

Analysis and performance modeling of the packet-level loss process in wireless channels

Eduardo Martínez Graciá
Information Engineering and
Communications Department
University of Murcia
edumart@um.es

Jordi Ortiz Murillo
Information Engineering and
Communications Department
University of Murcia
jordi.ortiz@um.es

Antonio F. Gómez
Skarmeta
Information Engineering and
Communications Department
University of Murcia
skarmeta@um.es

ABSTRACT

Losses in real-time multimedia communications have a big impact on the quality experienced by users. In wireless networks some mechanism of packet-level Forward Error Correction (FEC) is needed to protect the transmission. In order to adjust the configuration of FEC, a model of the packet-level loss process is required. This work proposes a methodology for the analysis of the loss process of packet-oriented communications using semi-Markov models, hierarchical Markov models, and Monte Carlo simulation. The methodology is applied to the analysis of HSDPA channels, and the models obtained are used to verify the performance of FEC implemented with Reed-Solomon codes.

Keywords

Semi-Markov modeling, Hierarchical Markov modeling, Monte Carlo simulation, HSDPA, FEC

1. INTRODUCTION

Losses in packet-based wireless networks cause a strong reduction in the quality of multimedia transmissions. Compression algorithms like H.264 reduce the bit rate of the multimedia source by employing compression techniques that take profit of the similarities between images that are near in the temporal dimension. The loss of information of an image can affect the reconstruction at the decoder side of images that have been compressed with reference to it. The use of key images that reset the list of previous reference images can help to reduce the extension in time of the propagation of errors, but this type of images have a poor compression ratio. As a consequence, protection mechanisms are needed to minimize, as much as possible, the impact of packet losses in differentially encoded images.

Generic packet-level FEC is a method commonly used to protect real-time communications. This type of FEC is implemented using error correcting codes, like Reed-Solomon

(RS) codes. These redundancy codes form blocks of n packets, where k packets are application data and $n - k$ packets are redundant data. RS codes guarantee that the original k application packets can be recovered out of any k packets of the block received. The use of packet-level FEC reduces the amount of bandwidth dedicated to the application data, and increases the end-to-end latency. To control the overhead of FEC, an optimization criteria is needed to decide the (n, k) configuration that maximizes the protection, subject to certain latency and bandwidth restrictions. This optimization needs a mechanism to measure the probability $P(m, n)$ of losing m packets out of a block of n packets. To obtain this measurement, a model of the packet-level loss process is needed. This work defines a methodology of performance modeling that uses a combination of semi-Markov models and hierarchical Markov models. This methodology is developed in the European ITEA2 project CAM4Home [1].

The remainder of the work is organized as follows. In section 2 we present a state of the art of the analysis of the loss process. In section 3 we describe the network traces that we use as a proof of concept of the analysis methodology presented in this work. In section 4 we describe the methodology proposed. In section 5 we present the use of the model obtained to evaluate of the performance of a packet-level FEC using RS codes. At the end, we present the conclusions of our work, and future lines of work.

2. STATE OF THE ART

Starting with the work developed by Shannon, Gilbert [15] defined an error model of a communication channel based on hidden Markov chains. This model has been extensively applied to the characterization of packet losses in many types of networks. In this model, a channel with errors has two states: G (good) and B (bad), which form a Markov chain with transition matrix:

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}$$

The transition diagram between the states is depicted in figure 1. The error probabilities follow a uniform distribution. In state G the error probability is $\varepsilon_1 = 0$; in state B the error probability is ε_2 . The parameters of the model are \mathbf{P} and ε_2 , and their value is approximated according to the observed behaviour of the channel.

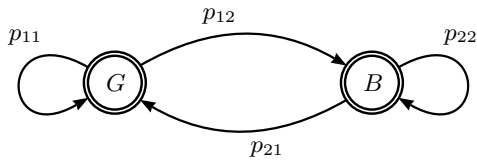


Figure 1: State diagram of Gilbert model

This model has several important limitations that makes it incapable of correctly representing real data. First, the distributions of the wait times in the states are geometric, implying the lack of memory of this random process. Second, the models suppose that the system is a renewal process, that is, the system moves from the G state to the B state without memory. This is not in line with the observed data, that show that the length of the intervals between errors has a memory (it is more probable that a long interval follows another long interval), and the same happens with the bursts of errors. Finally, it is usual that communication systems suffer a low error rate in the G state, instead of having zero probability. Elliott [18] modified Gilbert model to incorporate this last feature. Even if they have limitations, models with two errors have being broadly used due to its simplicity and to the capacity of obtaining analytical expressions of the communication performance.

To obtain a better model for the real loss behaviour of communication channels, Fritchman [6] proposed the use of a Markov model with a finite number of states n , with different loss probabilities. In figure 2 a model of this type with two good states (G_1 and G_2) and two bad states (B_1 and B_2) is shown. In [17] there is an example of use of this model to evaluate the performance of ATM networks.

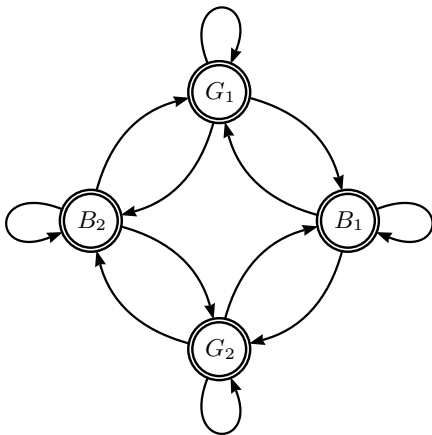


Figure 2: Fritchman model with four states

Although the previous model can obtain a good characterization of the observed data, there is an important number of works ([9], [10], [12], [13], [11], [19]) that use other models for wireless networks that are able to capture with a better fidelity the groupings of observed packet losses. These groupings are usually called clusters, bursts and gaps. Network traces consist of sequences of observations that indicate if there has been a correct transmission (0) or if there has been a loss (1). A burst is a sequence of ones without zeros between. A gap is a sequence of zeros without ones between. A cluster is a sequence of observations that begins with a 1,

ends with another 1, and the distance between internal errors is L or less. In other words, a cluster represents a period during which the communication system has some degree of errors, but admits some correct transmissions. The decision about which is the value of L that identifies correctly a cluster is a heuristic problem. For instance, Konrad uses the expression $L = m_e + sd_e$, where m_e is the average size of bursts, and sd_e is the standard deviation. This criteria permits to segment the trace into substraces that are, more or less, statistically stationary. In this work we propose the use of statistical hypothesis testing as the criteria to segment network traces.

