Infocommunications Journal

A PUBLICATION OF THE SCIENTIFIC ASSOCIATION FOR INFOCOMMUNICATIONS (HTE)

April 2012 **Volume IV** **Number 1** **ISSN 2061-2079**

PAPERS

EUROPEAN RESEARCH

CALL FOR PAPERS

Technically Co-Sponsored by

IEEE COMMUNICATIONS SOCIETY

hte

IEEE HUNGARY SECTION

Editorial Board

Indexing information

Infocommunications Journal is covered by INSPEC and Compendex.

The journal is supported by NCA NEMZETI CIVIL ALAPPROGRAM the National Civil Fund.

Infocommunications Journal

Technically co-sponsored by IEEE Communications Society and IEEE Hungary Section

Editorial Office (Subscription and Advertisements)**:**
Scientific Association for Infocommunications
H-1055 Budapest, Kossuth Lajos tér 6-8, Room: 422
Mail Address: 1372 Budapest Pf. 451. Hungary
Phone: +36 1 353 1027, Fax: +36 1 353 0451
E-mail: info@hte.hu
Web: www.hte.hu

Articles can be sent also to the following address:
Budapest University of Technology and Economics
Department of Telecommunications
Tel.: +36 1 463 3261, Fax: +36 1 463 3263
E-mail: szabo@hit.bme.hu

Subscription rates for foreign subscribers:
4 issues 50 USD, single copies 15 USD + postage

Publisher: PÉTER NAGY • Manager: ANDRÁS DANKÓ

HU ISSN 2061-2079 • Layout: MATT DTP Bt. • Printed by: FOM Media

www.infocommunications.hu

A Novel Algorithm for Spectral Shaping of Binary Data Streams

Peter Vámos

***Abstract*—By generalizing the accumulated charge concept, we introduce a new class of constraints, the generalized charge constraint to control the spectral properties of a binary sequence. The new constraint limits the level at the output of a digital filter, diminishing those spectral components of the channel sequence being enhanced by the filter. A suitable coder structure, a feedback controlled bit stuff encoder is suggested to implement the new constraint. We demonstrate the spectral shaping property of the new coder structure and derive an approximate formula for the spectrum of the output binary signal. We also show that the coder performs a sigma-delta-like operation and the method is capable of implementing spectral and run-length constraints simultaneously. As a demonstration, we present a few particular spectral characteristics shaped by different examples of loop filters.**

Index Terms: Channel coding, Modulation codes, Run-length codes, Signal processing, Sigma-delta modulation.

I. INTRODUCTION

In many applications it is either impossible or at least difficult to shape the digital pulse. Typical examples are digital optical transmission and magnetic recording. In such applications it is important to perform an appropriate spectral shaping of the binary data stream [1], [2], [3]. Moreover, there are usually additional constraints on the code, such as a bound of the number of consecutive identical symbols, i.e., the run-length. The run-length upper bound (k constraint) ensures the reliable clock recovery, while establishing a lower bound (d constraint) diminishes the intersymbol interference, both practically important [4]. The d and k constraints together are referred to as run-length limiting or RLL constraints.

The most code constructions in the literature concentrate on dc-suppression [Chapters 7 and 8 of 2], and one can find only a few general purpose spectrum shaping algorithm, and even less capable to control the spectrum and the run-length simultaneously. In 1987 Marcus and Siegel published an algorithm which can produce spectral nulls at rational submultiples of the signalling frequency [5].

The guided scrambling algorithm developed by Fair et al. [6] makes dc-suppression by adding one or more redundant bits to the blocks of data stream. It minimize the accumulated charge on the output of a scrambler over the possible values of the redundant bits. Applying the weighted running digital sum (WRDS) concept introduced in present paper, the algorithm can be used even for general purpose spectral shaping. It is also claimed that algorithm limits the run-length as well, however, it is carried out partly by limiting the block length, and the k constraint can not be prescribed explicitly. It makes further difficulties to keep the run-length limit at the block boundaries, and guided scrambling is not suitable for imposing d constraint at all.

The algorithm proposed by Cavers and Marchetto can be taken as special case of guided scrambling with enhanced spectral shaping properties [7]. It minimizes (or maximizes) the output of a digital FIR filter representing the spectral constraint by inverting some data blocks along the filter. The inversion is marked on flag bits added to each block. By this method, however, can not be handled RLL constraints at all, and it uses the computationally expensive Viterbi algorithm for the encoding.

The bit stuffing approach has been applied for decades to control the run-length in binary sequences. The well known HDLC (High-level Data Link Control) protocol inserts a ‘0’ bit after each sequence of five consecutive ‘1’ bits [8]. In 1993 Bender and Wolf suggested a bit stuffing algorithm for generating run-length limited (RLL) sequences with spectral null at zero frequency [9]. However, their solution can scarcely be applied in practice due to its bent for infinite error propagation caused by the infinite memory of the decoder. Next to the error propagation, what can be kept under control by limiting the coder's memory, the only drawback of the bit stuff algorithm is that it requires buffering to keep the transmission rate constant.

In the second half of the 2000s many authors published improvements to the bit stuffing algorithm for coding (d, k) constrained channels [10], [11], [12]. The rates of these improved algorithms are very close to the channel capacity, and, in some specific cases, even they reach it. However, all they use that the bound of the current run-length is constant, what doesn't hold for charge and generalized charge constrained codes [13] presented in this paper.

In Section II-A of this paper we generalize the accumulated charge concept used for generating dc-suppressed code spectrum and introduce a new class of constraints the generalized charge constraint. The new constraint limits the level at the output of a digital filter, so the spectral requirements can be described easily. A feedback controlled bit stuff encoder with loop filter is suggested in Section II-B to implement the new constraint. This structure can perform a very effective and flexible spectrum shaping, and moreover, it can also control the run-length of the output bit stream. The flexibility is due to the digital loop filter, that can be implemented by inexpensive and readily available DSP components. In contrast to guided scrambling, the coder controls the output sequence

Manuscript received 19 August 2011, revised 8 February, accepted for publication 8 April, 2012.

The research was supported by the Foundation of Industry for Modern Technical Education (IKMA)

Peter Vámos is with the Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Budapest H-1521, Hungary. email: *vamos@tmit.bme.hu*

continuously, what makes easy to implement both k and d constraints, allows the application of IIR filters, and enhances the effectiveness of spectral shaping.

In Section II-C we give a brief analysis of the coder demonstrating its spectral shaping property and supply an approximate formula for the spectrum of the output binary signal in case of i.i.d. input. We also show that the coder performs a sigma-delta converter-like operation. We present that the coder can enforce spectral and run-length constraint simultaneously, and we extend the approximate formula for the spectrum for the case when k and spectral constraint are applied together.

In Section III we give a detailed analysis of coders with FIR loop filters. The performance of those coders can be described as finite sate machine (FSM) with the corresponding discrete Markov model. We also deal with the case when explicit run-length constraints are imposed next to the spectral one. As an example, we scrutinize an actual coder scheme with low-pass window filter generating dc-suppressed code for ac-coupled channels.

In Section IV coders with IIR loop filters are covered. These coders have an infinite state space and the corresponding Markov chain becomes unstable. So either we use functional equations for the description, or we approximate the actual Markov process with a finite space Markov chain. As example, we present dc-suppression, spectral notches and their combination shaped by different IIR loop filters.

II. The Coder's Structure and Performance

A. Generalization of the Accumulated Charge Concept

The most common application of spectral shaping is coding for ac-coupled channel. Those channels have a low frequency cut-off which affects the low frequency components of the code, causing a slow fluctuation in level at the receiver. To avoid this fluctuation, the coded sequence should be poor in low frequency spectral components. It is carried out by charge constrained codes when the accumulated charge of the channel sequence $Y_1, Y_2, \ldots$ is limited:

$$C_n = \sum_{i=0}^{n-1} Y_{n-i} \quad \text{and} \quad |C_n| \leq c \ \text{ for any } n \in \mathbf{N}. \tag{1}$$

In binary case ($Y_i \in \{-1,+1\}$) the sequence C is commonly called as running digital sum (RDS). One can see that the RDS is generated by a lowpass filter:

$$C(z) = \frac{1}{1-z^{-1}}\, Y(z), \tag{2}$$

where $Y(z)$ is the z- or discrete Fourier transform of sequence Y with $z = \exp(j2\pi f/f_0)$ and f_0 stands for the bit frequency. By limiting the RDS, we limit the level at the output of a lowpass filter. So, low frequency components of the channel sequence enhanced by the filter will be suppressed. (Actually, the filter in (2) has an infinite enhancement around the zero frequency, consequently, Y must have zero spectral density at zero frequency if RDS is limited. Pierobon [14] has proved the limited RDS is also a necessary condition.) It implies the idea using filters with other characteristics to form an RDS like

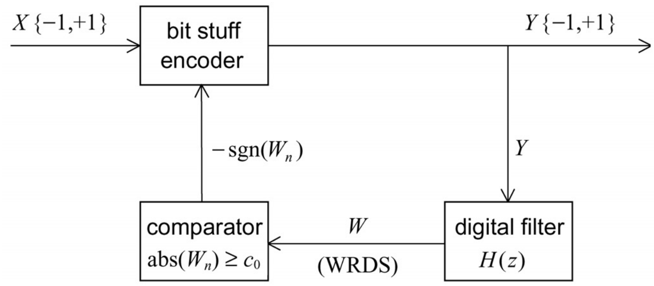


Fig. 1 Feedback controlled bit stuff encoder.

quantity will also shape the spectrum according to the applied filter. For this purpose, let us generalize the RDS concept by introducing the weighted running digital sum (WRDS):

Definition *The weighted running digital sum is the convolution of a binary sequence $Y_i \in \{-1,+1\}$ and a sequence of given constants $h_i \in \mathbf{R}$:*

$$W_n = \sum_{i=0}^{n-1} h_i\, Y_{n-i}, \quad \text{or} \quad W(z) = H(z)\, Y(z). \tag{3}$$

WRDS makes direct connection between the binary sequence and its spectrum. By limiting the WRDS we can enforce spectral constraints on the code. Those codes with limited WRDS we will refer to as generalized charge constrained or spectrum constrained codes.

