, \, \delta_n = \begin{cases} 1 & i = 0, \\ 0 & i = 1, 2, \dots \end{cases}, \quad (7)$$

where $E[W_A] = \pi_{1,A} G_{1,A} + \pi_{2,A} G_{2,A}$ is the mean of SPP.

In order to completely parameterize the mean process of SPP, we must provide four parameters $(G_{1,A}, G_{2,A}, \alpha_A, \beta_A)$. If we choose $G_{1,A}$ as a free variable with constraint $G_{1,A} < E[W_A]$ to satisfy $0 < \lambda_A \leq 1$, we can determine $G_{2,A}$, $\alpha_A$, and $\beta_A$ from the next set of equations [42], [44], [18]

$$\begin{cases} G_{2,A} = \frac{D[X]}{E[X] - G_{1,A}} + G_{1,A} \\ \alpha_A = \frac{(1 - K_X(1))(E[X] - G_{1,A})}{G_{2,A} - G_{1,A}} \\ \beta_A = \frac{(1 - K_X(1))(G_{2,A} - E[X])}{G_{2,A} - G_{1,A}} \end{cases}, \quad (8)$$

where $\{X(n), n = 0, 1, \dots, N\}$ are observations of the covariance stationary arrival process, $D[X]$ is the variance of $\{X(n), n = 0, 1, \dots, N\}$, $E[X]$ is the mean of $\{X(n), n = 0, 1, \dots\}$, $K_X(1)$ is the lag-1 value of the normalized ACF. Parameters $(E[X], D[X], K_X(1))$ should be estimated from empirical data.

The reason we use SPP as an example of the frame arrival process is that its parameters are easily controllable [18]. From (8) one may note that there is a degree of freedom in choosing the mean arrival rate in state 1. Indeed, $G_{1,A}$ can take on any value from $(0, E[X])$ and still match the ACF and the mean of the arrival process. As a result, distributions of these processes are different for different choices of $G_{1,A}$.

*3) Bit error model:* Let us denote the bit error process by $\{W_E(n), n = 0, 1, \dots\}$. When MMBP is allowed to have two states only, and at most a single arrival is allowed in a slot, it reduces to switched Bernoulli process (SBP). Since this process has only two states of the modulating Markov chain, its ACF (4) reduces to (6). Normalized ACF (NACF) is then $K_G(i) = \lambda_E^i$, $i = 1, 2, \dots$. It is clear that the NACF of the mean process of SBP exhibits geometrical decay that may produce fair approximation of empirical NACFs exhibiting nearly geometrical decay for small lags. It should also be noted that lag-1 autocorrelation coefficient completely specifies the behavior of NACF.

Since we are dealing with covariance stationary binary process, only mean and lag-1 autocorrelation coefficient out of triplet $(E[W_E], \sigma^2[W_E], K_E(1))$ have to be captured. In our previous work [20], [22] we demonstrated that there is SBP model exactly matching mean and lag-1 autocorrelation of covariance stationary bit error observations. This model is given by

$$\begin{cases} \alpha_E = (1 - K_E(1))E[W_E] \\ \beta_E = (1 - K_E(1))(1 - E[W_E]) \end{cases}, \begin{cases} f_{1,E}(1) = 0 \\ f_{2,E}(1) = 1 \end{cases}, \quad (9)$$

where $f_{1,E}(1)$ and $f_{2,E}(1)$ are probabilities of 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.

### C. Extension to the data-link layer

Bit error model cannot be directly used for performance control of application and should be firstly extended to the data-link layer. Assume that bits are consecutively transmitted over wireless channels, the length of frames at the data-link layer is constant and equals to $m$ bits.

Consider the stochastic process $\{N(l), l = 0, 1, \dots\}$, $N(l) \in \{0, 1, \dots, m\}$, describing the number of incorrectly received bits in consecutive bit patterns of length $m$. This process can be completely parameterized using parameters of the bit error process. Let us denote the probability of going from the state $i$ to state $j$ for the Markov chain $\{S_N(l), l = 0, 1, \dots\}$ with exactly $k$, $k = 0, 1, \dots, m$ incorrectly received bits in a bit pattern of length $m$ by $d_{N,ij}(k) = Pr\{N(l) = k, S_N(l) = j | S_N(l-1) = i\}$. Let the set of matrices $D_N(k)$, $k = 0, 1, \dots, m$, contain these transition probabilities. To completely parameterize $\{N(l), l = 0, 1, \dots\}$ we have to determine $m$-step transition probabilities between states $i$ and $j$ of the modulating Markov chain $\{S_E(n), n = 0, 1, \dots\}$ with $k$, $k = 0, 1, \dots, m$ incorrectly received bits in a bit pattern of length $m$. In this case, $\{N(l), l = 0, 1, \dots\}$, $N(l) \in \{0, 1, \dots, m\}$ is doubly stochastic with underlying Markov chain $\{S_N(l), l = 0, 1, \dots\}$, defined on the same state space as $\{S_E(n), n = 0, 1, \dots\}$. Thus, $S_N(l) = S_E(l) \in \{1, 2\}$. Matrices $D_N(k)$, $k = 0, 1, \dots, m$, can be found using $D_E(k)$, $k = 0, 1$, of the bit error process $\{W_E(n), n = 0, 1, \dots\}$ as

$$\begin{aligned} &D_N(0) = D_E^m(0), \\ &D_N(1) = \sum_{k=m-1}^{0} D_E^{m-k-1}(0) D_E(1) D_E^k(0), \\ &\dots \\ &D_N(m) = D_E^m(1), \end{aligned} \quad (10)$$

where $D_N(i)$, $i = 2, 3, \dots, m-2$ can be obtained by induction from $D_N(1)$ or $D_N(m-1)$. The easiest way is to induce $D_N(i)$, $i = 2, 3, \dots, \lfloor m/2 \rfloor$, from $D_N(1)$ and $D_N(i)$, $i = m, m-1, \dots, \lceil m/2 \rceil$ from $D_N(m-1)$. We only need $D_N(k)$, $k << m$. Note that computation according to (10) is still a challenging task. This becomes impossible when $m$ is sufficiently large. Instead, we may use the recursive method as outlined below. To estimate (10) recursive procedure is proposed in [45].