There are several alternatives to analyse the behaviour of the system inside and outside clusters: hierarchical Markov models [7], used in [9], [10], [12] and [11], or Markov models of order greater than one, used in [13]. Markov models of order greater than one employ transition probabilities that depend on the k previous states, being $k > 1$ the order of the Markov chain. The complexity of this type of models grows exponentially as a function of k , and it can be shown that any Markovian model of order k is equivalent to a Markovian model of order one [5].

Hierarchical models with two levels (the most common case) characterize the system with a first-level Markov chain whose states are described by second-level Markov chains. In [12] and [11] a simple hierarchical model is used in which the first-level Markov chain has two states: good and bad. Inside the good state, a Markov chain describes the alternating generation of gaps and bursts. The same occurs with the bad state, but obviously with different error rates state durations (first and second level chains can be semi-Markov chains). In [9] the authors model different types of clusters using a similar technique to the one used in voice analysis. Each type of cluster (characterized with a length and a maximum error rate) is modelled with a lineal Markov chain (the execution of the chain moves from an initial state to a final state, passing through a line of states), in which the bell form of the cluster is captured. These cluster lineal Markov chains are linked in a high level chain in which, from the initial state, there is a transition to the first state of each cluster chain with a certain probability.

Markovian models have been applied to the analysis of the loss process at the physical level [9], [17], at the link level [11], and at the packet level [10], [12], [11], [19], [16]. Although hierarchical models are interesting, it is complex to obtain from them simple analytical expressions of some important performance parameters. One of them is $P(m, n)$ that measures the probability of having m errors in a block of n consecutive packets. This measurement is fundamental to configure the FEC mechanism. In the works [11] and [10], dedicated to the analysis of UMTS networks, there is no strong statistical argument to verify the use of a Fritchman model. This type of model is equivalent to a semi-Markov model in some cases, and as it is shown in this paper, the characteristics of the network traces (long gaps) calls for the use of this equivalence. On the other hand, it is not common to find works using the Baum-Welch algorithm [5] to extract the model parameters from the traces. On the contrary, it is common to use the method of moments to obtain approximations of the parameters.

3. NETWORK TRACES

In this section it is shown the process that has been carried out to acquire the network traces that will be used to exemplify the analysis methodology proposed in this work. The traces come from a UDP communication using the downlink channel of a commercial UMTS network that implements the HSDPA technology. In subsection 3.1 there is a brief description of the UMTS networks and HSDPA. In subsection 3.2 the setup of the scenario to generate the network traces is described.

3.1 UMTS and HSDPA

The Universal Mobile Telecommunication System (UMTS) [3], [4] is a third generation cellular network based on the Wideband Code Division Multiple Access (WCDMA) technology in the radio interface. It uses a modulation technique with a large frequency spectrum. The base station distributes the information to each receptor using a specific dispersion signal. The receptor uses the same dispersion signal and frequency filters to recover the information transmitted to it. This technique, in combination with dynamic transmission power control, permits an increment in the effective bandwidth available in comparison with previous mobile network technologies.

Figure 3 shows a typical communication scenario. UMTS architecture is made up of two networks: the UMTS Terrestrial Radio Access Network (UTRAN), consisting of the Base Stations (BS) and the Radio Network Controllers (RNC), that manage groups of base stations. Equipments able to communicate with BSs are called User Equipments (UE). The UTRAN is connected to other packet networks, as Internet, through the UMTS Core Network (CN). This network consists of Serving GPRS Support Nodes (SGSN) and the Gateway GPRS Support Nodes (GGSN). SGSNs are in charge of supporting the packet based communication to the access network, and to manage the user mobility, whereas GGSNs maintain the connections to other types of networks, extern of the UMTS system.

High Speed Downlink Packet Access (HSDPA) could be considered an extension and improvement of UMTS, that permits to establish connections with bit rates of 1.8 Mbps, 3.6 Mbps, 7.2 Mbps and 14.4 Mbps in the downlink (from BS to UE), depending on the signal quality and the occupancy of the cell. The HSDPA channels are shared channels that use time division. HSDPA substitutes some of the mechanisms of WCDMA, as the power transmission control, with others as a packet scheduling that gives more time slots to those UEs with better signal reception. This means that the performance of the HSDPA can have strong fluctuations in time. Although a complete model of this channels should include the signal quality perceived by the UE and the occupancy of the cells, this information is not available to a common user of the network, but only to the network provider, so it will not be taken into account. The models develop in this work will reflect the view that end systems have of the loss process.

3.2 Network trace capture

Figure 3 shows the scenario employed to obtain network traces. A computer connected to the network of the University of Murcia sends packets with a constant rate to a second

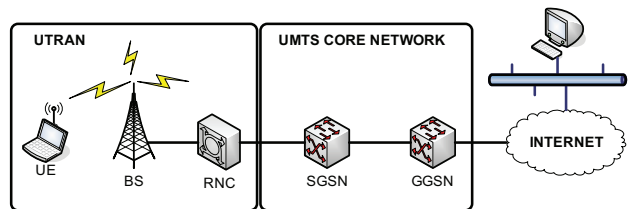


Figure 3: Scenario used to generate traces.

computer connected with a HSDPA link of the Spanish mobile operator Movistar. The UE employs a Huawei E220 modem. To study the behaviour of the HSDPA link with maximum use, we have used the `iperf` [2] tool to measure the available bandwidth. After several tests, we have verified that the link has a stable down rate of 1 Mbps. Knowing the available bandwidth, we have send a UDP flow with a constant rate of 1 Mbps during periods of 8 hours. The objective of this phase is to obtain empirical observations about the losses experienced in the network. The transmitter calculates the period between packets with the formula:

$$t = \frac{\text{payload} \times 8 \times 10^6}{\text{bandwidth}} \quad (1)$$

In our experiments we use $\text{payload} = 1472$ and $\text{bandwidth} = 1024 \times 10^3$ bits per second, so that $t = 11500$ microseconds. Each packet sent includes a sequence number that helps to detect packet losses. The receptor generates a file with a row for each packet received. The information recorded per packet includes the sequence number, the reception timestamp (expressed with the fields of `timeval`) and the size of the payload. Once the network trace is finished, packets are reordered and a new compact trace is generated. The new trace contains an alternate sequence of the gap lengths and burst lengths detected.