B. Principle of the Coding Algorithm

For coding WRDS limited channels we are using the coder structure in Fig. 1. The coder is actually a feedback controlled bit stuff encoder. The bit stuff encoder has two states: It either transmits a bit from the input to the output or inserts a redundant bit. The bit stuffing is controlled by the feedback loop. Whenever the level at the filter's output reaches a given threshold c_0, the bit stuff encoder inserts a bit with opposite sign to the filter's output signal:

$$Y_{n+1} = \begin{cases} X_{m+1}, & \text{if} \quad |W_n| < c_0; \\ -\mathrm{sgn}(W_n), & \text{if} \quad |W_n| \geq c_0, \end{cases} \tag{4}$$

where $X_i, Y_i \in \{-1,+1\}$, and $W_n = \sum_{i=0}^{\infty} h_i\, Y_{n-i}$. The indices of the input (X) and output (Y) sequences are different because of the previously stuffed bits: $n = m + s_n$, where s_n stands for the number of stuffed bits till Y_n. For threshold c_0 it must hold that

$$\begin{aligned} c_0 > c_m &= \min_{\epsilon_i} \left|\sum \epsilon_i\, h_i\right|, \quad \epsilon_i \in \{-1,+1\}; \\ c_0 \leq c_M &= \sum |h_i|. \end{aligned} \tag{5}$$

For $c_0 \leq c_m$ the rate would be zero (no information could be transmitted), while for $c_0 > c_M$ the rate would be 1 (no spectral shaping could be performed). The above structure works actually as a negative feedback with loop filter. The coder tries to keep low the output of the filter $\widetilde{H}(z) = 1 + z^{-1} H(z)$. The spectral components enhanced by the filter will be dominant in the control of the bit stuff encoder, so the coder's interventions are diminishing their power.

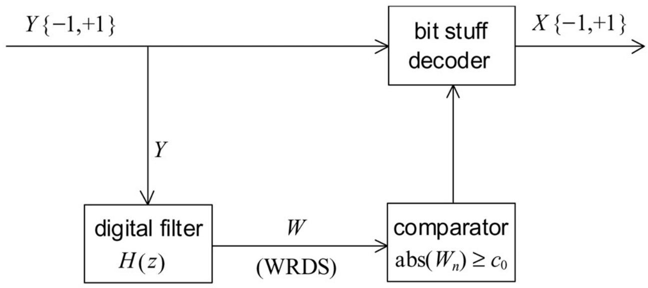


Fig. 2 The decoder's scheme.

The decoding can be performed with a similar, but feed-forward structure in Fig. 2. Whenever the forward filter's output reaches or exceeds the threshold the decoder removes a bit from the stream. Errors in transmission can cause additional errors during the decoding. In order to limit this error propagation we should limit the decoder's memory:

$$\sum |h_i| < \infty \,, \qquad (6)$$

i.e., all the poles of the loop filter should be outside the unit circle. It implies that only finite but arbitrarily large suppressions can be realized in the spectrum. The condition (6) ensures that the error propagation in the decoder always remains finite [15]. Further improvement can be reached in error propagation if we diminish the error probability at the bit stuff decoder's input by the application of an outer error correcting code with a Bliss scheme [16], [17].

Keeping the WRDS high by applying the coding rule

$$Y_{n+1} = \begin{cases} X_{m+1}, & \text{if} \quad |W_n| \geq c_0; \\ \operatorname{sgn}(W_n), & \text{if} \quad |W_n| < c_0, \end{cases}$$

instead of (4) can shape the spectrum as well. In this case the spectral components enhanced by the filter $\widetilde{H}(z)$ will be enhanced in the output signal too, so the spectral density of the output will emulate the filter's characteristics. This concept can be useful when the loop filter suitable for the desired spectrum doesn't ensure the stable performance of coder using coding rule (4).

C. Demonstration of the Spectral Shaping Property

To demonstrate the spectral shaping property, let us suppose that the input sequence X is a series of independent and identically distributed (i.i.d.) random variables, i.e. the input is a binary white noise. Under this circumstance, as far as the statistical properties of the output signal are concerned, there is no matter whether the coder inserts a bit or overwrites the incoming one, so the following coding rule can be set instead of (4):

$$Y_{n+1} = \operatorname{sgn}(X_{n+1} - \frac{1}{c_0} W_n).$$

To analyze the corresponding nonlinear system in Fig. 3.a, let us replace the one-bit quantizer by a quantization noise generator (Fig. 3.b). On the basis of the figure we can write:

$$\left.\begin{aligned} Y(z) &= X(z) - \tfrac{z^{-1}}{c_0} W(z) + Q(z) \\ W(z) &= H(z)\, Y(z) \end{aligned}\right\}$$

Expressing $Y(z)$, for the z-transform of the output signal we get:

$$Y(z) = \frac{X(z) + Q(z)}{1 + \frac{z^{-1}}{c_0} H(z)}. \qquad (7)$$

Now, supposing that process Q is an uncorrelated white noise [18], which is more or less satisfied while $0 < (c_0 - c_m)/(c_M - c_m) \ll 1$, i.e. when the coder performs definitely, for the spectral density of the output we have

$$S_Y(f) \approx \frac{1}{\left|1 + \frac{e^{-j2\pi f/f_0}}{c_0} H(e^{-j2\pi f/f_0})\right|^2}. \qquad (8)$$

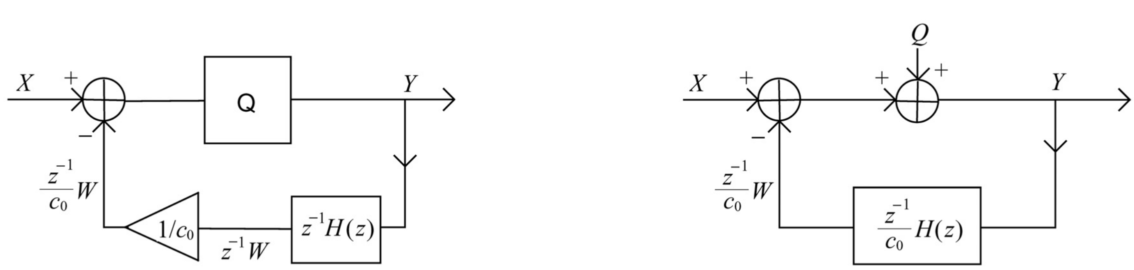
The above formula is an approximation, but one that describes well the main features of the code spectrum in most cases, and thus good basis for the coder's design.

To determine the suitable loop filter characteristics for the desired code spectrum, let us fix c_0 as 1. We can do it without loss of generality since the threshold's value can be included into the loop filter's characteristics, and it only trims the dynamics of the output spectrum, as can be seen in Fig. 8. So, the main features of the output spectrum are determined by the characteristics $\widetilde{H}(z) = 1 + z^{-1} H(z)$, and the loop filter can be designed on the basis of

$$H(z) = (\widetilde{H}(z) - 1)/z^{-1}.$$

From (7) one can see that the same filtering is applied both for the input signal X and the quantization noise Q. Using the notations

$$F(z) = \frac{1}{1 + \frac{z^{-1}}{c_0} H(z)} \quad \text{and} \quad X'(z) = F(z)\, X(z),$$



(a) (b)

Fig. 3 The coder's nonlinear equivalent circuitry (a); and its linearization (b).

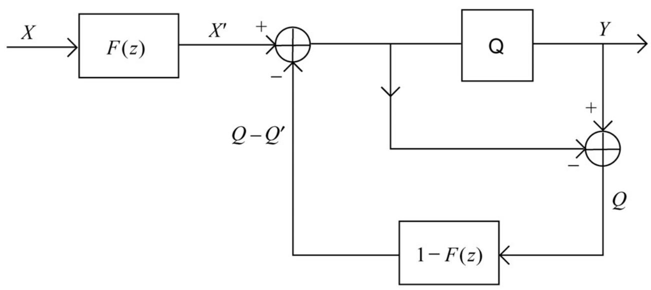


Fig. 4 The sigma-delta modulator-like equivalent of the coder structure.

as well as introducing the re-quantization noise as

$$Q'(z) = Y(z) - X'(z) = F(z)\,Q(z),$$

we can transform the layout in Fig. 3. The feedback quantizer on the right side of the yielding structure in Fig. 4 is exactly the circuitry suggested by Spang and Schultheiss for shaping the spectrum of the quantization noise [19], are widely used these days. It demonstrates the coder's performance well: the input binary signal is filtered according to the spectral requirements, then the resulted signal, in general with a continuous amplitude distribution, is re-quantized by a quantizer which also shapes the quantization noise spectrum in the same manner. We call the attention of the reader for the structural similarity of the layout in Fig. 4 and sigma-delta converters [20].

D. Mixing the Constraints

For a process with finite set of values there can be defined a corresponding run-length process:

Definition *A run is a substring of identical symbols. Let us define the transition times of the process Y as*

$$t_i = \min\{j > t_{i-1} | Y_j \neq Y_{j-1}\} \quad \text{and} \quad t_0 = 0.$$

Then the run-lengths are given as the differences $T_i = t_i - t_{i-1}$, and the process $T(Y)$ is called as the run-length process associated with Y.

In most applications it is also important to limit the run-length [1], [2]. In addition to some spectral constraint, now let us impose a (d, k) constraint as well upon the code, so no runs shorter than $d+1$ bits and longer than $k+1$ bits are allowed. ($d < k$ is always required.) To implement these constraints we should insert additional feedback loops into the coder for monitoring the run-length. The coding rule will be the following:

$$Y_{n+1} = \begin{cases} -\text{sgn}(W_n), \ \text{if } |W_n| \geq c_0; \ \text{(spectral constraint)} \\ -Y_n, \ \text{if } \left|\sum_{i=0}^{k} Y_{n-i}\right| = k+1; \ (k \text{ constraint}) \\ Y_n, \ \text{if } \left|\sum_{i=0}^{d} Y_{n-i}\right| < d+1; \ (d \text{ constraint}) \\ X_{m+1}, \quad \text{otherwise.} \qquad \text{(no stuffing)} \end{cases} \tag{9}$$

Some kind of spectral constraints may occasionally clash with one or other run-length constraints, and they are forcing the bit stuff encoder inserting bit with different sign at the same time. This can be either prevented by imposing an auxiliary constraint upon the code, or resolved by setting priorities for the constraints. An example for the former solution is presented in Section III-D when a low-pass window-filter is applied in the feedback loop to form a dc-suppressed (d, k) constrained code. When priorities are set, usually it is advisable to order higher priority to the run-length constraints since those are more crucial if they are set and generally the false interventions will not deteriorate the spectrum too much.