Consider now the frame error process $\{F(l), l = 0, 1, \dots\}$, $F(l) \in \{0, 1\}$, where '0' indicates the correct reception of a frame, '1' denotes the incorrect frame reception. Process $\{F(l), l = 0, 1, \dots\}$ is modulated by the underlying Markov chain $\{S_F(l), l = 0, 1, \dots\}$. The state space of $\{S_F(l), l = 0, 1, \dots\}$ and $\{S_N(l), l = 0, 1, \dots\}$ is the same. Denote the probability of going from state $i$ to state $j$ for the Markov chain $\{S_F(l), l = 0, 1, \dots\}$ with correct $(F(l) = 0)$ or incorrect $(F(l) = 1)$ reception of the frame by $d_{F,ij}(k)$, $k = 0, 1$. These probabilities are then combined in matrices $D_F(0)$ and $D_F(1)$. $\{N(l), l = 0, 1, \dots\}$, $N(l) \in \{0, 1, \dots, m\}$ is related to $\{F(l), n = 0, 1, \dots\}$, $F(l) \in \{0, 1\}$ as

$$D_F(0) = \sum_{k=0}^{F_T - 1} D_N(k), \qquad D_F(1) = \sum_{k=F_T}^{m} D_N(k), \quad (11)$$

where $F_T$ is the frame error threshold. Expressions (11) are interpreted as follows: if the number of incorrectly received bits in a frame is greater or equal to the computed value of the frame error threshold ($k \geq F_T$), frame is incorrectly received and $F(l) = 1$. Otherwise ($k < F_T$), it is correctly received and $F(l) = 0$. Setting $F_T = 1$ we get no FEC at the data-link layer. Practically, $(F_T - 1)$ denotes the number of bit errors that can be corrected by a FEC code.

### D. Performance evaluation

*1) Service process of the wireless channel:* The straightforward way to represent the frame transmission process over a dedicated constant bit rate (CBR) wireless channel is to use $G_A/G_S/1/K$ queuing system, where $G_A$ is the frame arrival process, $G_S$ is the service process of the wireless channel, $K$ is the capacity of the system. Here, the service process is defined as times required to successfully transmit successive frames over the wireless channel. Characteristics of this process are determined by the frame error process and error concealment schemes of the data-link layer.

It is known that both interarrival time of frames and transmission time of frames till successful reception are generally not independent but autocorrelated. These properties make analysis of $G_A/G_S/1/K$ queuing system quite complex task even when arrival and service processes are modeled by Markovian processes. Indeed, theoretical background of queuing systems with autocorrelated arrival and service processes is not well-studied. Among few others, one should mention BMAP/SM/1 queuing system and some modifications considered in [46], [47], [48]. Analysis of these systems is more computationally intensive compared to queuing systems with renewal service process. It usually involves imbedded Markov chains of high dimensions. From this point of view, such a performance model does not provide significant improvements over other approaches.

*2) Service process with SW-ARQ/FEC:* Consider a class of preemptive-repeat priority systems with two Markovian arrival processes. We allow both processes to have arbitrary autocorrelation structures of homogenous Markovian type. Assume that the first arrival process represents the frame arrival process from the traffic source. To provide an adequate representation of unreliable transmission medium, we assume that the second arrival process is one-to-one mapping from the frame error process. That is, every time an error occurs, an arrival happens from this arrival process. In what follows, we refer to this process as 'artificial arrival process'. According to this mapping, probabilistic properties of the stochastic model remain unchanged. Making this process to be high priority one, and allowing its arrivals to interrupt ongoing service of low priority arrivals, we assure that when an arrival occurs from this process it immediately seizes the server for service, while the ongoing service is interrupted. A frame whose service is interrupted remains in the system (if allowed) and enters the server again after service completion of high priority arrival. The service provided till the point of interruption is lost. It is interpreted as an incorrect reception of the frame from the traffic source. The priority discipline is referred to as preemptive-repeat.

To emulate behavior of SW-ARQ protocol, we assume an infinite number of retransmission attempts. We also assume that the feedback channel is completely reliable (perfect). Indeed, feedback acknowledgements are usually small in size and well protected by FEC code. We also assume that the feedback is instantaneous. These assumptions were tested and used in many studies and found to be appropriate for (relatively) high speed wireless channels [49], [50], [51], [52]. Since we extended the wireless channel model to the data-link layer, FEC capabilities are explicitly taken into account. Note that the described model is also suitable to represent 'ideal' SR-ARQ scheme as in [53], [54]. In SR-ARQ frames are continuously transmitted and only incorrectly received frames are selectively requested. According to 'ideal' operation of SR-ARQ, round trip times (RTT) are assumed to be zero. In this case SR-ARQ and SW-ARQ schemes become identical and can be represented using the proposed model.

Analysis of queuing systems with preemptive-repeat priority discipline is still a challenging task. However, a number of assumptions can be further introduced to make the queuing model less complicated. In what follows, we limit our model to the discrete-time environment and require each arrival from both arrival processes to have a service time of one slot in duration. According to such a system, arrivals occur just before the end of slots. Since there can be at most one arrival from the arrival process representing the frame error process of the wireless channel, these arrivals do not wait for service, enter the service in the beginning of nearest slots, and, if observed in the system, are being served. To provide adequate representation of erroneous nature of the wireless channel, we also have to ensure that all these arrivals are accommodated by the system. Following these assumptions, it is no longer needed to require preemptive-repeat priority discipline. Since all arrivals occur simultaneously in batches, it is sufficient for such a system to have non-preemptive priority discipline.

*3) Description of the system:* Consider D-BMAP/D/1/K queuing system, where the arrival process, denoted by $\{W(n), n = 0, 1, \dots\}$, is the superposition of $\{W_F(n), n = 0, 1, \dots\}$ and $\{W_A(n), n = 0, 1, \dots\}$. Analysis of D-BMAP/D/1/K queuing system was carried out in many studies. Here, we take the method of imbedded Markov chain.