3.3 Characteristics of the traces

Figure 4 shows three selected traces captured between 12:00 and 20:00 in three different days. Figures show the cumulative probabilities of gaps and bursts of the three traces. It is evident that the network behaviour was quite different in the three observations. Table 1 shows a summary of the most important values of the traces. It can be seen that the packet loss rate \hat{P}_e is very important in the three cases. Additionally, the standard deviation is quite important in the subtraces of gaps and bursts. The first trace has big gaps and also big bursts. The second trace corresponds to a situation of big gaps, but not so big bursts. Finally, the third trace has relatively short bursts, and also short gaps, most of them of size 1 (90%), but also an important number around 10 packets.

Instead of using directly the captured traces, a removal of extreme data has been carried out to guarantee a more coherent statistical analysis. The filter used in the gaps is:

$$|g(i) - \bar{g}| \leq \alpha \times \sigma_g$$

where $g(i)$ is the i th sample of the gaps subtrace, \bar{g} is the average value of the subtrace and σ_g is the standard deviation

Table 1: Network traces characteristics

Trace	Packets	Received	Lost	\bar{P}_e	Gaps				Bursts			
					average	std	min	max	average	std	min	max
T1	2507862	2482508	25354	0.010	1393.1	10414	1	229962	14.23	392.83	1	16509
T2	2504285	2498771	5514	0.002	3574.8	12159	1	117895	7.89	45.86	1	717
T3	2373206	2340506	32700	0.014	842.21	978.32	1	8209	11.77	350.55	1	13804

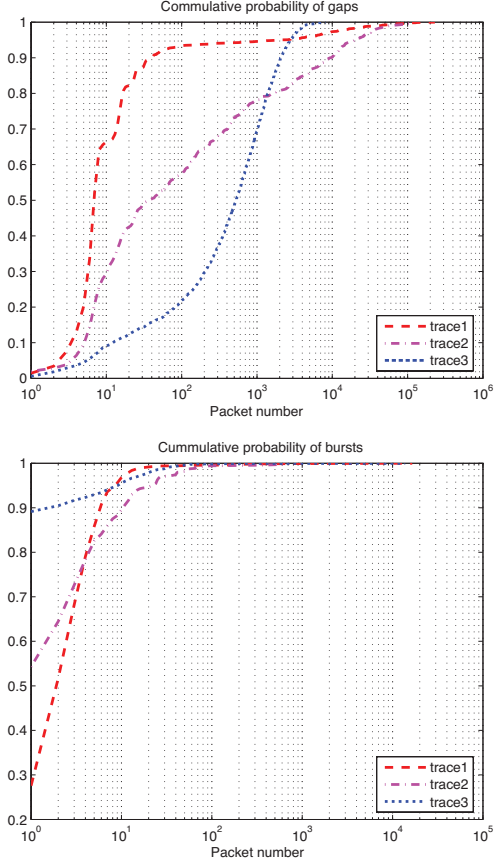


Figure 4: Cumulative probabilities of gaps and bursts

of the gaps subtrace. The parameter α permits to adjust the limit to cut extreme values. In this work we have used $\alpha = 3$. A similar filter has been used in the bursts subtrace.

4. METHODOLOGY

In this section we describe the methodology used to generate a model of the packet-level loss process, starting with network traces similar to the one described in the previous section. In subsection 4.1 a theoretical result used in the methodology is introduced. In subsection 4.2 the first phase of the methodology is exposed, consisting in the verification of applicability of a semi-Markov model to the network trace. In subsection 4.3 the second phase of the methodology is described, consisting in the adjustment of the state duration probabilities. Finally, in subsection 4.4 the third phase is proposed, in which a hierarchical model is built if

needed.

4.1 Markov chains and semi-Markov chains

Markov chains and semi-Markov chains have interesting features to model the packet-level loss process. On the one hand, Markov chains permit to obtain simple analytical expressions to compute performance measurements as $P(m, n)$. On the other hand, it is easier to adjust semi-Markov chains to experimental data. The most commonly used method to adjust Markov chain parameters is the Baum-Welch algorithm [5]. It is an iterative method based on the concept of maximum likelihood. Be τ the set of parameters of a model and \mathbf{o} a sequence of observations. The Baum-Welch algorithm searches the set τ that maximizes the probability $Pr(\mathbf{o} | \tau)$. In our case, the chain of observations have a quite considerable length (8 hours per trace, with nearly 87 packets per second). The amount of calculations needed to adjust a Markov chain with this type of trace is prohibitive, as each 0 or 1 observation calls for a multiplication of the transition matrix. An alternative is needed. When compact traces are used, it is easier to adjust the probability distributions of the lengths of gaps and bursts, and this can be applied to the parameterization of semi-Markov chains. Additionally, it is easier to simulate semi-Markov chains. Unfortunately, the disadvantage of semi-Markov chains is the difficulty of obtaining an analytical expression of $P(m, n)$.

Luckily, there is a theoretical result that permits to establish an equivalency between a certain type of semi-Markov chains and Markov chains. Two models λ and μ are equivalent if, for any sequence of observations \mathbf{o} , the probability of occurrence of that sequence is the same in both models:

$$Pr(\mathbf{o} | \lambda) = Pr(\mathbf{o} | \mu)$$

Let us suppose that the probability of state durations $w_i(l)$ in a semi-Markov chain is polygeometric, that is:

$$w_i(l) = \sum_{j=1}^{J_i} a_j (1 - q_j) q_j^{l-1} \quad (2)$$

where $\sum_{j=1}^{J_i} a_j = 1$. This can be expressed in matrix form as:

$$w_i(l) = \mathbf{a}_i \mathbf{Q}_i^{l-1} (\mathbf{I} - \mathbf{Q}_i) \mathbf{1}$$

where $\mathbf{a}_i = [a_1 \ a_2 \ \dots \ a_{J_i}]$ and $\mathbf{Q}_i = \text{diag}\{q_i\}$ have non null elements q_i in the diagonal. Let us suppose that the transition matrix of the Markov chain embedded in the semi-Markov process is:

$$\hat{\mathbf{P}} = [\hat{p}_{ij}]_{s,s}$$

It can be shown [5] that there exists an equivalent Markov chain whose transition matrix is:

$$\begin{bmatrix} \mathbf{Q}_1 & \hat{p}_{12}(\mathbf{I} - \mathbf{Q}_1)\mathbf{1a}_2 & \dots & \hat{p}_{1s}(\mathbf{I} - \mathbf{Q}_1)\mathbf{1a}_s \\ \hat{p}_{21}(\mathbf{I} - \mathbf{Q}_2)\mathbf{1a}_1 & \mathbf{Q}_2 & \dots & \hat{p}_{2s}(\mathbf{I} - \mathbf{Q}_2)\mathbf{1a}_s \\ \dots & \dots & \dots & \dots \\ \hat{p}_{s1}(\mathbf{I} - \mathbf{Q}_s)\mathbf{1a}_1 & \hat{p}_{s2}(\mathbf{I} - \mathbf{Q}_s)\mathbf{1a}_2 & \dots & \mathbf{Q}_s \end{bmatrix}$$

As a consequence, if we are able to model the system traces with a semi-Markov process with polygeometric state durations, we can obtain an equivalent Markov chain to evaluate the probability $P(m, n)$ or other performance measurement. Additionally, it is interesting to observe that, if the semi-Markov process has a transition matrix like this:

$$\hat{\mathbf{P}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (3)$$

then the equivalent Markov chain is a Fritchman model with two groups of states (see figure 2). The amount of state types is determined by the order of the polygeometric distribution of gaps and bursts in equation (2). Another interesting characteristic of this model is that the error probability is 0 in the good states, and 1 in the bad states, so the parameter evaluation is simplified. In the rest of the paper we employ semi-Markov models with matrix (3).