From (9) one can see that we are using low-pass FIR loop filters for monitoring the run-length both for d and k constraints. Moreover, k constraint is implemented with a coding rule that complies with (4), which maintains the WRDS low, similarly to one applied for the spectral constraint. It implies that when only k constraint is applied with spectral constraint, in case of independent binary white noise input, the bit stuff encoder can be replaced with summing circuits and one-bit quantizers, similarly as we did in the previous section, so the output spectrum can be estimated as

$$S_Y(z) \approx \frac{1}{\left|1 + \frac{z^{-1}}{c_0} H(z) + \frac{z^{-1}}{k+1} \frac{1 - z^{-(k+1)}}{1 - z^{-1}}\right|^2}. \tag{10}$$

The above formula is less accurate than (8) due to the fact that for k constraint the bit stuff threshold is set to $k+1$, i.e. the condition $(c_0 - c_m)/(c_M - c_m) \ll 1$ is barely satisfied since the loop works at the performance limit.

E. Coding Biased Sources

So far we have tacitly supposed that the input sequence is unbiased, i.e., the probabilities $p = \Pr(X = +1)$ and $q = \Pr(X = -1)$ are both 1/2. If this is not satisfied (or we have no information about it), the input signal has (or might have) a discrete component in dc proportional to the square of the bias $p - q$. Since the coder can produce only finite suppression, the discrete components can not be fully suppressed, and moreover, they stimulate the coder for unnecessarily large number of interventions diminishing the code rate. The usual solution to this problem is the precoding of the biased source [2], [21]. The precoded signal X' is defined as $X'_m = X_m X'_{m-1}$, $(X, X' \in \{-1, +1\})$. With the mapping $+1 \to 0$ and $-1 \to 1$ one can see that the precoding process is a mod 2 integration: $X'_m = X_m \oplus X'_{m-1}$. If the input process is i.i.d., the output is an i.i.d. run-length process with geometrical distribution: $\Pr(T_i(X') = n) = qp^{n-1}$ for any $i \in \mathbf{N}$, which has really no discrete spectral components. When $p = q = 1/2$, the statistical properties of the output signal are the same as that of the input.

For the sake of simple implementation, we have integrated the precoder with the bit stuff encoder by changing the coding rule of "no stuffing" cases using $Y_{n+1} = X_{m+1} Y_n$ instead of $Y_{n+1} = X_{m+1}$ in coding rules (4) and (9):

$$Y_{n+1} = \begin{cases} X_{m+1} Y_n, & \text{if } |W_n| < c_0; \\ -\text{sgn}(W_n), & \text{if } |W_n| \geq c_0. \end{cases} \tag{11}$$

The new coding rule will continue the current run with probability p (when "+1" inputted), and will start a new one with probability q (when "-1" inputted) when no stuffing is applied. However, when the bias is non-zero, the precoded signal will not be uncorrelated, so (8) and (10) can not be used to estimate the output signal's spectrum.

III. CODING WITH FIR FILTERS

According to coding rules (4) and (11) the output is determined by the WRDS and the input. For coders applying FIR filters, the momentary WRDS W_n is given by the sequence $Y_{n-r}^n = Y_{n-r}, Y_{n-r+1}, \ldots, Y_n$, where r is the filter's order. Since Y is a binary sequence, the coder may have only 2^{r+1} states, so it can be described as FSM with the corresponding state transition matrix $\mathbf{Q}$. Let s_i denote a particular state of the coder, then the elements of $\mathbf{Q}$, i.e. the transition probabilities are given as $q_{i,j} = \Pr(Y_{n-r}^n = s_j \mid Y_{n-r-1}^{n-1} = s_i)$.

A. The Special Properties of the Transition Matrix

For the analysis we will use the following symmetry properties of the transition matrix.

Property 1 *Let denote N the number of internal states of the coder and $\bar{s}_i$ the bitwise inverse of the state s_i Then with the labelling*

$$\bar{s}_i = s_{N+1-i}, \quad (i = 1, 2, \ldots, N) \tag{12}$$

the coder's transition probability matrix is centrosymmetric, i.e., $q_{i,j} = q_{N+1-i,N+1-j}$, or with the exchange matrix $\mathbf{J}$ having ones only on the reverse diagonal:

$$\mathbf{J}\,\mathbf{Q}\,\mathbf{J} = \mathbf{Q}. \tag{13}$$

Proof: The states s_i and $\bar{s}_i$ have WRDS with the same magnitude but with opposite sign. The coding rule (11) implies that if s_i transits into s_j for a given input, then $\bar{s}_i$ transits into $\bar{s}_j$ for that very same input. So we can write:

$$\begin{aligned} q_{i,j} &= \Pr(Y_{n-r}^n = s_j \mid Y_{n-r-1}^{n-1} = s_i) \\ &= \Pr(Y_{n-r}^n = \bar{s}_j \mid Y_{n-r-1}^{n-1} = \bar{s}_i) \\ &= \Pr(Y_{n-r}^n = s_{N+1-j} \mid Y_{n-r-1}^{n-1} = s_{N+1-i}) \\ &= q_{N+1-i,N+1-j}\,. \end{aligned} \tag{14}$$

■

(13) implies that the transition matrix can be given in the following form [22]:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2\mathbf{J} \\ \mathbf{J}\mathbf{Q}_2 & \mathbf{J}\mathbf{Q}_1\mathbf{J} \end{bmatrix}. \tag{15}$$

Property 2 *It is a necessary and sufficient condition for centrosymmetry if the matrix has two invariant subspaces orthogonal to each other. An even one ($\mathcal{E}$) consisting of $\mathbf{v}_e = [\mathbf{v}, \mathbf{vJ}]$ even, and an odd one ($\mathcal{O}$) consisting of $\mathbf{v}_o = [\mathbf{v}, -\mathbf{vJ}]$ odd vectors.*

Proof: The necessity can be proven with the help of decomposition (15):

$$\mathbf{v}_e\mathbf{Q} = [\mathbf{v}(\mathbf{Q}_1 + \mathbf{Q}_2),\ \mathbf{v}(\mathbf{Q}_1 + \mathbf{Q}_2)\mathbf{J}] = [\mathbf{v}\mathbf{Q}_e,\ \mathbf{v}\mathbf{Q}_e\mathbf{J}]; \tag{16.a}$$

$$\mathbf{v}_o\mathbf{Q} = [\mathbf{v}(\mathbf{Q}_1 - \mathbf{Q}_2), -\mathbf{v}(\mathbf{Q}_1 - \mathbf{Q}_2)\mathbf{J}] = [\mathbf{v}\mathbf{Q}_o, -\mathbf{v}\mathbf{Q}_o\mathbf{J}], \tag{16.b}$$

where $\mathbf{Q}_e = \mathbf{Q}_1 + \mathbf{Q}_2$ and $\mathbf{Q}_o = \mathbf{Q}_1 - \mathbf{Q}_2$ are the equivalent transformations of the $N/2$ dimensional reduced subspaces.

The sufficiency follows from that for any $\mathbf{v} \in \mathcal{E} \cup \mathcal{O}$: $\mathbf{vQ} = \mathbf{vJQJ}$, that is, $\mathbf{Q}$ and $\mathbf{JQJ}$ are equivalent. ■

There are two important corollaries of the above properties:

- From (14) one can see that each state can be joined with its inverse, due to the symmetry. Then the state transition probability matrix reads as $\mathbf{Q}_1 + \mathbf{Q}_2 = \mathbf{Q}_e$.
- Since a matrix and its any power have the same eigenvectors, if $\mathbf{Q}$ is centrosymmetric, $\mathbf{Q}^{\pm n}$ is alike, and $(\mathbf{Q}^{\pm n})_e = \mathbf{Q}_e^{\pm n}$ and $(\mathbf{Q}^{\pm n})_o = \mathbf{Q}_o^{\pm n}$.

Property 3 *Labelling the states according to the last output bit Y_n too as*

$$\begin{aligned} 1 &\le i \le N/2, \quad \text{if} \quad Y_n = +1; \\ N/2 &< i \le N, \quad \text{if} \quad Y_n = -1; \end{aligned} \tag{17}$$

matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$ have actual physical meaning: $\mathbf{Q}_1$ stands for repeating the last input bit, i.e. continuing the current run, while $\mathbf{Q}_2$ stands for adding an inverse bit to the last one starting a new run with opposite sign. According to coding rule (11), it implies that any non-zero and non-one elements of $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are p and q respectively.

Commonly, applying an order r FIR loop filter, the most plausible labelling satisfying both (12) and (17) is the lexicographical ordering of the filter's states:

$$i = \sum_{k=0}^{r} (1 - Y_{n-k})\, 2^{r-k-1} + 1,$$

$$\text{i.e.,} \quad s_1 = [+1, \ldots, +1], \ldots, s_{2^{r+1}} = [-1, \ldots, -1].$$

B. The Properties of Bit-Stuff Generated Finite State Codes

The redundancy of a bit stuff encoder stems from stuffings, so the coder's rate can be calculated from the stuffing probability P_{stuff}:

$$R = 1 - P_{\text{stuff}}.$$

The stuffing probability is given by the sum of stationary probabilities of states where stuffing is applied:

$$P_{\text{stuff}} = \sum_{|\mathrm{WRDS}(s_i)| \ge c_0,\, i \le \frac{N}{2}} \pi_i,$$

where π_i is the stationary probability of state s_i, i.e., the element of eigenvector $\boldsymbol{\pi}_e$ of $\mathbf{Q}_e$ associated with the maximal eigenvalue 1. (Due to the symmetry the stationary distribution $\boldsymbol{\pi}$ of $\mathbf{Q}$ must be element of $\mathcal{E}$: $\boldsymbol{\pi} = \frac{1}{2}[\boldsymbol{\pi}_e, \boldsymbol{\pi}_e\mathbf{J}]$; so it is determined by $\mathbf{Q}_e$ only.)