Time diagram of D-BMAP/D/1/K queuing system is shown in Fig. 6. According to such a system, frames arrive in batches, batch of frames arrives just before the end of a slot. Frames are not allowed to enter service immediately and the service of any frame starts at the beginning of a slot. Frames depart from the system just after a batch arrival (if any). The state of the system is observed just after the departure (if any) and these points are imbedded Markov points. The sojourn (service) time is counted as the number of slots spent by a frame in the system. The system can accommodate at most *K* frames. We assume partial batch acceptance strategy. According to this strategy, if a batch of $R$ frames arrives when $k$ frames are in the system and $R > (K - k)$, only $(K - k)$ frames are accommodated and $(R - K + k)$ frames are discarded.

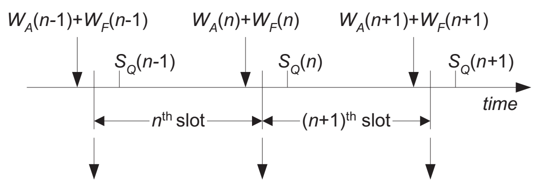



Fig. 6. Time diagram of D-BMAP/D/1/K discrete-time queuing system.

Observing Fig. 6, it can be deduced that the arrival from the frame error process is not accepted by the system in the slot $(n + 1)$ if and only if the number of customers in the system in the slot $(n - 1)$ is zero, there is an arrival of $K$ frames in the time slot $n$, and one frame arrives from the frame error process in the slot $(n + 1)$. Contrarily, if there is at least one frame in the system in the slot $(n - 1)$, one frame departs at the boundary between slots $n$ and $(n + 1)$, and there is always at least one position in the system for the next arrival. Thus, the frame from the frame error process (if any) is not lost in the slot $(n+1)$. To assure that the frame from the frame error process is always accepted by the system we do not allow the overall number of arrivals from both processes to be more than $(K-1)$. This implies that the maximum number of arrivals from the frame arrival process is $(K-2)$, that is usually sufficient for real applications.

Complete description of the queuing system requires two-dimensional Markov chain $\{S_Q(n), S(n), n = 0,1,\dots\}$ imbedded at the moments of frame departures from the system, where $S(n) = S_A(n) \otimes S_F(n)$ is the state of superposition of the frame arrival and frame error processes, and $S_Q(n) \in \{0,1,\dots,K-1\}$ is the number of frames in the system just after frame departures. Introducing matrices $D(\geq k)$, $k = 0,1,..,K-1$, containing transition probabilities with at least $k = 0,1,..,K-1$ arrivals, respectively, one can define the transition probability matrix, $T$, of the Markov chain $\{S_Q(n), S(n), n = 0,1,\dots\}$ as usual (see [55] among many others). Let $\vec{x} = (x_{0,1},..,x_{K-1,M})$ be the row array containing steady-state probabilities of $\{S_Q(n), S(n), n = 0,1,..\}$. Solving matrix equations $\vec{x}T = \vec{x}$, $\vec{x}\vec{e} = 1$, one can compute steady-state probabilities $x_{kj} = \lim_{n\to\infty} Pr\{S_Q(n) = k, S(n) = j\}$. There are a number of algorithms to compute these probabilities [56], [57], [58].

*4) Loss and delay performance:* The proposed system has been completely analyzed for loss and delay performance in [45]. Particularly, it was demonstrated that the probability function of the number of $l = 1,2,\dots,K-2$ lost frames in a slot from the frame arrival process is given by

$$f_L(l) = \frac{\sum_{k=3}^{K-1} \vec{x}_k D(0, K-k+l)\vec{e}}{\vec{\pi}_A \left(\sum_{i=1}^{K-2} D_A(i)\right) \vec{e}_A} + \frac{\sum_{k=2}^{K-1} \vec{x}_k D(1, K-k+l-1)\vec{e}}{\vec{\pi}_A \left(\sum_{i=1}^{K-2} D_A(i)\right) \vec{e}_A}, \quad (12)$$

where $D(l,k)$, $l = 0,1$, $k = 0,1,\dots,K-2$ are transition probability matrices of the superposed arrival process with exactly $l$ arrivals from the frame error process and $j$ arrivals from the frame arrival process, $\vec{x}_k = (x_{k1}, x_{k2}, \dots, x_{k(M_F M_A)})$ is the vector containing steady-state probabilities that there are $k$ frames in the system and the state of the modulating Markov chain of the superposed arrival process is $i = 1,2,\dots,M_F M_A$, $\vec{e}$ is the vector of ones of appropriate size, $\vec{\pi}_A$ is the steady-state vector of the modulating Markov chain of the frame arrival process. Matrices $D(l,k)$, $l = 0,1$, $k = 0,1,\dots,K-2$ are given by

$$D(l,k) = D_F(l) \otimes D_A(k),\ l = 0,1,\ k = 0,1,\dots. \quad (13)$$

Let the random variable $Q$, $Q \in \{1,2,\dots\}$ denote the full delay in the system (sojourn time) experienced by arrival from the frame arrival process and let $f_Q(q) = Pr\{Q(n) = q | W_A(n) > L(n)\}$, $q = 1,2,\dots$ be its probability function, where $W_A(n)$ is the number of arriving frames from the frame arrival process in the slot $n$, $L(n)$ is the number of lost frames in the slot $n$. In [] the delay was found to be

$$\begin{aligned} f_Q(1) &= \vec{f}_Q(1,0)\vec{e} \\ f_Q(2) &= \vec{f}_Q(2,0)D(0,\cdot)\vec{e} \\ f_Q(q) &= \sum_{i=2}^{q} \vec{f}_Q(i,0)T(q-i,q-2)D(0,\cdot)\vec{e}, \\ f_Q(q) &= \sum_{i=2}^{K-1} \vec{f}_Q(i,0)T(q-i,q-2)D(0,\cdot)\vec{e}, \end{aligned} \quad (14)$$

where $\vec{e}$ is the unit vector of appropriate size, $\vec{f}_Q(q,0)$, $q = 1,2,\dots,K-1$, are the vectors containing probabilities that the tagged frame arriving in the slot $n$ is at the position $q$ just after the slot boundary between slots $n$ and $(n+1)$ and the state of the superposed arrival process is $j$ given that at least one frame arriving from frame arrival process is not lost, $T(i,m)$, $i = 0,1,\dots,m$, $i \leq m$ are transition probability matrices with exactly $i$ arrivals from $\{W_F(n), n = 0,1,\dots\}$ in $m$, $m = 1,2,\dots$ successive slots, starting from the slot $(n+1)$, $D(l,\cdot)$, $l = 0,1$ are transition probability matrices with exactly 0 or 1 arrivals from the frame error process.