4.2 Applicability of semi-Markov models

Before modeling the compact traces with semi-Markov processes, we must verify that the traces can be modelled with this tool using a χ^2 statistical hypothesis testing. Let us suppose that we have a trace $l_0, l_1, l_2, \dots, l_n$ with alternate gaps and bursts, as described in subsection 3.2. The likelihood function of a semi-Markov model with a set of parameters τ and transition matrix (3) is:

$$L(\tau) = p_{s_0} w_{s_0}(l_0; \tau) \prod_{i=1}^n w_{s_i}(l_i; \tau)$$

The first state s_0 corresponds to gaps if the trace begins with a gap. Probability distributions $w_{s_i}(l_i; \tau)$ establish the probability of moving from state s_i to the complementary state (from gap to burst or vice versa) in l_i packets. We consider only autonomous semi-Markov processes, that is, these probabilities depend only on the current state. These probability distributions must be adjusted with experimental data. The adjustment must maximize the likelihood function, which can be expressed in this other way:

$$L(\tau) = p_{s_0} w_{s_0}(l_0; \tau) \prod_l w_1(l; \tau)^{n(1,2,l)} \prod_l w_2(l; \tau)^{n(2,1,l)}$$

where $n(1, 2, l)$ is the amount of transitions from state 1 to state 2 in l steps observed in the trace, and $n(2, 1, l)$ is the corresponding amount from state 2 to state 1 in l steps. The best adjustment for this expression is obtained with the τ that maximizes the function. The logarithmic likelihood function permits to express this fit easier:

$$\ln L(\tau) = J_0(\tau) + J_1(\tau)$$

where:

$$\begin{aligned} J_0(\tau) &= \ln p_{s_0} w_{s_0}(l_0; \tau) \\ J_1(\tau) &= \sum_l n(1, 2, l) \ln w_1(l; \tau) + \sum_l n(2, 1, l) \ln w_2(l; \tau) \end{aligned}$$

Taking into account that $J_0/J_1 = 0$ when $n \rightarrow \infty$, we can ignore J_0 . We do not make any assumption about the form of the probability distribution of transitions to verify that the traces can be modelled by a semi-Markov chain, so every value of $w_1(l)$ and $w_2(l)$ for any l is considered a parameter of the fit. The maximum likelihood approximation of these values are the experimental frequencies [5]:

$$\begin{aligned} \hat{w}_1(l) &= \frac{n(1, 2, l)}{N(1)} \\ \hat{w}_2(l) &= \frac{n(2, 1, l)}{N(2)} \end{aligned}$$

where $N(1) = \sum_l n(1, 2, l)$ and $N(2) = \sum_l n(2, 1, l)$.

Considering that l belongs to the interval $[1, \infty)$, the computation of the frequencies could require a large amount of memory. Additionally, the value of $n(\xi, \eta, l)$ with $\xi, \eta \in [1, 2]$ and $\xi \neq \eta$ should not be very low to have a solid test. For this reason, it is usual to divide the domain of l in a finite quantity $u(\xi, \eta)$ of subintervals for a pair of states $\xi \neq \eta$:

$$[1, \infty) = \bigcup_{m=1}^{u(\xi, \eta)} [x_m(\xi, \eta), x_{m+1}(\xi, \eta)] \text{ for each } \xi \rightarrow \eta$$

With this grouped data we get to this equivalent expression of the approximation:

$$\hat{W}_\xi(m) = \sum_{l=x_m}^{x_{m+1}-1} \hat{w}_\xi(l)$$

Now we can perform the test of the hypothesis H_0 that the data can be modelled with an autonomous semi-Markov process. Let it be $N(\xi, \eta, \zeta, m, k)$ the amount of transitions $\xi \rightarrow \eta \rightarrow \zeta$ that are observed in the trace with waiting times in the interval m and k . Let it be:

$$N(\xi, \eta, m) = \sum_{\zeta, k} N(\xi, \eta, \zeta, m, k)$$

The χ^2 criteria to verify H_0 is:

$$\chi^2 = \sum_{\xi, \eta, \zeta, m, k} \frac{[N(\xi, \eta, \zeta, m, k) - N(\xi, \eta, m)\hat{W}_\zeta(k)]^2}{N(\xi, \eta, m)\hat{W}_\zeta(k)}$$

The degrees of freedom of this distribution is determined by three values: N_t the number of possible transitions $\xi \rightarrow \eta \rightarrow \zeta$, N_s the number of possible transitions $\xi \rightarrow \eta$ and N_p the number of parameters of the model if we do not assume any particular form of the distribution probabilities of state durations. It can be shown [5] that the degrees of freedom of this distribution is:

$$f = N_t - N_s - N_p$$

where $N_t = 2u(1,2)u(2,1)$, $N_s = u(1,2) + u(2,1)$ and $N_p = u(1,2) + u(2,1) - 2$. Using a significance level $1 - \alpha = 0.99$ we obtain the results shown in table 2. We use the greatest number of possible equiprobable subintervals with 5 values at least.