If the coding algorithm is greedy, i.e., it can generate all the possible sequences obeying the given constraint, the set of edges of the coder's and the constrained channel's state transition graph are identical. So, one can get the constrained channel's adjacency matrix from the transition probability matrix by substituting ones in places of its nonzero elements: $\mathbf{A} = [\mathbf{Q} \neq 0]$. Since $\mathbf{A}$ is centrosymmetric too, and $\mathbf{A}_e = [\mathbf{Q}_e \neq 0]$ is non-negative, so it comes into the maximal eigenvalue $\lambda_{\max}$ of $\mathbf{A}$, which determines the channel capacity given as $C = \log_2 \lambda_{\max}$ [2], [23].

The spectral density of a Markov chain is defined as the Fourier transform of the autocorrelation:

$$S(f) = \sum_{k=-\infty}^{\infty} R(k) e^{-j2\pi k f/f_0} \tag{18}$$

So, to get the spectral density, first we should determine the autocorrelation $R_Y(k) = \mathrm{E}(Y_n Y_{n+k})$. Using that the coder's states are ordered such a manner that $Y_n = +1$ for $S_1 \dots S_{\frac{N}{2}}$ and $Y_n = -1$ for $S_{\frac{N}{2}+1} \dots S_N$, and moreover, the symmetry of the stationary distribution $\boldsymbol{\pi}$, the autocorrelation can be written as

$$R_Y(k) = \frac{1}{2} [\boldsymbol{\pi}_\mathrm{e}, -\boldsymbol{\pi}_\mathrm{e}\mathbf{J}] \, \mathbf{Q}^{|k|} \begin{bmatrix} \mathbf{1} \\ -\mathbf{1} \end{bmatrix},$$

where $\mathbf{1}$ denotes the full-of-one vector $[1, 1, \dots, 1]^*$. Since the vector on the left is element of $\mathcal{O}$, according to (16.b), the autocorrelation can be expressed with $\mathbf{Q}_\mathrm{o}$:

$$R_Y(k) = \frac{1}{2} [\boldsymbol{\pi}_\mathrm{e}\mathbf{Q}_\mathrm{o}^{|k|}, -\boldsymbol{\pi}_\mathrm{e}\mathbf{Q}_\mathrm{o}^{|k|}\mathbf{J}] \begin{bmatrix} \mathbf{1} \\ -\mathbf{1} \end{bmatrix} = \boldsymbol{\pi}_\mathrm{e}\mathbf{Q}_\mathrm{o}^{|k|}\mathbf{1}. \tag{19}$$

Substituting (19) into (18), for the spectrum we get:

$$S_Y(z) = \boldsymbol{\pi}_\mathrm{e}[(\mathbf{I} - z\mathbf{Q}_\mathrm{o})^{-1} + (\mathbf{I} - z^{-1}\mathbf{Q}_\mathrm{o})^{-1} - \mathbf{I}]\,\mathbf{1}.$$

C. Taking the Run-Length into Account

To describe the simultaneously spectrum and (d, k) constrained channel, let us set out from the state transition graph of the RLL channel in Fig. 5. The labelling of the vertices corresponds to the current run-length. Taking into account the spectral constraint, we should use a hyper graph: Each vertex contains a set of states with same length of closing run represented by vectors, and each edge corresponds to an edge adjacency matrix describing the connection between the state vectors of neighboring vertices. There are two kinds of edge matrices: $\mathbf{A}_1$ which continues the current run, and $\mathbf{A}_2$ which closes it, starting a new run with opposite sign, as $\mathbf{Q}_1$ and $\mathbf{Q}_2$ do. So, those can be derived from the original state transition probability matrix $\mathbf{Q}$ by putting ones in place of nonzero elements of $\mathbf{Q}_1$ and $\mathbf{Q}_2$ respectively:

$$\mathbf{A}_1 = [\mathbf{Q}_1 \neq 0] \quad \text{(the run is continuing);}$$
$$\mathbf{A}_2 = [\mathbf{Q}_2 \neq 0] \quad \text{(a new run is starting).}$$

Using the variable length symbol representation [24], [25], i.e. the edges can correspond to sequences of different length, the state transition diagram can be reduced into a one-vertex graph (Fig. 6), which is described with the following adjacency matrix:

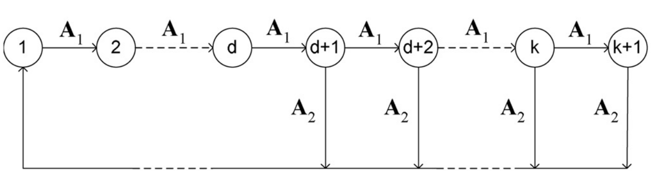


Fig. 5 The state transition graph of the (d, k) constrained channel.

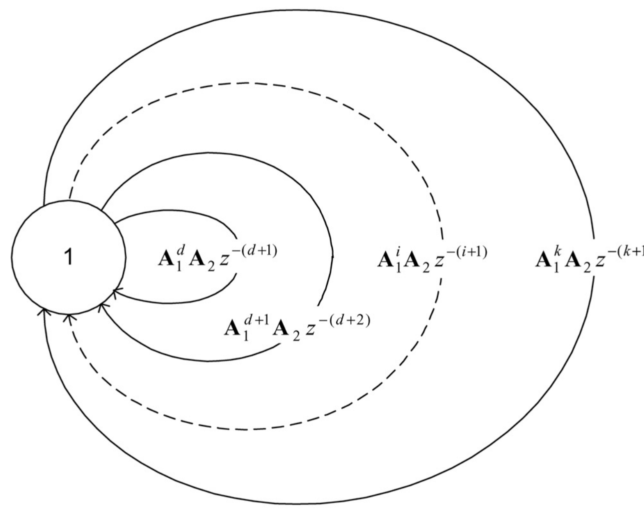


Fig. 6 The variable length graph of the simultaneously spectrum and (d, k) constrained channel.

$$\mathbf{A}_{d,k}(z) = \sum_{i=d}^{k} z^{-(i+1)} \mathbf{A}_1^i \mathbf{A}_2.$$

Then the channel capacity is given as the base two logarithm of the largest root of the characteristic polynomial $\det[\mathbf{A}_{d,k}(z) - \mathbf{I}]$ [23], [24].

The zero capacity indicates that the spectral and run-length constraints can not be matched without breaching either of them, so setting priorities is inevitable. Conversely, the non-zero capacity means that the constraints can be straight matched or with the help of some auxiliary constraints at most.

The state transition diagram of a coder for simultaneously spectrum and (d, k) constrained channel can be seen in Fig. 7, which is a hyper graph too. The edge matrix $\widetilde{\mathbf{Q}}_1$ corresponds to continuing the current run, while $\widetilde{\mathbf{Q}}_2$ stands for closing it by inserting $d+1$ bits with opposite sign, and $\widetilde{\mathbf{A}}_2$ does the same but unconditionally. If the run-length and spectral constraints never clash, these matrices can be straight derived from the original spectrum constrained system's transition matrix:

$$\widetilde{\mathbf{Q}}_1 = \mathbf{Q}_1; \quad \widetilde{\mathbf{Q}}_2 = \mathbf{Q}_2\mathbf{A}_1^d; \quad \widetilde{\mathbf{A}}_2 = \mathbf{A}_2\mathbf{A}_1^d.$$

The clash of constraints means that there are states with transition probability less than 1. Making the matrix $\widetilde{\mathbf{Q}}_1 + \widetilde{\mathbf{Q}}_2$ stochastic by successive removing the dead end states or

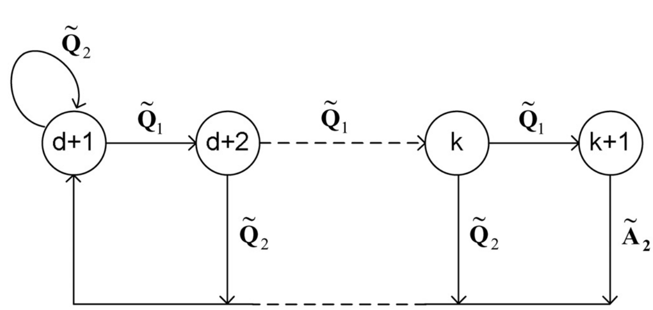


Fig. 7 The state transition diagram of the coder.

inserting new transitions, and then setting the probability of lonely transitions to 1, we can patch the broken Markov chain in vertices "d+1",…,"k". Establishing new transitions in $\widetilde{\mathbf{Q}}_2$ corresponds to setting priority of d constraint over the spectral constraint, while by removing dead end states or turning the probability of the single transitions to 1, we implement auxiliary constraints which can preserve the validity of both constraints. Moreover, we should secure the prevailing of k constraint in vertex "$k+1$" by adding transitions to $\widetilde{\mathbf{A}}_2$ in states where k and spectral constraints collide.

On the basis of Fig. 7 we can construct the variable length transition matrix of the coder:

$$\mathbf{Q}_{d,k}(z) = \sum_{i=0}^{k-d-1} z^{d+1+i}\,\widetilde{\mathbf{Q}}_1^i\widetilde{\mathbf{Q}}_2 + z^{k+1}\,\widetilde{\mathbf{Q}}_1^{k-d}\widetilde{\mathbf{A}}_2.$$

Let $\boldsymbol{\pi}_{d,k} = \boldsymbol{\pi}_{d,k}\mathbf{Q}_{d,k}(1)$ the stationary distribution vector for vertex "$d+1$", i.e., the stationary distribution under the condition that a new run has started. Then the generating function of the distribution of run-length can be given as

$$g(z) = \boldsymbol{\pi}_{d,k}\mathbf{Q}_{d,k}(z)\,\mathbf{1}. \tag{20}$$

To get the coder's rate, first we determine the average run-length at the output, what is given by $g'(1)$:

$$\overline{N}_{\text{out}} = \boldsymbol{\pi}_{d,k}\left[\sum_{i=0}^{k-d-1}(d+1+i)\,\widetilde{\mathbf{Q}}_1^i\widetilde{\mathbf{Q}}_2 + (k+1)\,\widetilde{\mathbf{Q}}_1^{k-d}\widetilde{\mathbf{A}}_2\right]\mathbf{1}.$$

Introducing the indicator vector $\mathbf{i}=\mathbf{1}-[\widetilde{\mathbf{Q}}_1\mathbf{1}==1]-[\widetilde{\mathbf{Q}}_2\mathbf{1}==1]$, which selects the states when no stuffing is applied and the coder gets a bit from the input, the average number of input bits during an output run reads as

$$\overline{N}_{\text{in}} = \boldsymbol{\pi}_{d,k}\sum_{i=0}^{k-d-1}\widetilde{\mathbf{Q}}_1^i\,\mathbf{i}.$$

Then the rate is given as $R = \overline{N}_{\text{in}}/\overline{N}_{\text{out}}$.