Matrices $D(l,\cdot)$, $l = 0,1$ are given by

$$D(l,\cdot) = \sum_{j=0}^{K-2} D(l,j), \qquad l = 0,1. \quad (15)$$

Vectors $\vec{f}_Q(q,0)$, $q = 1,2,\dots,K-1$ are given by

$$\begin{aligned} \vec{f}_Q(q,0) = &\frac{\sum_{i=q}^{K-2} \vec{x}_0 D(0,i)\psi_{K,i} + \sum_{i=q-1}^{K-2} \vec{x}_0 D(1,i)\psi_{K-1,i}}{Pr\{W_A(n) > L(n)\}} + \\ &+ \frac{\sum_{k=1}^{q}\sum_{i=q-k+1}^{K-2} \vec{x}_k D(0,i)\psi_{K-k,i}}{Pr\{W_A(n) > L(n)\}} + \\ &+ \frac{\sum_{k=1}^{q-1}\sum_{i=q-k}^{K-2} \vec{x}_k D(1,i)\psi_{K-k-1,i}}{Pr\{W_A(n) > L(n)\}}. \end{aligned} \quad (16)$$

where $\psi_{v,i}$ is the probability that the tagged arrival is accommodated at the place $i$ in the system when there are $v$ waiting positions available for arrivals from the frame arrival process, $Pr\{W_A(n) > L(n)\} = Pr\{W_A(n) \geq 1\} - Pr\{L(n) = W_A(n) \geq 1\}$ is the probability that at least one arrival is not lost in the slot $n$, and $Pr\{L(n) = W_A(n) \geq 1\}$ is the probability that all arrivals from the frame arrival process in the slot $n$ are lost.

Probabilities $\psi_{v,i}$, are given by

$$\psi_{v,i} = \frac{1}{\min(v,i)}. \quad (17)$$

Probability $Pr\{W_A(n) > L(n)\}$ is found to be

$$Pr\{W_A(n) > L(n)\} = \vec{\pi}_A \left(\sum_{i=1}^{K-2} D_A(i)\right) \vec{e}_A - \sum_{i=1}^{K-2} \vec{x}_{K-1} D(1,i)\vec{e}. \quad (18)$$

Matrices $T(i,m)$, $i = 0, 1, \ldots, m$, $i \leq m$ are given by

$$
\begin{aligned}
&T(0,m) = D^m(0,\cdot),\\
&T(1,m) = \sum_{k=m-1}^{0} D^{m-k-1}(0,\cdot)D(1,\cdot)D^k(0,\cdot),\\
&\ldots\\
&T(m-1,m) = \sum_{k=m-1}^{0} D^{m-k-1}(1,\cdot)D(0,\cdot)D^k(1,\cdot),\\
&T(m,m) = D^m(1,\cdot).
\end{aligned} \tag{19}
$$

## V. NUMERICAL EXAMPLES

### A. Bit error models

In this section we use a number of SBP wireless channel models with different means and lag-1 autocorrelations. We constructed 90 models of the bit error process as follows. For each mean out of $E[W_E] \in \{0.1, 0.02, \ldots, 0.09\}$ we generated models with the following lag-1 autocorrelations $K_E(1) \in \{0.0, 0.1, \ldots, 0.9\}$. Parameters of the bit error models, $\alpha_E$ and $\beta_E$, as a function of $E[W_E]$ and $K_E(1)$ are shown in Fig. 7.

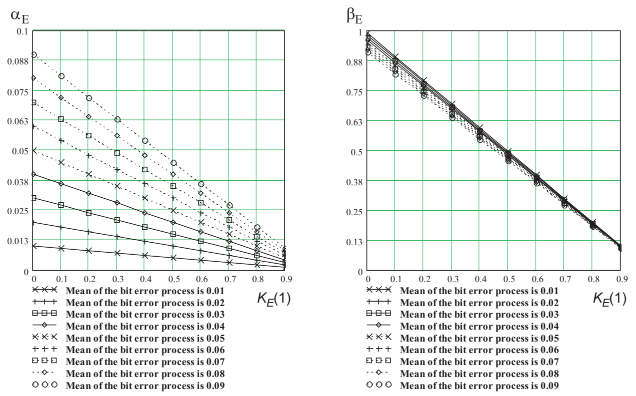



Fig. 7. Parameters $\alpha_E$ and $\beta_E$ as a function of $E[W_E]$ and $K_E(1)$.

### B. Performance evaluation

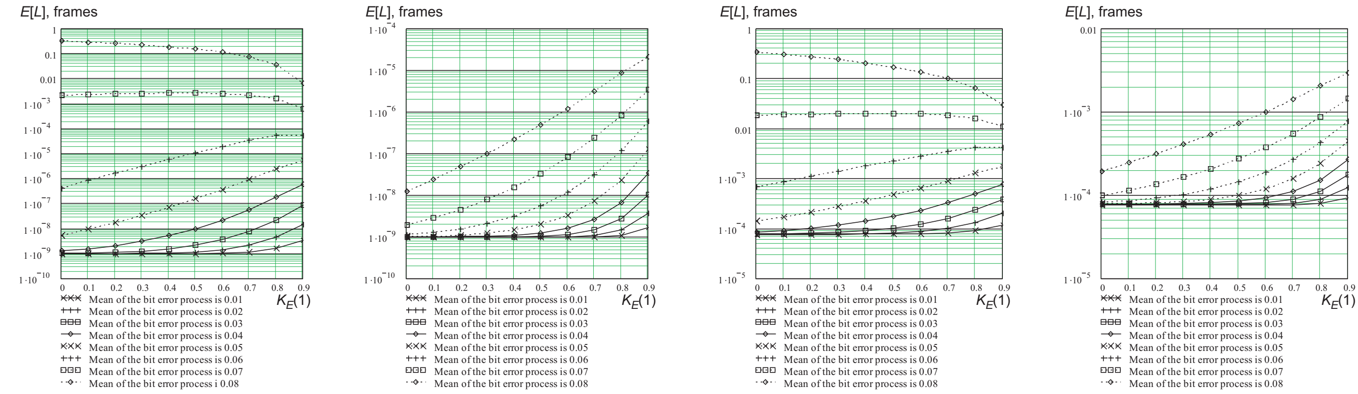
We study performance of two Bose-Chaudhuri-Hochquenghem (BCH) FEC codes denoted by triplet $(m, n, l)$, where $m$ is the length of the frame in bits, $n$ is the number of data bits in a frame and $l$ is the number of incorrectly received bits that can be corrected. The codes are $(255, 131, 18)$ and $(255, 87, 26)$, whose rates are approximately $1/2$ and $1/3$, respectively. According to our model, the former coding scheme delivers approximately $1.507$ times more data bits in a single slot.