Table 2: $\chi_{0.99}^2$ semi-Markov hypothesis test

Trace	$u(1,2)$	$u(2,1)$	f	χ^2	$\chi_{0.99,f}^2$
T1	64	19	2268	6584.89	2427.61
T2	96	22	3990	4307.68	4200.75
T3	489	31	29280	16630.13	29845.98

Only the third trace verifies the hypothesis. The second one is lightly above the verification value, whereas the first one is quite distant. Taking a look at figure 4 it can be seen that the first and second traces have a segment with a highly non stationary behaviour at the beginning. After it, curves adopt a more smooth form. The test fails because it is not possible to fit in a unique semi-Markov model the complete trace. To solve this problem, we propose to divide the complete trace T in two subtraces, TG that corresponds to the good state of the system, where gaps are big, and TB that corresponds to the bad state of the system, where gaps are shorter. It is interesting to note that the proposed division is equivalent to splitting the trace in one subtrace corresponding to clusters (TB) and another subtrace with the observations out of the clusters (TG). We propose to identify clusters with a value L (which determines the maximum distance between errors inside the cluster) equal to the lower integer that generates a subtrace TG that passes the semi-Markov hypothesis test. To avoid a resulting TB trace with very few values, we limit the minimum value of L to 10. With this strategy, we guarantee that the good state can be modelled taking advantage of semi-Markov models. Regarding the bad state, we must employ a semi-Markov model with caution, using a probability distribution more flexible than the polygeometric one, as we will describe after. Table 3 shows the results of the semi-Markov hypothesis test using the χ^2 criteria with the subtrace TG and the value of L that permits to pass that test.

Table 3: $\chi_{0.99}^2$ semi-Markov hypothesis test of TG

Trace	L	$u(1,2)$	$u(2,1)$	f	χ^2	$\chi_{0.99,f}^2$
T1	23	44	10	774	839.03	868.46
T2	10	86	14	2210	2174.16	2367.60

To conclude this subsection, we resume the proposed first phase of the analysis methodology: if a trace passes the semi-Markov hypothesis test, a unique model of the trace is generated; if the test is not verified, the trace is split in two subtraces, and a model is generated for each of them, having the guarantee that a semi-Markov model is applicable to the good subtrace (out of clusters).

4.3 Fit of state duration distributions

In the previous subsection, no particular form for the state duration distributions $w_i(l)$ was proposed. In this section, we show how to perform the fit of these distributions. As it is explained in subsection 4.1, it is interesting to model

these distributions with polygeometric functions (2), as this enables the equivalency of the semi-Markov chain with a Markov chain. In the previous subsection it is shown that, in some cases, the polygeometric fit is not applicable to the complete set of data. In this case, the subtrace corresponding to clusters must be treated in a special way. In this work we propose the use of a Weibull fit in this case.

4.3.1 Polygeometric fit

Parameters of polygeometric distributions (2), also called phase-type distributions, can be adjusted from the observed data by means of the Expectation-Maximization (EM) algorithm. In this work we consider semi-Markov models that have only two states, one corresponding to gaps and the other to bursts. Each polygeometric distribution have the form:

$$w(l, \tau) = \sum_{i=1}^u a_i (1 - q_i) q_i^{l-1} \quad (4)$$

that is, it is a weighted sum of u geometric distributions. We can write the previous equation in matrix form:

$$w(l, \tau) = \mathbf{a} \mathbf{Q}^{l-1} \mathbf{b} \quad (5)$$

where $\mathbf{a} = [a_1 \ a_2 \ \dots \ a_u]$, $\mathbf{Q} = \text{diag}\{q_i\}$ and $\mathbf{b} = [(1 - q_1) \ (1 - q_2) \ \dots \ (1 - q_u)]^t$. The parameters included in τ are a_i and q_i . The EM algorithm is an iterative algorithm that searches the set of parameters τ that maximizes the previous expression. Each step of the algorithm updates the set τ with a new version of the parameters. It can be shown [5] that the maximum likelihood adjustment can be performed with the EM algorithm using the following re-evaluation expressions of the parameters, from step p to step $p + 1$:

$$\begin{aligned} a_{i,p+1} &= \frac{a_{i,p}(1 - q_{i,p})}{T} \sum_{m=1}^T \frac{q_{i,p}^{l_m-1}}{w(l_m, \tau_p)} \\ q_{i,p+1} &= 1 - \frac{\sum_{m=1}^T \frac{q_{i,p}^{l_m-1}}{w(l_m, \tau_p)}}{\sum_{m=1}^T \frac{l_m q_{i,p}^{l_m-1}}{w(l_m, \tau_p)}} \end{aligned}$$

where l_m are the values of the lengths of gaps or bursts. These expressions can be easily implemented, but an important concern must be taken into account: the \mathbf{Q}^l values go quickly to 0, as the values of l are very big (see section 3.3). Consequently, it is necessary to perform a scaling to avoid this. We use these auxiliary expressions:

$$\alpha(l, \tau) = \alpha(l - 1, \tau) \mathbf{Q} \quad (6)$$

$$\bar{\alpha}(l, \tau) = \frac{\alpha(l, \tau)}{\alpha(l, \tau) \mathbf{1}} = \frac{1}{\mu_l} \alpha(l, \tau) \quad (7)$$

where $\alpha(1, \tau) = \mathbf{a}$. The algorithm can work directly with the expressions $\bar{\alpha}(l, \tau)$, as it is easy to show that:

$$\bar{\alpha}(l, \tau) = \frac{\bar{\alpha}(l - 1, \tau) \mathbf{Q}}{\bar{\alpha}(l - 1, \tau) \mathbf{Q} \mathbf{1}}$$

The computation of $\bar{\alpha}(l, \tau)$ has as subproduct the evaluation of $w(l, \tau)$, because:

$$w(l, \tau) = \mu_i \bar{\alpha}(l, \tau) \mathbf{b}$$

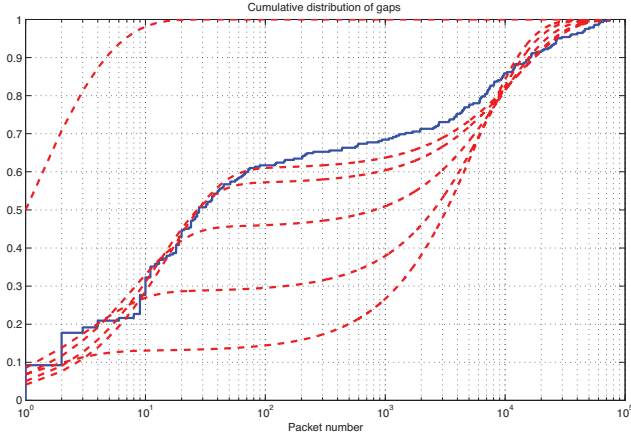


Figure 5: Fit of polygeometric distribution.