The spectral density of the output signal can also be calculated with the help of generating function (20) by the formula published in [26]:

$$S_Y(z) = \frac{4\,\boldsymbol{\pi}_{d,k}[(\mathbf{I}+\mathbf{Q}_{d,k}(z))^{-1} + (\mathbf{I}+\mathbf{Q}_{d,k}(z^{-1}))^{-1} - \mathbf{I}]\,\mathbf{1}}{\overline{N}_{\text{out}}\,|1-z|^2}.$$

D. An Example: The Window-Charge Constraint

In many applications (e.g. ac-coupled channels) it is an important requirement that the code spectrum should be poor around the zero frequency [2], [27], [28]. With the application of a low-pass loop filter we can satisfy this requirement. Using the order r window filter $H(z) = \sum_{i=0}^{r} z^{-i}$ as loop filter, according to (8), the output sequence will have a dc-suppressed spectrum with ripples in passband region (Fig. 8). Increasing the order of the filter, the suppression will grow, while the width of the suppressed region will diminish. Certainly, diminishing the threshold will enhance the suppression. To reach the same suppression the filter with higher order will result in a higher code rate.

Now the WRDS is defined as $W_n = \sum_{i=0}^{r} Y_{n-i}$. The values of possible W_n's form a finite set of even or odd integers depending on the value of r. Without loss of generality, we can confine the coders's threshold value on that very same set, supposing that c_0 is a mod 2 congruent integer to $r+1$: $(r+1-c_0) \bmod 2 \equiv 0$. Then the coding rule reads as

$$Y_{n+1} = \begin{cases} X_{m+1}Y_n, & \text{if } \left|\sum_{i=0}^{r} Y_{n-i}\right| < c_0; \\ -\text{sgn}(W_n), & \text{if } \left|\sum_{i=0}^{r} Y_{n-i}\right| = c_0. \end{cases} \tag{21}$$

Taking the incoming bit into account, one can see that the coder limits the accumulated charge in an $r+2$ bit long sliding window, and the channel sequence will comply with the following constraint:

$$\left|\sum_{i=0}^{r+1} Y_{n-i}\right| \le c_0 - 1 \quad \text{for any } n \in \mathbf{N}.$$

We refer the constraints of above type to as window-charge, or shortly (w,c) constraint. The w and c code parameters stand for the window's length, now $w=r+2$, and the charge limit, now $c=c_0-1$. The coding algorithm defined by (21) is greedy for the above constraint, i.e., it can generate all the possible sequences obeying the constraint with parameters $(r+2, c_0-1)$. Certainly, the output sequence will also comply with the constraint $(r+1, c_0)$, however, the coding algorithm will not be greedy for that constraint because it makes an unnecessary stuffing when $|W_n| = c_0$ and the outgoing bit $Y_{n-r} = \text{sgn}(W_n)$. It implies that a sequence obeying the constraint $(r+1, c_0)$ will not necessarily comply with $(r+2, c_0-1)$, i.e., the latter one is a stronger condition.

The performance of the coder can be analyzed with the help of the loop filter's states. Let us call a state light if its disparity is smaller than the bound: $|W| < c_0$; and heavy if those are equal: $|W| = c_0$. Introducing the notations $n_1 = (r+1-c_0)/2$ and $n_2 = (r+1+c_0)/2$ for the minimum and maximum number of identical bits in the window, the number of light and heavy states are

$$N_l = \sum_{i=n_1+1}^{n_2-1}\binom{r+1}{i}; \quad N_h = \binom{r+1}{n_1} + \binom{r+1}{n_2} = 2\binom{r+1}{n_1}.$$

In case of unbiased, i.i.d. input the stationary distribution can be determined simply:

Theorem *If the input process X is i.i.d. with* $\Pr(X=+1) = \Pr(X=-1) = 1/2$*, the stationary distribution of the coder's states depends only on the weight of the states. The probabilities of all light states and the probabilities of all heavy states are equal, and the probability of a light state is the double of a heavy one:*

$$p_l = \frac{2}{2N_l + N_h}; \qquad p_h = \frac{1}{2N_l + N_h}.$$

Proof: According to coding rule, each light state has two parents and two children, while a heavy state has only one of each. Since the input process is i.i.d. and $\Pr(X=+1) = \Pr(X=-1) = 1/2$, the stationary probability distribution is

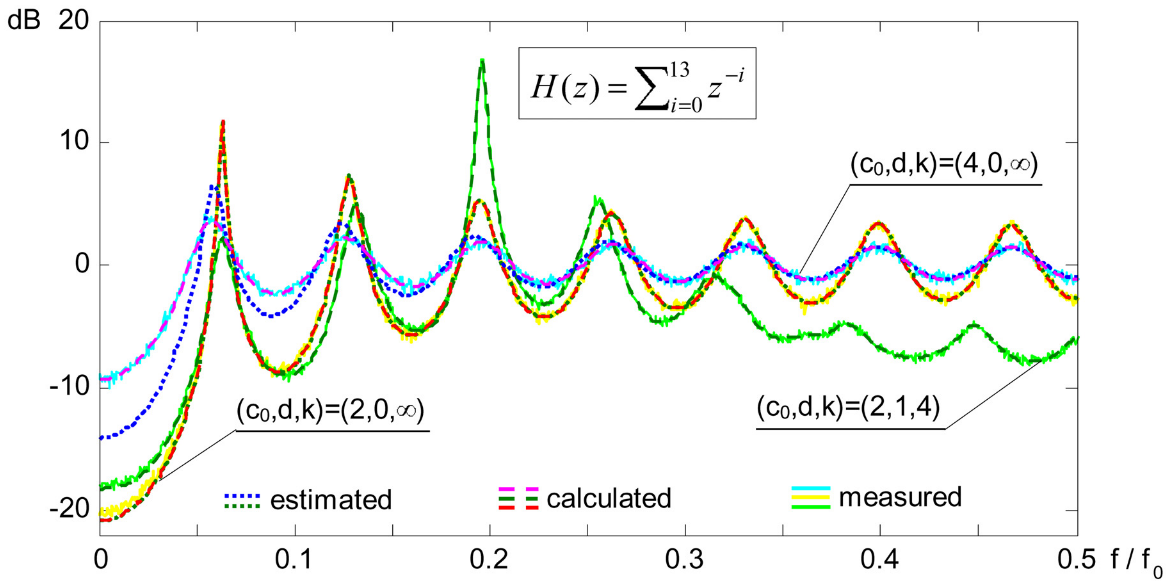


Fig. 8 Dc-suppressed code spectra generated by an order 13 window filter.

determined by the set of following four kinds of equations:

$\pi_i = \frac{1}{2}\pi_j + \frac{1}{2}\pi_k$, if state i, j and k are each light;
$\pi_i = \frac{1}{2}\pi_j + \pi_k$, if state i and j are light while k is heavy;
$\pi_i = \frac{1}{2}\pi_j$, if state i is heavy and j is light;
$\pi_i = \pi_j$, if state i and j are both heavy.

The above system of linear equations can be satisfied with any π_i's such that

$$\pi_i = \begin{cases} 2p_h, & \text{if state } i \text{ is light;} \\ p_h, & \text{if state } i \text{ is heavy.} \end{cases}$$

Then the probability p_h can be determined by the condition $\sum \pi_i = 1$:

$$p_h = \frac{1}{2N_l + N_h}.$$

According to Perron–Frobenius theorem [29], the above solution is unique as well. ∎

The redundancy is given by the stuffing probability $\Pr(|W| = c)$, so the rate can be calculated as

$$R = 1 - \Pr(|W| = c_0) = 1 - p_h N_h = \frac{2N_l}{2N_l + N_h}.$$

The error propagation, i.e., the expected value of the number of false detections induced by a single error, can be calculated under the condition that the error and the channel process are independent. An error during the transmission can cause two kinds of errors at the decoder. The type one error is when an originally light state with disparity c_0-2 is detected as heavy, causing a false removal. Its probability is

$$p_{e1} = 2p_l \frac{n_1+1}{r+1}\binom{r+1}{n_1+1} = 2p_h \frac{n_2}{r+1} N_h.$$

The type two error occurs when a heavy state is detected as light, leaving a redundant bit in the stream. The probability of this error is

$$p_{e2} = 2\,p_h \frac{n_2}{r+1}\binom{r+1}{n_2} = p_h \frac{n_2}{r+1} N_h.$$

Then the total decoding error probability, under the condition that a single error has occurred during the transmission, reads as

$$p_e = p_{e1} + p_{e2} = 3\,\frac{n_2}{r+1}\,p_h N_h = \frac{3}{2}\left(1 + \frac{c_0}{r+1}\right)(1-R).$$

Since the erroneous bit remains in the memory of the decoder for $r+1$ steps, the expected value of the number of false detections induced by a single error is

$$\mathrm{E}(n_e) = (r+1)\,p_e = \frac{3}{2}\,(r+1+c_0)\,(1-R).$$

The (w, c) constraint also imposes an upper bound on the run-length, since a run can not be longer than $(w + c)/2$ bits and the special case when $w = c+2$ exactly corresponds to the run-length constraint $k = c$. However, we can prescribe RRL constraints explicitly as well. According to (9), the coding rule will be the following:

$$Y_{n+1} = \begin{cases} -\mathrm{sgn}(W_n), & \text{if } \left|\sum_{i=0}^{r} Y_{n-i}\right| = c_0\,; \\ -Y_n, & \text{if } \left|\sum_{i=0}^{k} Y_{n-i}\right| = k+1; \\ Y_n, & \text{if } \left|\sum_{i=0}^{d} Y_{n-i}\right| < d+1; \\ Y_n, & \text{if } Y_n \sum_{i=0}^{r-d} Y_{n-i} \le d - c_0\,; \quad (*) \\ X_{m+1} Y_n, & \text{otherwise} \quad \text{(no stuffing).} \end{cases}$$

The last stuffing case denoted by $(*)$ is an auxiliary constraint. It is applied to avoid the collision of (w, c) and d constraints after short runs, forcing the coder to lengthen the current run.