Fig. 8 demonstrates the mean loss performance for both FEC codes, different wireless channel conditions and different statistics of the arrival process. The system of interest is SPP$_A$+SPP$_E$/D/1/K. Parameters of the frame arrival process were set to $E[W_A] = 0.5$, $D[W_G] = 0.5$, $G_{1,A} = 2E[W_A]/9$. If the performance metric of interest is the mean number of successfully delivered frames, one can see from Fig. 8 that $(255, 87, 26)$ FEC code outperforms $(255, 131, 18)$ FEC code for all conditions of the wireless channel. However, if the performance metric of interest is the mean number of successfully delivered bits, the result can be different. This stems from the fact that $(255, 131, 18)$ FEC code tries to transmit approximately $1.5$ times more data bits in a single slot. As a result, a new comparison should be made. The mean number of successfully delivered bits in a slot can be computed as $(E[W_A(n)|W_A(n) \geq 1] - E[L(n)|W_A(n) \geq 1])n$, where $n$ is the number of data bits in a single frame. For $K_G(1) = 0.0$, it can be computed that $(255, 87, 26)$ FEC code is only better when the bit error rate is $0.08$ and the lag-1 autocorrelation is less than $0.4$. The performance at the data-link layer in terms of both metrics is significantly worse when the lag-1 autocorrelation of the mean frame arrival process increases from $0.0$ to $0.5$. Note that the final decision on the choice of the FEC code should also take into account the mean delay performance of both FEC codes as explained below.

Fig. 9 illustrates the delay experienced by a frame when $(255, 131, 18)$ and $(255, 87, 26)$ FEC codes are used at the data-link layer. Recall that in real environment ARQ protocols sometimes limit the number of retransmission attempts. For example, if the number of retransmission attempts is limited by 6, ARQ protocol with $(255, 131, 18)$ FEC code does not perform well on a wireless channel with the bit error rate higher than $0.07$. For most considered wireless channel conditions, ARQ protocol with $(255, 87, 26)$ FEC code performs better than with $(255, 131, 18)$ FEC code. For example, for $E[W_E] = 0.08$, $K_E(1) = 0.4$, $K_G(1) = 0.0$ approximately 3 retransmission attempts are required on average. For the same wireless channel conditions and traffic parameters ARQ protocol with $(255, 131, 18)$ FEC code requires around 36 retransmission attempts which is not tolerable. Note that when the lag-1 autocorrelation of the mean frame arrival process of SPP$_A$ increases to $K_G(1) = 0.5$ and the wireless channel conditions are $E[W_E] = 0.08$ and $K_E(1) = 0.4$, even $(255, 87, 26)$ FEC code leads to poor performance as it now requires approximately 5 retransmission attempts to deliver a single frame correctly.

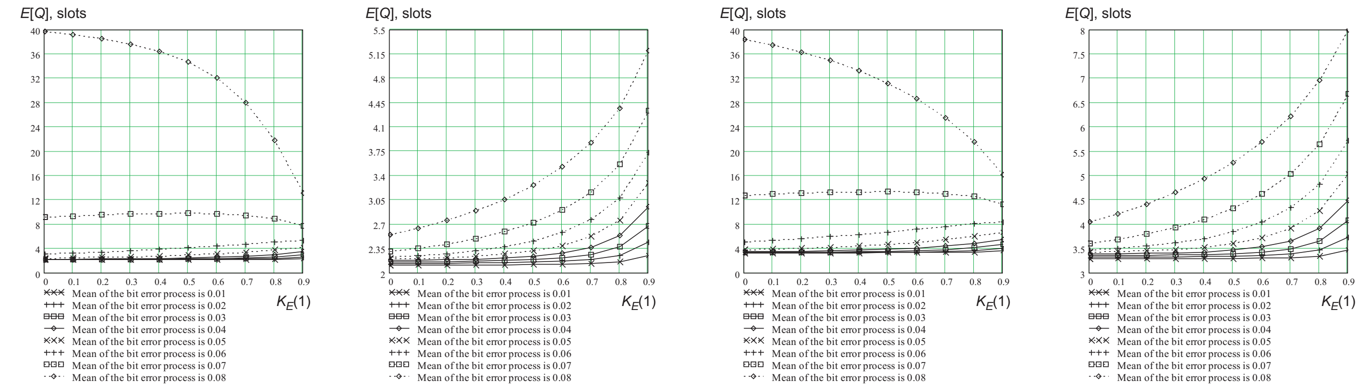
### C. Performance control system

Let us consider theoretical performance of the whole system. We assume that a certain channel is covariance stationary for 50% of time with mean bit error rate $E[W_E] = 0.08$ and lag-1 autocorrelation $K_E(1) = 0.0$ and then changes to covariance stationary process with $E[W_E] = 0.02$, $K_E(1) = 0.0$. Frame arrival process is assumed to be covariance stationary with $E[W_A] = 0.5$, $\sigma^2[W_A] = 0.5$, $K_A(1) = 0.0$. Only two FEC codes were allowed to be used. They are $(255, 131, 18)$ and $(255, 87, 26)$. ARQ was enabled. For all experiments the buffer space was set to $K = 40$. Initially, the performance control system was initialized with the most powerful FEC code, $(255, 87, 26)$. We compare the results obtained for our system to those obtained when no performance control algorithm used at the wireless channel. Since we are dealing with multimedia applications, when the trade-off between losses and delays occurs the preference is given to delay.