It is interesting to observe that the normalization factors μ_i are cancelled in the numerators and denominators of the re-evaluation expressions (6) and (7). The algorithm has been implemented in MATLAB. The initial fit of the parameters τ_0 can have some influence on the speed of convergence of the algorithm. We have verified that a_i can have identical values, giving the same initial importance to all the geometric distributions. We have use u equidistant values for q_i in the $(0, 1)$ interval. Regarding the degree of the distribution, we have verified that $u = 3$ is enough to have a good fit of our experimental traces in 40 iterations of the algorithm. A more exhaustive study of the value of u and the amount of iterations will require a χ^2 test for each distribution $w_i(l, \tau)$ adjustment. In figure 5 we can see the progress of the polygeometric curve fits of the gaps of subtrace TG of trace 1. The cumulative empirical distribution is shown in blue. The first fit corresponds to the initial parameter assignment $\mathbf{a} = [1/3 \ 1/3 \ 1/3]$ and $\mathbf{q} = [1/4 \ 2/4 \ 3/4]$. After five iterations, the algorithm gets the values $\mathbf{a} = [0.0054 \ 0.6026 \ 0.3920]$ and $\mathbf{q} = [0.1652 \ 0.9394 \ 0.9999]$.

4.3.2 Weibull fit

Weibull distribution is a continuous probability distribution with probability density function:

$$f(x; k, \lambda) = \frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}$$

and cumulative distributed function:

$$F(x; k, \lambda) = 1 - e^{-(x/\lambda)^k}$$

k is the parameter that determines the form of the probability curve, and λ determines the scale. When $k = 1$, the distribution is equivalent to an exponential distribution. Due to the ability of this function to fit empirical distributions, it is very useful to model failure processes and waiting times. Figure 6 shows the fit of gaps for subtrace TB of trace 1, obtained with MATLAB.

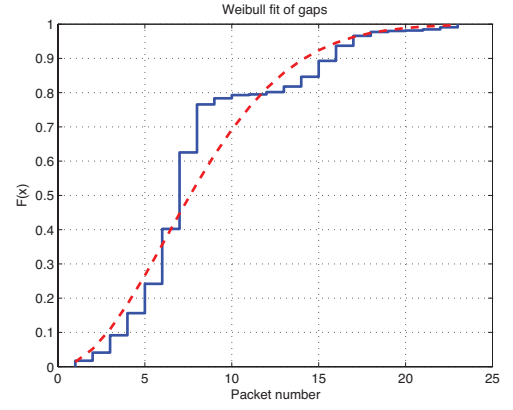


Figure 6: Weibull fit of gaps.

4.4 Markov models of the network traces

This section concludes the methodology proposed in this work. It is obvious that we are not interested specifically in the numerical results, but in the complete analysis process, from the generation of network traces to the production of final models.

Table 4: Fit of trace T3

$w_i(l)$	a_1	a_2	a_3	q_1	q_2	q_3
Gaps	0.0090	0.1145	0.8765	0.9035	0.9073	0.9988
Bursts	0.3924	0.4973	0.1103	0.0009	0.0113	0.9257

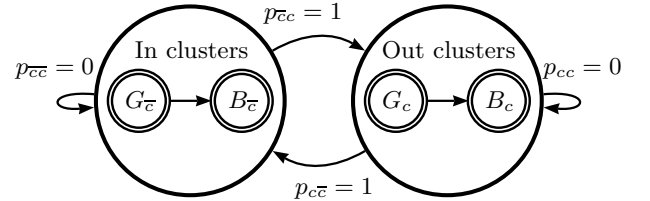


Figure 7: Hierarchical model of T1 and T2.

The analysis made in subsection 4.2 shows that trace T3 can be modelled completely with a semi-Markov process. The values of the parameters are shown in table 4. With this information, we can build the matrix \mathbf{P}_{T3} (3) of the equivalent Markov model, according to the equivalency presented in 4.1:

$$\begin{bmatrix} 0.9035 & 0 & 0 & 0.0378 & 0.0480 & 0.0106 \\ 0 & 0.9073 & 0 & 0.0364 & 0.0461 & 0.0102 \\ 0 & 0 & 0.9988 & 0.0005 & 0.0006 & 0.0001 \\ 0.0090 & 0.1144 & 0.8757 & 0.0009 & 0 & 0 \\ 0.0089 & 0.1132 & 0.8666 & 0 & 0.0113 & 0 \\ 0.0007 & 0.0085 & 0.0652 & 0 & 0 & 0.9257 \end{bmatrix}$$

With this matrix we can generate the stationary probability vector of the Markov chain:

$$\mathbf{p}_{T3} = [0.0001 \ 0.0016 \ 0.9952 \ 0.0005 \ 0.0007 \ 0.0019]$$

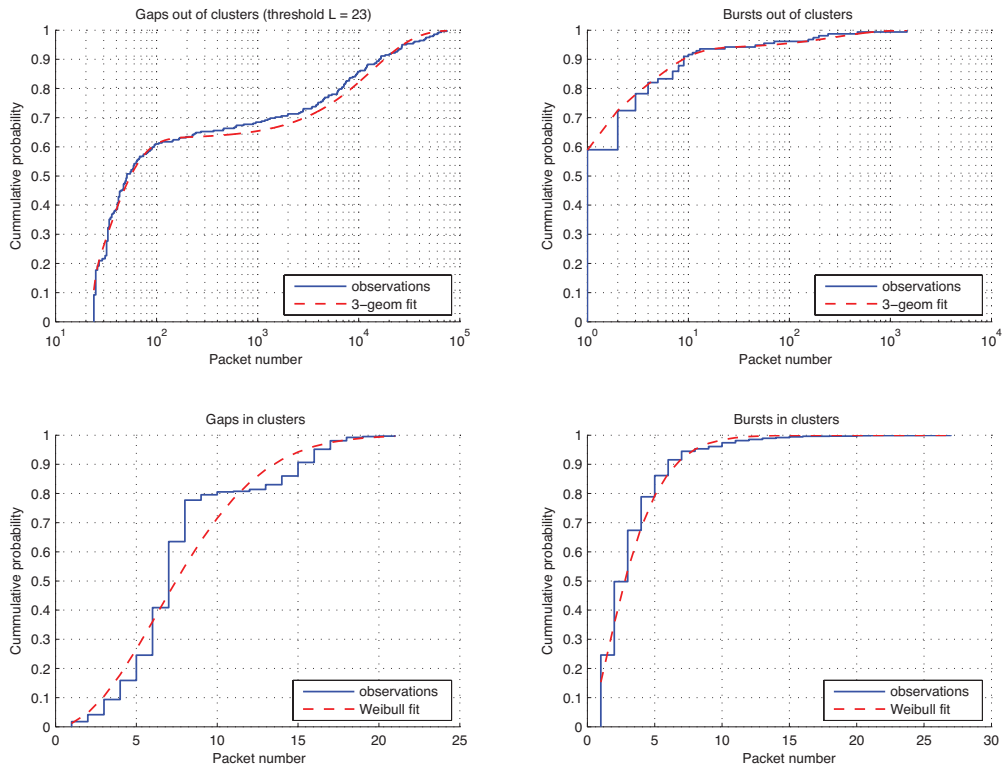


Figure 8: Fit of trace T1

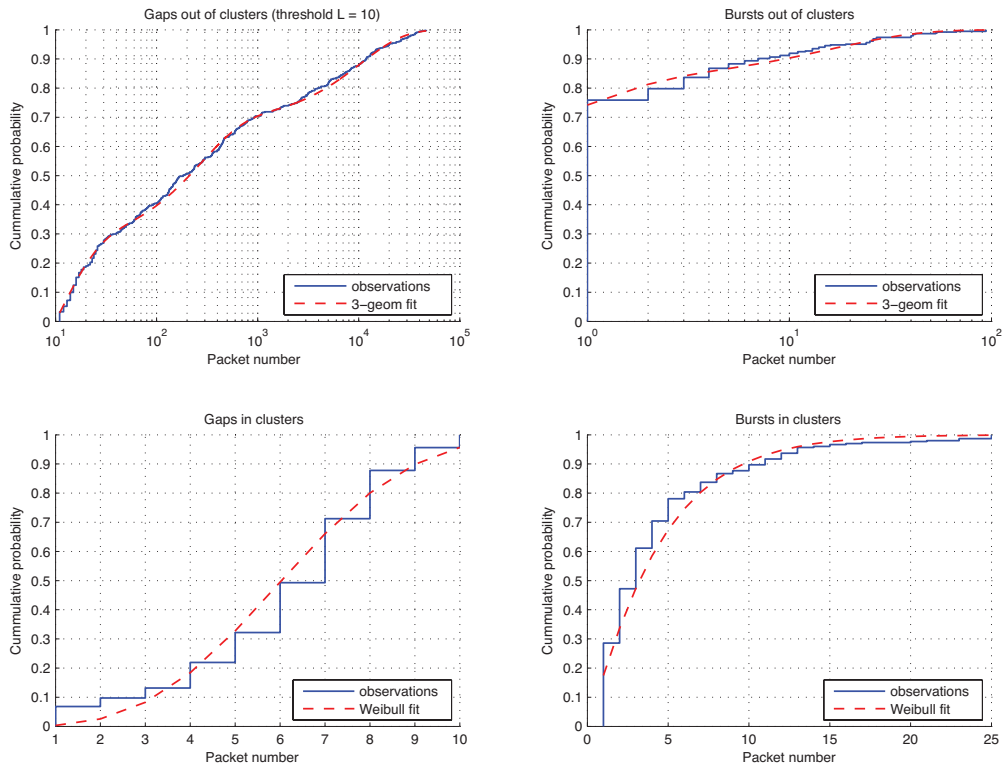


Figure 9: Fit of trace T2

Table 5: Parameter adjustment of trace T1

Distributions out of clusters	a_1	a_2	a_3	q_1	q_2	q_3
$w_{\tau_1}(l)$ (gaps)	0.1375	0.4910	0.3715	0.3565	0.9597	0.9999
$w_{\tau_2}(l)$ (bursts)	0.6210	0.3158	0.0631	0.1585	0.7897	0.9966
Distributions in clusters	k	λ				
$w_{c_1}(l)$ (gaps)	2.0238	8.9321				
$w_{c_2}(l)$ (bursts)	1.3971	3.6159				

Traces T1 and T2 have four adjustments, two for the gaps and burst out of clusters, and two for gaps and burst in clusters. The first two adjustments are made with tri-geometric distributions, while the others are made with Weibull distributions. Figures 8 and 9 show the curves of the fits for traces T1 and T2, and table 5 show the values of the parameters for T1 after the fit. At the end we get a two-level hierarchical model, shown in figure 7.

5. PERFORMANCE OF FEC CODES

This section exemplifies how to use the models obtained in the previous section. We analyse the performance of a generic redundancy code at the packet level. Given k application data packets, the sender adds $n - k$ redundant packets using Reed-Solomon codes [8]. If k out of the n are received, the original block of application packets can be fully recovered. The probability of recovering the information (a measurement of the FEC performance) is given by the probability of having less than $n - k + 1$ packet losses in a block of n packets:

$$P_{FEC} = \sum_{m=0}^{n-k} P(m, n)$$

Obtaining an analytical expression of $P(m, n)$, which gives the probability of having m errors in a block of n consecutive packets, becomes relatively easy when we have a Markov model of the system, as happens with trace T3. Let \mathbf{P} be the transition matrix, and \mathbf{p} the stationary probability vector. Let $\mathbf{F}(1)$ be the diagonal matrix with the probabilities of having a packet loss in each state, and $\mathbf{F}(0) = \mathbf{I} - \mathbf{F}(1)$. In trace T3 we have:

$$\mathbf{F}(1) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

We define $\mathbf{P}(0) = \mathbf{P}\mathbf{F}(0)$ and $\mathbf{P}(1) = \mathbf{P}\mathbf{F}(1)$ as the probability matrix whose (i, j) element has the probability of doing a transition $i \rightarrow j$ without and with error, respectively. By definition, we have $\mathbf{P}(0, 1) = \mathbf{P}(0)$ and $\mathbf{P}(1, 1) = \mathbf{P}(1)$. Recursively, we can define $\mathbf{P}(m, n)$:

$$\mathbf{P}(m, n) = \mathbf{P}(m, n-1)\mathbf{P}(0, 1) + \mathbf{P}(m-1, n-1)\mathbf{P}(1, 1) \quad (8)$$

The scalar probability is obtained using \mathbf{p} , with the expression $P(m, n) = \mathbf{p}\mathbf{P}(m, n)\mathbf{1}$. In [5] there is an algorithm to obtain P_{FEC} efficiently using the discrete Fourier transform

of (8). Figure 12 shows the results obtained for trace T3 and several configurations of n .

Hierarchical models of traces T1 and T2 do not help to obtain analytical expressions of $P(m, n)$. In this case, it is easier to find a measurement of the FEC performance using the Monte Carlo method [14]. In our case, the probability of having m errors in a block of n packets can be estimated with $\hat{P}(m, n) = v_m/N$ where v_m is the amount of blocks of length n of the data obtained with the simulation that have m packet losses, and $N \times n$ is the total amount of simulated packets. We have generated about 10^7 packet transmissions with MATLAB, representing more than 320 hours of transmission. Figures 10 and 11 show P_{FEC} for the same values of n used for trace T3.