The coder's transition matrix should be modified accordingly by turning the probabilities of lonely transitions to 1 making the matrix stochastic:

$$\widetilde{\mathbf{Q}}_1 = \mathrm{diag}(\mathbf{1} - \widetilde{\mathbf{Q}}_2\mathbf{1})\mathbf{A}_1.$$

The spectrum of the coded signal can be seen in Fig. 8.

IV. CODING WITH IIR FILTERS

In case of IIR loop filters, there are infinite many non-zero among the coefficients h_i in definition of WRDS. So, as the process advances, the number of possible values of W, i.e., state space of the Markov chain is growing permanently and tends to infinite. Moreover, along with that growing, the Markov chain is getting unstable, i.e., the probabilities of the states tend to zero, so the discrete Markov model can not be used anymore. For the exact mathematical description the process should be taken as Markov process with a multidimensional continuous state space. The dimension of sate space corresponds to the applied filter's order. The transitions are described as functions of the state space, and the stationary distribution can be earned as the solution of a functional equation. We present the method with first order low-pass loop filter, however, the whole method is cumbersome and hard to solve even in this simplest case.

Rather than following the exact model, we can approach the original Markov process with a Markov chain by discretizing the state space. Then it can be handled with the matrix method applied for coders with FIR loop filters. It can fairly model the original system since, in the strict sense, the actual coder also performs as Markov chain rather than Markov process due to the roundoff errors of the filter circuits. The only difficulty with the method is the state space is prone to getting fast unmanageably large by growing the filter's order. Therefore, it is useful to find the exact support of the multidimensional distribution function before the discretization, which can be itself a hard problem.

A. The α-Charge Constrained Code

Applying the IIR low-pass filter $H(z) = 1/(1 - \alpha z^{-1})$ $(0<\alpha<1)$ as loop filter, the WRDS is defined as

$$W_n = \sum_{i=0}^{n-1} \alpha^i Y_{n-i} = Y_n + \alpha W_{n-1}. \tag{22}$$

In accordance with (8), it will also result in a DC-suppressed code spectrum, as can be seen in Fig. 9. The greater α we use, the deeper and steeper suppression, but the error propagation is also increased. On the basis of (5), for threshold c_0 it should hold that $1/(1+\alpha) < c_0 < 1/(1-\alpha)$. According to coding rule (4), the WRDS defined by (22) is bound by $c = \alpha c_0 + 1$, and the output sequence will obey the following constraint:

$$\left| \sum_{i=0}^{n-1} \alpha^i Y_{n-i} \right| < c \quad \text{for any} \quad n \in \mathbf{N}.$$

This constraint will be referred to as α-charge or shortly (α, c) constraint. When $\alpha = 1$, we get the conventional charge (RDS) constraint, however, this value of α should not be used with the bit stuffing method since it breaches the finite memory condition (6) causing an infinite error propagation with a probability of 1 for any finite error rate. The coding algorithm defined by (4) and (22) is greedy in generating $(\alpha, \alpha c_0 + 1)$ constrained sequences.

To find the capacity of α-charge constrained channel, let us consider the set $\mathcal{A}^n$ of n dimensional binary vectors: $[a_1, a_2, \ldots, a_n]$, $a_i \in \{-1, +1\}$, and define the rectified WRDS (RWRDS) on the elements of $\mathcal{A}^n$ as

$$\begin{aligned} \widehat{W}_i &= a_i \textstyle\sum_{j=0}^{i-1} \alpha^j a_{i-j} \quad (i = 1, \ldots, n) \\ &= 1 + \frac{a_i}{a_{i-1}} \alpha \widehat{W}_{i-1} \; (i = 2, \ldots, n) \quad \text{and} \quad \widehat{W}_1 = 1. \end{aligned} \tag{23}$$

The RWRDS of an α-charge constrained sequence is confined within the interval $(1 - \alpha c,\ c)$. For the lower bound we will use the shortcut $c' = 1 - \alpha c$. It is convenient to rectify the WRDS according to the sign of the current run since RWRDS is always increasing during a run, so it is enough to set an upper bound:

$$\left| \sum_{i=0}^{n-1} \alpha^i Y_{n-i} \right| < c \;\Leftrightarrow\; \sum_{i=0}^{n-1} Y_n \alpha^i Y_{n-i} < c$$

Now let us consider the subset $\mathcal{A}_c^n$ of $\mathcal{A}^n$ where the RWRDS is limited: $\widehat{W}_1, \widehat{W}_2, \ldots, \widehat{W}_n < c$. Let $S(n)$ denote the number of elements in $\mathcal{A}_c^n$ and $F_n(x)$ the distribution of $\widehat{W}_n$ on the set, i.e., the probability that the RWRDS of a randomly chosen element of $\mathcal{A}_c^n$ is smaller than x: $F_n(x) = \Pr(\widehat{W}_n < x)$. Then, on the basis of (23), we can write:

$$\begin{aligned} S(n)F_n(x) = \;& S(n-1)F_{n-1}(\tfrac{x-1}{\alpha}) \\ &+ S(n-1)[1 - F_{n-1}(-\tfrac{x-1}{\alpha})]. \end{aligned} \tag{24}$$

The term $F_{n-1}(\frac{x-1}{\alpha})$ in the above equation refers to the case when the current run is continued, while $1 - F_{n-1}(-\frac{x-1}{\alpha})$ is standing for starting a new run, and the arguments are the values of RWRDS of the step before. Specially, for $x = c$ $F_n(c) = 1$, while, since the value of the second argument $(1-c)/\alpha = -c_0$ is always below c', the probability $F_{n-1}(-\frac{c-1}{\alpha})$ is zero. Thus for $x = c$ (24) reads as

$$S(n) = S(n-1)\,[1 + F_{n-1}(\tfrac{c-1}{\alpha})]. \tag{25}$$

Combining (24) and (25), function $F_n(x)$ can be given recurrently:

$$F_n(x) = \frac{F_{n-1}(\frac{x-1}{\alpha}) + [1 - F_{n-1}(-\frac{x-1}{\alpha})]}{1 + F_{n-1}(\frac{c-1}{\alpha})}.$$

On the basis of the above recurrence, for the stationary distribution $F(x) = \lim_{n\to\infty} F_n(x)$ we get the following functional equation:

$$F(x) = \begin{cases} 0, & \text{if } x \le c'; \\ \dfrac{1 + F(\frac{x-1}{\alpha}) - F(-\frac{x-1}{\alpha})}{1 + F(\frac{c-1}{\alpha})}, & \text{if } c' < x \le c; \\ 1, & \text{if } x > c. \end{cases} \tag{26}$$

From (25) one can see that the number $S(n)$ of elements in $\mathcal{A}_c^n$ increases exponentially for large n's:

$$S(n) \asymp [1 + F(\tfrac{c-1}{\alpha})]^n,$$

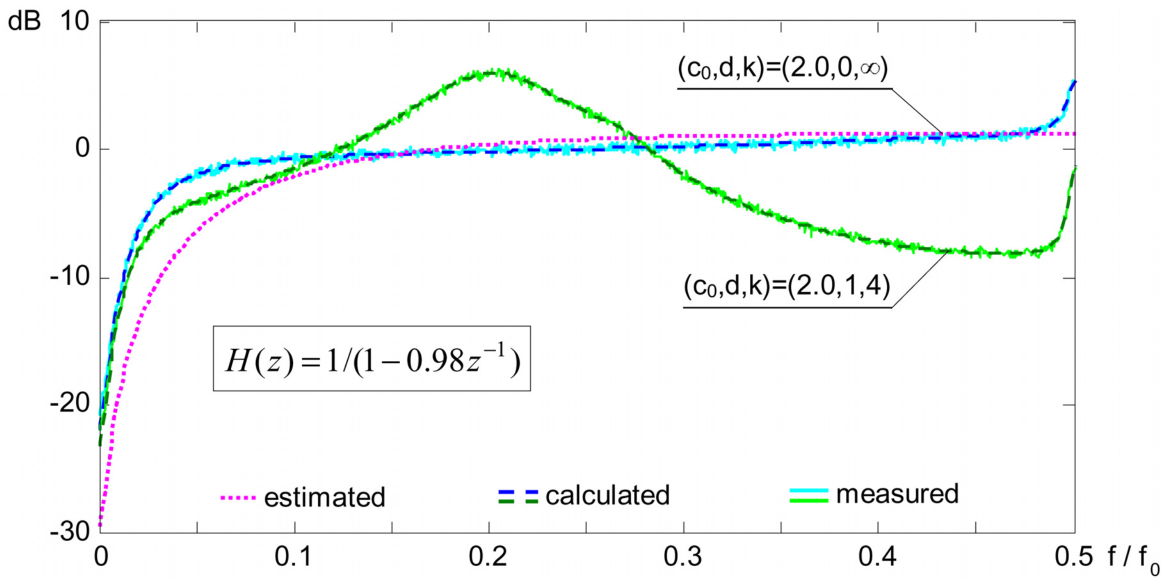


Fig. 9 Dc-suppression by IIR low-pass loop filter with no run-length constraint (blue/magenta), and with an additional (d, k) constraint (green).