(a) $(255, 131, 18)$, $K_G(1) = 0.0$ (b) $(255, 87, 26)$, $K_G(1) = 0.0$ (c) $(255, 131, 18)$, $K_G(1) = 0.5$ (d) $(255, 87, 26)$, $K_G(1) = 0.5$

Fig. 8. Case study: the mean loss response for different FEC codes.




(a) $(255, 131, 18)$, $K_G(1) = 0.0$ (b) $(255, 87, 26)$, $K_G(1) = 0.0$ (c) $(255, 131, 18)$, $K_G(1) = 0.5$ (d) $(255, 87, 26)$, $K_G(1) = 0.5$

Fig. 9. Case study: the mean delay response for different FEC codes.

Performance results of the proposed control system are shown in Table I, where $E[Q]$ is delay measured in slots, $T$ is throughput measured in bits per slot. It is clear that usage $(255, 87, 26)$ FEC code for bit error rate $E[W_E] = 0.08$ and $(255, 131, 18)$ FEC code for $E[W_E] = 0.02$ provides the best possible option in terms of throughput and delay. As shown in the Table I other choices result in significantly worse performance. Table II demonstrates results obtained for the same wireless channel parameters and covariance stationary frame arrival process with $E[W_A] = 0.5$, $\sigma^2[W_A] = 0.5$, $K_A(1) = 0.5$. One can notice that the autocorrelation of the frame arrival process affects delay performance of the frame service process more severely compared to throughput.

TABLE I
PERFORMANCE RESULTS: $E[W_A] = 0.5$, $\sigma^2[W_A] = 0.5$, $K_A(1) = 0.0$.

| $E[W_E] = 0.08$ | $E[W_E] = 0.02$ | $E[Q]$ | $T$ |
|---|---|---|---|
| (255,131,18) | (255,131,18) | 20.8899 | 44.9873 |
| (255,131,18) | (255,87,26) | 20.8909 | 33.9873 |
| (255,87,26) | (255,131,18) | 2.3346 | 54.4999 |
| (255,87,26) | (255,87,26) | 2.3355 | 43.4999 |

TABLE II
PERFORMANCE RESULTS: $E[W_A] = 0.5$, $\sigma^2[W_A] = 0.5$, $K_A(1) = 0.5$.

| $E[W_E] = 0.08$ | $E[W_E] = 0.02$ | $E[Q]$ | $T$ |
|---|---|---|---|
| (255,131,18) | (255,131,18) | 20.8176 | 43.1183 |
| (255,131,18) | (255,87,26) | 20.8176 | 32.1200 |
| (255,87,26) | (255,131,18) | 3.6773 | 54.4866 |
| (255,87,26) | (255,87,26) | 3.6773 | 43.4883 |

Let us now simulate performance of the proposed control system. For this reason two bit error traces were generated. Each trace consists of two covariance stationary parts. First 5E4 samples are observation of covariance stationary process with $E[W_E] = 0.08$, $K_E(1) = 0.0$, latter $5E4$ observations were generated using process with $E[W_E] = 0.02$, $K_E(1) = 0.0$. Parameters of EWMA control charts were always set such that in-control ARL is kept at $400$. Performance results of the proposed system with both FEC and ARQ enabled are demonstrated in Tables III and IV, where they are compared with theoretical results and fixed FEC code. One can see that fixed FEC codes leads to non-optimal performance of information transmission. Using the proposed performance control system we achieve near optimal results in terms of both delay and throughput. These results are close to theoretical which are presented in Tables I and II.

TABLE III
PERFORMANCE RESULTS: $E[W_A] = 0.5$, $\sigma^2[W_A] = 0.5$, $K_A(1) = 0.0$.

| | Bit error trace 1 | | Bit error trace 2 | |
|---|---|---|---|---|
| Approach | $E[Q]$ | $T$ | $E[Q]$ | $T$ |
| Theoretical | 2.3346 | 54.4999 | 2.3346 | 54.4999 |
| Control system | 2.2759 | 58.8986 | 2.2469 | 62.6552 |
| (255,131,18) | 13.6519 | 53.1967 | 8.8977 | 60.2612 |
| (255,87,26) | 2.2761 | 43.4999 | 2.2469 | 43.4999 |

TABLE IV
PERFORMANCE RESULTS: $E[W_A] = 0.5$, $\sigma^2[W_A] = 0.5$, $K_A(1) = 0.5$.

| | Bit error trace 1 | | Bit error trace 2 | |
|---|---|---|---|---|
| Approach | $E[Q]$ | $T$ | $E[Q]$ | $T$ |
| Theoretical | 3.6773 | 54.4866 | 3.6773 | 54.4866 |
| Control system | 3.5689 | 58.8201 | 3.5140 | 62.0498 |
| (255,131,18) | 14.2067 | 52.0059 | 10.4926 | 59.1888 |
| (255,87,26) | 3.5689 | 43.4902 | 3.5140 | 43.4914 |

## VI. CONCLUSIONS

For wireless access technologies with dynamic adaptation of the protocol parameters to time-varying wireless channel and traffic conditions the performance control system has been proposed. Controllable parameters include the strength of FEC code, ARQ functionality, size of PDU at different layers, the rate with which traffic is generated. According to the proposed system it is still possible to implement protocols at different layers independently. The only requirement we impose is that certain protocols should be controllable and must export information about their current parameters using appropriate set of interfaces. Note that there is no need for all protocols at all layers to be controllable. Instead, the proposed system can be implemented incrementally. We also highlight that the proposed performance control concept is not limited to those channel adaptation mechanisms, considered in this paper, but can be extended to include MIMO and AMC functionality of the physical layer. In this case appropriate modifications to the proposed system are required.

The core of the proposed system is the change-point detection algorithm that adopted to detect parameter changes in time-varying arrival and channel processes. The current states of both processes are then considered as covariance stationary and used to estimate protocol parameters that provide the best possible performance at the current instant of time. Numerical results demonstrate that the proposed system provide significant performance gain compared to static configuration of protocols at different layers.