6. CONCLUSIONS

This work proposes a methodology for the analysis and performance modeling of the packet-level loss process. The objective of the methodology is the construction of stochastic models based on the use of semi-Markov processes and hierarchical processes, adequate to evaluate performance parameters of the communication system. The methodology is verified in the context of a FEC performance analysis in HSDPA links, giving useful information about the packet-level loss process. This methodology is general enough to be used in other applications, as could be the evaluation of a control flow protocol or the quality of a video transmission. The methodology is based on solid statistical tools, as the χ^2 semi-Markov hypothesis testing.

There are many open issues to tackle. The methodology can be extended to use a statistical test of the goodness of the polygeometric fit. This test could permit to select the degree of the polygeometric distributions, and the amount of iterations of the EM algorithm. Another interesting extension of this work is the study of how many observed data is needed to have a consistent model of the system. The methodology can be evaluated in different types of networks (801.11, WiMax, etc). Finally, additional information of network traces can be incorporated to the modeling phase, as it could be the case of the packet timestamps. Interactive multimedia applications have a temporal restriction that limits the latency between the capture and rendering of data to a maximum of 150 milliseconds. In this type of applications, a late packet can be treated as a lost packet.

7. REFERENCES

- [1] Cam4home. <http://www.cam4home-itea.org/>.
- [2] Iperf. <http://dast.nlanr.net/projects/iperf>.
- [3] *UMTS Networks. Architecture, Mobility and Services*. John Wiley and Sons. ISBN 0-471-48654-X, 2001.

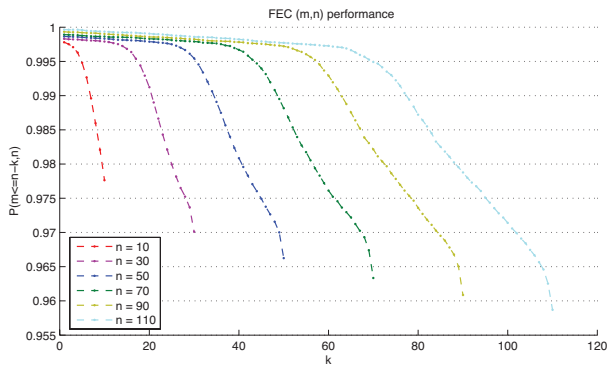


Figure 10: FEC recover probability for trace T1

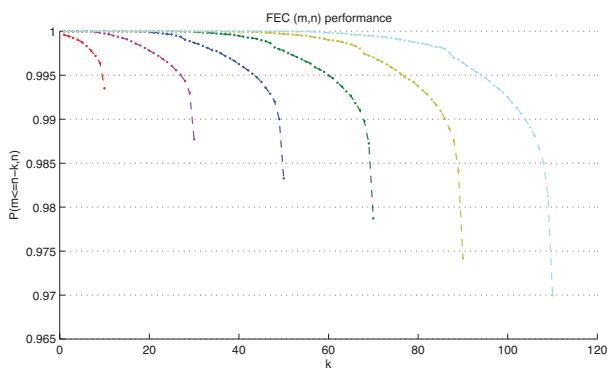


Figure 11: FEC recover probability for trace T2

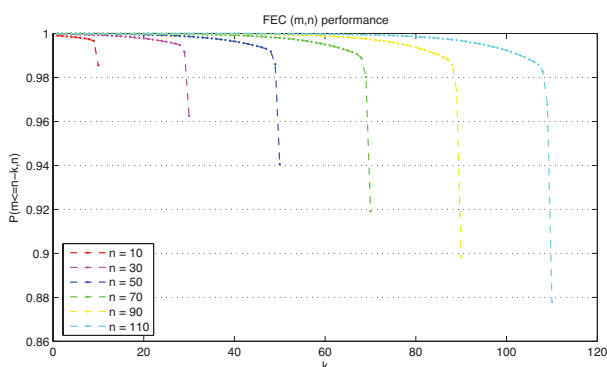


Figure 12: FEC recover probability for trace T3

- [4] *UMTS The Fundamentals*. John Wiley and Sons. ISBN 0-470-8455-7, 2003.
- [5] *Performance Analysis and Modeling of Digital Transmission Systems*. Information Technology Series. Kluwer. ISBN 0-306-48191-X, 2004.
- [6] F. B. D. A binary channel characterization using partitioned markov chains. *IEEE Trans. Inform. Theory*, 13:221–227, 1967.
- [7] T. N. Fine S., Singer Y. The hierarchical hidden markov model: Analysis and applications. *Machine Learning, Kluwer*, 32:41–62, 1998.
- [8] V. O. Frossard P. Joint source/fec rate selection of quality-optimal mpeg-2 video delivery. *IEEE Trans. Image Processing*, (10):1815–1825, 2001.
- [9] C. P. M. García-Frías J. Hidden markov models for burst error characterization in indoor radio channels. *IEEE Trans. Vehicular Tech*, 46:1006–1020, 1997.
- [10] P. J. C. J. F. Gomis V., Cabrera J. Combined markov model for packet loss characterization in umts channels. In *IEEE Vehicular Tech. Conf.*, pages 782–786, 2007.
- [11] S. P. R. M. Karner W., Nemethova O. Link error analysis and modeling for video streaming cross-layer design in mobile communication networks. *ETRI Journal*, 29:569–595, 2007.
- [12] R. H. L. D. Khayam S. A., Karande S. Performance analysis and modeling of errors and losses over 802.11b lans for high-bit-rate real-time multimedia. *Signal Processing: Image Communication, Elsevier*, 18:575–595, 2003.
- [13] J. A. D. L. R. Konrad A., Zhao B. Y. A markov-based channel model algorithm for wireless networks. *Wireless Networks, Kluwer*, 9:189–199, 2003.
- [14] K. W. D. Law A. M. *Simulation modeling and analysis*. McGraw-Hill, ISBN 0-07-100803-9, 1991.
- [15] G. E. N. Capacity of a burst-noise channel. *Bell System Technical Journal*, 39:1253–1264, 1960.
- [16] M. L. Nafaa A., Taleb T. Forward error correction strategies for media streaming over wireless networks. *IEEE Communications Magazine*, pages 72–79, January 2008.
- [17] T. T. Noorchashm M. Performance parameters of atm networks with bursts of errors. *Telecom Systems. Kluwer Academic Publishers.*, 16(3,4):255–269, 2001.
- [18] E. E. O. Estimates of error rates for codes on burts-noise channels. *Bell System Technical Journal*, 42:1977–1997, 1963.
- [19] W. A. Willig A., Kubisch M. Results of bit error rate measurements with an ieee 802.11 compliant phy. Technical Report TKN-00-008, Technical University Berlin, 2000.