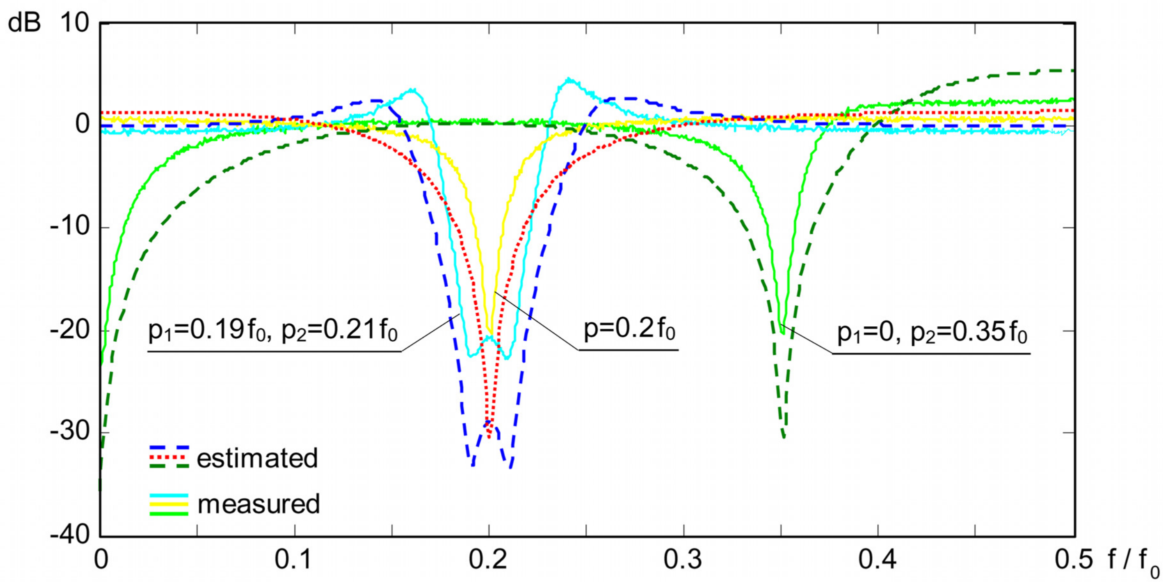


Fig. 10 Spectral notch by IIR band pass loop filter with a single pole at $0.2f_0$ (yellow/red), widened by double poles at $0.19f_0$ and $0.21f_0$ (cyan/blue), and combined with dc-suppression (light green/green).

thus the channel capacity is given as

$$C = \lim_{n\to\infty} \frac{\log_2 S(n)}{n} = \log_2 [1 + F(\tfrac{c-1}{\alpha})].$$

Using the rectified WRDS, the coding rule for the α-charge constrained channel with threshold $c_0 = (c-1)/\alpha$ will be the following:

$$Y_{n+1} = \begin{cases} -Y_n, & \text{if } \widehat{W}_n \geq c_0; \\ X_{m+1}Y_n, & \text{if } \widehat{W}_n < c_0. \end{cases} \tag{27}$$

Supposing that the input process is i.i.d. with $\Pr(X=+1) = p$ and $\Pr(X=-1) = q$, for the stationary distribution $G(x) = \lim_{n\to\infty} \Pr(\widehat{W}_n < x)$ we can get a functional equation similar to (26):

$$G(x) = \begin{cases} 0, & \text{if } x \leq 1-\alpha c = c'; \\ 1 - G(-\frac{x-1}{\alpha}), & \text{if } c' < x \leq 1-\alpha c_0 = 2-c; \\ p\,G(\frac{x-1}{\alpha}) + q\,[1 - G(-\frac{x-1}{\alpha})] & \\ \qquad + p\,[1 - G(\frac{c-1}{\alpha})], & \text{if } 2-c < x \leq c; \\ 1, & \text{if } x > c. \end{cases} \tag{28}$$

The second row of (28) refers to the transitions when stuffing is applied forcing to close the current run. The low values of RWRDS can be reached only this way since, as we have seen, $-c_0 < c'$, therefore the term $G(\frac{x-1}{\alpha})$ standing for continuing the current run is missing. The third row refers to the transitions when no stuffing is applied. The constant

TABLE I The most important parameters of some spectrum constrained codes

loop filter characteristics		threshold c_0	channel capacity	rate	error propagation
$\sum_{i=0}^{r} z^{-i}$	$r = 13$	2.0	0.626	0.533	11.2
		4.0	0.887	0.825	4.7
	$r = 13$, d=1, k=4	2.0	0.297	0.261	5.3
$\dfrac{1}{1-\alpha z^{-1}}$	$\alpha = 0.98$	2.0	0.853	0.808	26.2
	α=0.98, d=1, k=4	2.0	0.503	0.468	16.3
$H(z,p) = \dfrac{\alpha\cos(p\,2\pi) - \alpha^2 z^{-1}}{1-2\alpha\cos(p\,2\pi)+\alpha^2 z^{-2}}$	α= 0.98, $p = 0.2$	1.8	no data	0.813	36.3
$H(z,p_1)+H(z,p_2)+H(z,p_1)H(z,p_2)z^{-1}$	$\alpha = 0.98$, p_1= 0.19, p_2= 0.21	3.0	no data	0.597	42.4
	$\alpha = 0.99$, p_1= 0, p_2= 0.3	1.2	no data	0.684	42.5

term $\frac{1}{2}[1 - G(\frac{c-1}{\alpha})]$ provides the continuity of the distribution function.

Earlier we have showed that the rate of a bit stuff encoder is given as $1 - P_{\text{stuff}}$, so according to (27), the coder's rate is

$$R = 1 - P_{\text{stuff}} = 1 - \Pr(\widehat{W} \geq c_0) = G(c_0).$$

The α-charge constraint, as most of the constraints defined by low pass filter, constitutes an upper bound on the run-length in and of itself:

$$T_{\max} = \left\lceil \frac{\log[1-(1-\alpha)c_0] - \log[1+(1-\alpha)(1+\alpha c_0)]}{\log \alpha} \right\rceil,$$

which corresponds to the k constraint $k = T_{\max} - 1$. However, explicit RLL constraints can also be given. Neither k nor d constraint can collide with (α, c) constraint if the condition $d{+}1 < T_{\max}$ is satisfied. The spectrum of a simultaneously α-charge and (d,k) constrained sequence can be seen in Fig. 9.

B. Forming Spectral Notches

Applying a loop filter which sets bandpass characteristics to $\widetilde{H}(z)$ will result in a bandstop-like code spectrum forming a notch in the spectrum. Such code spectra are used to accommodate auxiliary information in spectrum [7], e.g., pilot tracking tone for head positioning mechanism of digital magnetic and optical recorders [2]. Applying a notch at $f_0/2$ will suppress the power around the half of the symbol rate rendering protection against band-edge filter distortion. The IIR filter

$$H(z,p) = \frac{\alpha\cos(p\,2\pi) - \alpha^2 z^{-1}}{1 - 2\alpha\cos(p\,2\pi)\,z^{-1} + \alpha^2 z^{-2}}$$

it will generate a spectral notch at pf_0 (Fig. 10). It behaves similarly to the characteristics by the low-pass IIR filter, that is, the greater α is chosen, the deeper and steeper suppression will be, with a higher error propagation susceptibility. Placing another pole in $\widetilde{H}(z)$ next to the first one, we can enhance the width of the spectral notch without diminishing the suppression (Fig. 10):

$$H(z,p_1,p_2) = H(z,p_1) + H(z,p_2) + H(z,p_1)H(z,p_2)z^{-1} \quad (29)$$

Letting $p_1 = 0$ in (29), the first pole will appear in dc. It combines the spectral notch with a dc-suppressed code spectrum (Fig. 10), which is often required.

In Table I we have collected the most important parameters of spectrum constrained codes discussed in this paper. The error propagation is defined as the average number of false detections (false removals or remanent stuffed bits) induced by a single error. It has been measured at a BER of 10^{-4}.

V. CONCLUSION

In this paper we have generalized the accumulated charge concept and introduced a new class of constraints the generalized charge constraint. With the new constraint the spectral requirements can be described easily in the time domain. A feedback controlled bit stuff encoder with loop filter is suggested to implement the new constraint. We have studied the performance of the new coder structure and demonstrated its spectral shaping property. We have presented a few spectral characteristics of practical interest generated by low- and bandpass loop filters.

ACKNOWLEDGMENT

I wish to thank Peter Tatai who has first called my attention to sigma-delta converters and László Osváth for his insightful discussions and comments, and for his contribution to the proof of Theorem on stationary distribution of window-charge constrained code. I also wish to express my thanks to Géza Gordos and Gyula Sallai, who supported my work, and to Rudy Kalman, who gave me the occasion to talk over my ideas.

REFERENCES

[1] H. Kobayashi, "A survey of coding schemes for transmission or recording of digital data," *IEEE Trans. Commun. Technol.*, vol. COM-19, pp. 1067–1100, Dec. 1971.

[2] K.A.S. Immink, *Coding Techniques for Digital Recorders.* London: Prentice Hall, 1991.

[3] Y.L. Hsueh, M.S. Rogge, et al., "Smooth upgrade of existing passive optical networks with spectral-shaping line-coding service overlay," *IEEE/OSA Journal of Lightwave Technology*, vol. 23, No. 5, pp. 2629–2637, Sept. 2005.

[4] D.T. Tang and L.R. Bahl, "Block codes for a class of constrained noiseless channels," *Information and Control*, vol. 17, pp. 436–461, 1970.

[5] B.H. Marcus and P.H. Siegel, "On codes with spectral nulls at rational submultiples of the symbol frequency," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 557–568, July 1987.

[6] I.J. Fair, W.D. Grover, W.A. Krzymien and R.I. McDonald, "Guided scrambling: A new line coding technique for high bit rate fiber optic transmission systems," *IEEE Trans. Commun.*, vol. COM-39, pp. 289–236, Feb. 1991.

[7] J.C. Cavers and R.F. Marchetto, "A new coding technique for spectral shaping of data," *IEEE Trans. Commun.*, vol. COM-40, pp. 1418–1422, Sept. 1992.

[8] ISO/IEC 3309 standard.

[9] P.E. Bender and J.K. Wolf, "A universal algorithm for generating optimal and nearly optimal run-length-limited charge-constrained binary sequences," in *Proc. IEEE Symp. on Inform. Theory*, San Antonio, Texas, Jan. 1993, p. 6.

[10] S. Aviran, P.H. Siegel and J.K. Wolf, "An improvement to the bit stuffing algorithm," *IEEE Trans. Inform. Theory*, vol. IT-51, pp. 2885–2891, August 2005.

[11] Y. Sankarasubramaniam and S.W. McLaughlin, "Capacity achieving code constructions for two classes of (d, k) constraints," *IEEE Trans. Inform. Theory*, vol. IT-52, pp. 3333–3343, July 2006.