We specifically stress that the proposed system does not guarantee that the application receives the quality it requests. Instead, the system tries to achieve the best possible performance at any instant of time. The service it tries to implement is similar to the best-effort service in the wired Internet.

## REFERENCES


[1] V. Srivastava and M. Motani. Cross-layer design: a survey and the road ahead. *IEEE Comp. Comm.*, 43(12):112–119, Dec. 2005.

[2] V. Raisinghani and S. Lyer. Cross-layer design optimizations in wireless protocol stacks. *IEEE Comp. Comm.*, 27:720–724, 2004.

[3] G. Wu, Y. Bai, J. Lai, and A. Ogielski. Interactions between TCP and RLP in wireless Internet. In *Proc. IEEE GLOBECOM'99*, pages 661–666, Rio de Janeiro, Dec. 1999.

[4] P. Sudame and B. Badrinath. On providing support for protocol adaptation in mobile wireless networks. *Mob. Netw. and Appl.*, 6(1):43–55, Jan./Feb. 2001.

[5] B.-J. Kim. A network service providing wireless channel information for adaptive mobile applications. In *Proc. IEEE ICC'01*, pages 1345–1251, Helsinki, June 2001.

[6] K. Chen, S. Shan, and K. Nahrstedt. Cross-layer design for accessibility in mobile ad hoc networks. *Wireless Pers. Comm.*, 21(1):49–76, Apr. 2002.

[7] D. Moltchanov. The structure of the reactive performance control system for wireless channels. In *Proc. NEW2AN'06*, pages 164–176, 2006.

[8] Q. Wang and M. Abu-Rgheff. Cross-layer signalling for next-generation wireless systems. In *Proc. WCNC'03*, pages 1084–1089, March 2003.

[9] V. Kawadia and P.R. Kumar. A cautionary perspective on cross-layer design. *IEEE Pers. Comm.*, 12(1):3–11, Feb. 2005.

[10] Q. Lui, S. Zhou, and G. Ganniakis. Combining adpative modulation and coding with truncated ARQ enhances throughput. In *IEEE Workshop on Signal Processing*, pages 110–114, 2003.

[11] J. Tang and X. Zhang. Capacity analysis for integrated multiuser and antenna diversity over Nakagami-m fading channels in mobile wireless networks. In *Proc. CISS'05*, 16-18 2005.

[12] X. Zhang, J. Tang, H.-H. Chen, S. Ci, and M. Guizani. Cross-layer-based modeling for quality of service guarantees in mobile wireless networks. *IEEE Comm. Mag.*, pages 100–106, Jan. 2006.

[13] M. Zorzi and R. Rao. The effect of correlated errors on the performance of TCP. *IEEE Comm. Let.*, 1(5):127–129, Sep. 1997.

[14] M. Zorzi, A. Chockalingam, and R. Rao. Throughput analysis of TCP on channels with memory. *IEEE JSAC*, 18(7):1289–1300, July 1999.

[15] A. Toledo, X. Wang, and B. Lu. A cross-layer TCP modelling framework for MIMO wireless systems. *IEEE Trans. Wir. Comm.*, 5(4):920–929, Apr. 2006.

[16] Q. Lui, S. Zhou, and G. Ganniakis. TCP performance in wireless access with adaptive modulation and coding. In *Proc. ICC*, pages 3989–3993, 2004.

[17] S.-Q. Li and C.-L. Hwang. Queue response to input correlation functions: discrete spectral analysis. *IEEE Trans. Netw.*, 1:522–533, Oct. 1997.

[18] B. Hajek and L. He. On variations of queue response for inputs with the same mean and autocorrelation function. *IEEE Trans. Netw.*, 6(5):588–598, Oct. 1998.

[19] D. Moltchanov, Y. Koucheryavy, and J. Harju. Loss performance model for wireless channels with autocorrelated arrivals and losses. *Comp. Comm.*, 29(13–14):2646–2660, Aug. 2006.

[20] D. Moltchanov, Y. Koucheryavy, and J. Harju. Cross-layer modeling of wireless channels for IP layer performance evaluation of delay-sensitive applications. *Comp. Comm.*, 29(7):827–841, Apr. 2006.

[21] Y.-Y. Kim and S.-Q. Li. Capturing important statistics of a fading/shadowing channel for network performance analysis. *IEEE JSAC*, 17(5):888–901, May 1999.

[22] D. Moltchanov, Y. Koucheryavy, and J Harju. Simple, accurate and computationally efficient wireless channel modeling algorithm. In *Proc. WWIC*, Xanthi, Greece, May 2005.

[23] S. Khayam, S. Karande, M. Ilyas, and H. Radha. Improving wireless multimedia quality using header detection with priors. In *Proc. IEEE ICC'2006, To appear*, Istanbul, Turkey, June 2006.

[24] A. Konrad, B. Zhao, A. Joseph, and R. Ludwig. Markov-based channel model algorithm for wireless networks. *Wireless Networks*, 9(3):189–199, 2003.

[25] S. Khayam. IEEE 802.11b error traces. Technical report, Michigan State University, Available at: http://www.egr.msu.edu/waves/people/Ali_files/, Accessed: 11.11.2004.
[26] J. Beran, R. Sherman, M. Taqqu, and W. Willinger. Long-range dependence in variable bit rate video traffic. *IEEE Trans. on Comm.*, 5:1566–1579, 1995.
[27] M. Garret and W. Willinger. Analysis, modeling and generation of self similar VBR video traffic. *Comp. Comm. Review*, 24:269–281, 1994.
[28] D. Heyman and T. Lakshman. What are the implications of the long-range dependence for vbr video traffic engineering? *IEEE/ACM Trans. on Netw.*, 4(3):123–134, June 1996.
[29] S. Bates and S. McLaughlin. Is VBR video non-stationary or self-similar? Implications for ATM traffic characterisation. In *Proc. 10th ESM*, pages 27–34, Budapest, Hungary, June 1996.
[30] T. Borsos. On the temporal characteristics of video traffic. In *Proc. LCN*, pages 500–508, Tampa, USA, Nov. 2001.
[31] M. Grasse, M. Frater, and J. Arnold. Testing vbr video traffic for stationarity. *IEEE Trans. Circuits Syst. Video Techn.*, 10(3):448–459, 2000.
[32] MPEG-4 and H.263 video traces for network performance evaluation. Available at: http://www.tkn.tu-berlin.de/research/trace/trace.html, Accessed on: 12.05.2006, Technical University of Berlin.
[33] P. Seeling, M. Reisslein, and B. Kulapala. Network performance evaluation using frame size and quality traces of single-layer and two-layer video: a tutorial. *IEEE Comm. Surveys*, 6(3):58–78, 3rd Quarter 2004.
[34] MPEG traffic archieve. Available at: http://www.info3.informatik.uniwuerzburg.de /mpeg/traces/, Accessed on: 12.09.2003, University of Wuerzburg.
[35] A. Pettitt. A simple cumulative sum type statistic for the change-point problem with zero-one observations. *Biometrika*, 61(1):79–84, 1980.
[36] D. Hinkley. Inference about the change-point from cumulative sum tests. *Biometrika*, 58(3):509–523, 1971.
[37] D. Montgomery. *Introduction to statistical quality control*. John Wiley & Sons, New York, 3rd edition, 1996.
[38] T. Ryan. *Statistical methods for quality improvement*. John Wiley & Sons, New York, 1989.
[39] J. Wieringa. Control charts for monitoring the mean of AR(1) data. Available at: http://www.ub.rug.nl/eldoc/som/a/98a09/98a09.pdf, University of Groningen, Department of Econometrics, Faculty of Economic Sciences, Accessed on 06.07.2005.
[40] J. Wieringa. Statistical process control for serially correlated data, PhD Thesis. Available at: http://dissertations.ub.rug.nl/files/faculties /eco/1999/j.e.wieringa/, University of Groningen, Department of Econometrics, Faculty of Economic Sciences, Accessed on 18.10.2005.
[41] C. Blondia. A discrete-time batch Markovian arrival process as B-ISDN traffic model. *Belgian Journal of Oper. Res.*, 32(3,4):3–23, 1993.
[42] A. Lombardo, G. Morabito, and G. Schembra. An accurate and treatable Markov model of MPEG video traffic. In *Proc. of IEEE INFOCOM*, pages 217–224, 1998.
[43] K. Spaey and C. Blondia. Circulant matching method for multiplexing ATM traffic applied to video sources. Technical report, Department of Mathematics and Computer Science, University of Antwerp, 1998.
[44] D. Moltchanov, Y. Koucheryavy, and J. Harju. The model of single smoothed MPEG traffic source based on the D-BMAP arrival process with limited state space. In *Proc. of ICACT*, pages 55–60, Phoenix Park, R. Korea, January 2003.
[45] D. Moltchanov, Y. Koucheryavy, and J. Harju. Performance response of wireless channels for quantitatively different loss and arrival statistics. *Perf. Eval.*, In Press, Dec. 2007.
[46] A. Dudin and A. Karolik. BMAP/SM/1 queue with markovian input of disasters and non-instantaneous recovery. *Perf. Eval.*, 45(1):19–32, May 2001.
[47] B. Choi, Y. Chung, and A. Dudin. The BMAP/SM/1 retrial queue with controllable operation modes. *Eur. J. of Oper. Res.*, 131(1):16–30, May 2001.
[48] A. Dudin and O. Semenova. A stable algorithm for stationary distribution calculation for a BMAP/SM/1 queueing system with markovian arrival input of disasters. *J. Appl. Probab.*, 41(2):547–556, 2004.
[49] M. Zorzi, R. Rao, and L. Milstein. ARQ error control for fading mobile radio channels. *IEEE Trans. on Veh. Tech.*, 46(2):445–455, May 1997.
[50] M. Zorzi, R. Rao, and L. Milstein. Performance analysis of ARQ Go-Back-N protocol in fading mobile radio channels. In *Proc. Milcom*, pages 576–580, November 1995.
[51] M. Zorzi and R. Rao. Throughput analysis of Go-Back-N ARQ in Markov channels with unreliable feedback. In *Proc. Globecom*, pages 1232–1237, June 1995.
[52] M. Zorzi and R. Rao. ARQ error control for delay-constrained communications on short-range burst-error channels. In *Proc. VTC*, pages 1528–1532, May 1997.
[53] M. Krunz and J.-G. Kim. Fluid analysis of delay and packet discard performance for QoS support in wireless networks. *IEEE JSAC*, 19(2):384–395, February 2001.
[54] A. Fantacci. Queuing analysis of the selective repeat automatic repeat request protocol for wireless packet networks. *IEEE Trans. Veh. Tech.*, 45:258–264, May 1996.
[55] T. Takine, T. Suda, and T. Hasegawa. Cell loss and output process analyses of a finite-buffer discrete-time ATM queuing system with correlated arrivals. *IEEE Trans. on Comm.*, 43(2/3/4):1022–1037, February/March/April 1995.
[56] L. Gemignani. Efficient and stable solutions of structured Hessenberg linear systems arising from difference equations. Technical report, Departments of Mathematics, University of Pisa, 1995.
[57] B. Meine. Fast algorithms for the numerical solutions of structured Markov chains. Technical report, Departments of Mathematics, University of Pisa, 1997.
[58] N. Akar and K. Sohraby. Matrix-geometric solutions of M/G/1 type Markov chains: a unifying generalized state-space approach. Technical report, University of Missouri, Kansas, 1998.

**Dmitri Moltchanov** Dmitri Moltchanov is a Senior Research Scientist in the Institute of Communications Engineering at the Tampere University of Technology, Finland. He received the M.Sc. and Candidate of Science degrees from State University of Telecommunications, St.Petersburg, Russia in 2000 and 2002, respectively, and the PhD degree from Tampere University of Technology, Tampere, Finland in 2006. His research interests include performance evaluation and optimization issues of wireless and wired IP networks, ad hoc and sensor networks and P2P networks. Dmitri Moltchanov serves as a TPC member in a number of international conferences. He authored more than 40 publications.