[12] S. Aviran, P.H. Siegel and J.K. Wolf, "Optimal parsing trees for run-length coding of biased data," *IEEE Trans. Inform. Theory*, vol. IT-54, pp. 841–849, Feb. 2008.

[13] P. Vámos, "A coding theorem of runlength limited channel," in *Proceedings of IEEE International Symposium on Information Theory*, Washington, DC, June 24-29, 2001, p.66.

[14] G.L. Pierobon, "Codes for zero spectral density at zero frequency," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 435–439, March 1984.

[15] P. Vámos and L. Osváth, "The error propagation of bit-stuff coding systems," in *Proceedings of IEEE Symp. on Inform. Theory*, Seattle, WA, July 2006.

[16] W.G. Bliss, "Circuitry for performing error correction calculations on baseband encoded data to eliminate error propagation," *IBM Techn. Discl. Bul.*, vol. 23, No. 10, pp. 4633–4634, March 1981.

[17] K.A.S. Immink, "Code configuration for avoiding error propagation," *Electronics Letters*, vol. 32, pp. 2191–2192, Nov. 1996.

[18] R.M. Gray, "Quantization noise spectra," *IEEE Trans. Inform. Theory*, vol. IT-36, pp. 1220–1244, Nov. 1990.

[19] H.A. Spang and P.M. Schulthess, "Reduction of quantizing noise by use of feedback," *IRE Trans. Commun. Systems*, vol. COM-10, pp. 373–380, Dec. 1962.

[20] H. Inose and Y. Yasuda, "A unity bit coding method by negative feedback," *Proc. IEEE*, vol. 51, pp. 1524–1535, Nov. 1963.

[21] W.H. Kautz, "Fibonacci codes for synchronization control," *IEEE Trans. Inform. Theory*, vol. IT-11, pp. 284–292, April 1965.

[22] A.R. Collar, "On centrosymmetric and centroskew matrices," *Quart. Journ. Mech. and Appl. Math.*, vol. 15, pp. 265–281, 1962.

[23] C.E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pt. I, pp. 379–423, 1948; pt. II, pp. 623–656, 1948.

[24] C.D. Heegard, B.H. Marcus and P.H. Siegel, "Variable-length state splitting with applications to average runlength-constrained (ARC) codes," *IEEE Trans. Inform. Theory*, vol. IT-37, pp. 759–777, May 1991.

[25] K.J. Kerpez, A.Gallopoulos and C.D.Heegard, "Maximum entropy charge-constrained run-length codes," *IEEE Journ. Sel. Areas Commun.*, vol. 10, pp. 242–253, Jan. 1992.

[26] A.Gallopoulos, C.D.Heegard and P.H. Siegel, "The power spectrum of run-length-limited codes," *IEEE Trans. Commun.*, vol. COM-37, pp. 906–917, Sept. 1989.

[27] K.A.S. Immink, "Performance of simple binary dc-constrained codes," *Philips J. Res.*, vol. 40, No. 1, pp. 1–21, 1985.

[28] A.X. Widmer and P.A. Franaszek, "Transmission code for high-speed fiber-optic data networks," *Electronics Letters*, vol. 19, pp. 202–203, March 1983.

[29] F.R. Gantmacher, *Theory of Matrices.* New York: Chelsea Publishing Co., 1960.

Techniques for Modeling Mobile Application Spreading

Ádám Horváth, and Károly Farkas, *Member, IEEE*

***Abstract*—While the number of mobile devices and applications increases considerably, application spreading by using direct communication between mobile devices has not got too much attention so far. However, exploiting the advantages of modern communication paradigms has even economic relevance. For instance, direct communication in ad hoc networks can be used to advertise and spread applications, which influences the number of purchases what application providers may want to estimate.**

In this paper, we present two main techniques for modeling application spreading in this environment. First, we model the spreading process using Closed Queuing Networks, assuming a single type user behavior. Then, to capture different user behaviors, we show how to model application spreading by Stochastic Petri Nets. We define a basic Petri net model that we analyze using a mean field based methodology. Unfortunately, introducing more realistic user behaviors makes the analytical handling of the Petri net too complex, or even impossible. We show an example of this case extending the basic Petri net model with an additional, more realistic user behavior, and investigate this extended model via simulations. Moreover, we compare the investigation results of the different models and point out their relations.

***Index Terms*—Application Spreading, Closed Queuing Networks, Mathematical Modeling, Stochastic Petri Nets, Mean Field Based Methodology**

I. INTRODUCTION

With the proliferation of mobile devices, the number of mobile applications is increasing significantly. However, mobile application spreading is not a frequently investigated research topic today. Being aware of the characteristics of application spreading is important for the application provider not only from technical, but also from economic point of view. The application provider has to know or at least assess how much money he can earn from the purchases of a given application; how much time is needed to realize it; and which factors influence the spreading process and how.

Traditionally, mobile applications are spread via a central entity, like an Internet webshop. Users can browse the website of the merchant, select, purchase and download the application they like. However, decentralized technologies such as self-organized or mobile ad hoc networks allow the users to get the application software directly from each other. Hence, direct application download can change the characteristics of traditional application spreading. The participants of this direct communication can even try out the applications and be motivated to purchase the ones they liked (purchasing is available only via a traditional way, because secure payment in this environment is still a challenging issue today).

In this type of communication, a lot of factors can change the characteristics of the spreading process, especially from economic viewpoint, which have not taken into consideration yet (e.g., community experience when playing a multi-player game). These factors can give more motivation to the users to purchase the application than they would have seen only some advertisements.

In this paper, we present two modeling techniques by which we can investigate application spreading in mobile environment. First, we propose the use of Closed Queuing Networks (CQNs) [1] for modeling the application spreading process. Assuming a single, homogeneous user behavior, we can simply and quickly obtain the expected number of purchases by using a CQN.

Then, we propose Stochastic Petri Nets (SPNs) [2] based modeling to capture more sophisticated user behaviors. We define first a basic SPN model assuming different user types. We analyze this model using a fluid approximation method, originally presented in [3]. In this method, we transform the SPN into ordinary differential equations (ODEs), which can be evaluated quickly even assuming a huge user population. Unfortunately, if we want to include more realistic user behaviors in the model, the analytical handling of the Petri net can become too complex, or even impossible. We show an example of this case extending the basic Petri net model with an additional, more realistic user behavior, and investigate this extended model via transient simulation. However, we have to keep in mind that simulating the transient behavior of Petri nets with large state space is much slower and performance intensive task than analytical evaluation.

Finally, we compare the investigation results of the different models and point out their relations.

The rest of the paper is organized as follows. In Section II, we present a short overview of the related works. In Section III, we describe the communication model and define the user behavior types. We present two techniques to investigate the application spreading process, thus our CQN and SPN models together with some results in Section IV and Section V, respectively. Finally, we compare the results and conclude the paper in Section VI.

Manuscript received February 29, revised April 23, accepted for publication April 27, 2012.

Á. Horváth is with the Institute of Informatics and Economics, University of West Hungary (UWH), Sopron, Hungary (corresponding author, phone: +36-99-518-606; fax: +36-99-518-367; e-mail: horvath@inf.nyme.hu).

K. Farkas is with the Department of Telecommunications, Budapest University of Technology and Economics (BME), Budapest, Hungary (e-mail: farkask@hit.bme.hu).

II. RELATED WORKS

With the proliferation of modern communication paradigms, the investigation of application spreading using new ways becomes more and more important. However, it has not got too much attention so far. Besides our previous contributions [4]-[6], only a few papers touch even the commercial use of ad hoc networks and direct communication.

On the other hand, epidemic spreading is a popular research topic today and this area is similar to our context. In [7], the authors present a model, by which they investigate the propagation of a virus in a real network. In [8], the authors present scale-free networks for modeling the spreading of computer viruses and also give an epidemic threshold, which is an infection rate. Information spreading is also investigated by using epidemic spreading models, such as the susceptible-infected-resistant (SIR) model [9], or other models based on the network topology [10], [11]. In [12], malicious software spreading over mobile ad hoc networks is investigated. The authors propose the use of the susceptible-infected-susceptible (SIS) model based on the theory of Closed Queuing Networks.

In [13], the authors propose the commercial use of ad hoc networks and present a radio dispatch system using mobile ad hoc communication. In the proposed system, the connectivity of the nodes is the key element of information dissemination. In our models, we do not consider the network topology as a key element of application spreading, since no real-time information dissemination is needed between the users. For the same reason, we do not deal with mobility models such as random walk model, which are well presented in many contributions [14]-[16].

Although the above mentioned proposals show some similarities with our work, none of them deals with application spreading and, except [13], they do not touch the commercial benefits of direct communication. Moreover, the authors in [13] consider information dissemination as a tool, and not as a goal.

III. COMMUNICATION MODEL AND USER TYPES

In this section, we present the communication model which we use in our investigations. Moreover, we introduce three user types based on different user behaviors.

A. Communication Model

We refer to the individuals who are interested in the use of the application as users. The population that we investigate is composed of users only, and we do not take uninterested users into consideration, because they do not influence the spreading process. Therefore, we assume a closed user population.

We investigate the spreading of a given multi-user application having two versions, a trial and a full version. The users can be categorized into different classes depending on whether they do possess any version of the given application or do not. We named the classes after the terminology of epidemics, since our model shows similarity to the epidemic spreading models. A user is called (1) infected, if he has got the full version of the application; (2) susceptible, if he possesses only the trial version of the application; and (3) resistant, if he has got none of them, or he has already lost the interest of using the application.

Users with their mobile devices form self-organized networks from time to time, in which direct communication takes place. The trial version of the application is free and available in these networks, so users can download it and even try it out. However, it has some restrictions (see later), so the users have to purchase the application for unrestricted usage via a traditional way of purchasing. Later, also these users can spread the trial version of the purchased application further.

Since we want susceptible users to be motivated in purchasing the full version of the application, some limitations must be made in using the trial version. Therefore, we apply a limit (leech[1] limit) that restricts how many nodes possessing the trial version (leech) can connect to a node possessing the full version (seed[1]). In this sense, the seeds can be considered as servers, which can serve a limited number of clients. A seed is always an infected user, while a leech may be either infected or susceptible. Fig. 1 depicts the case when a susceptible user purchases the application.

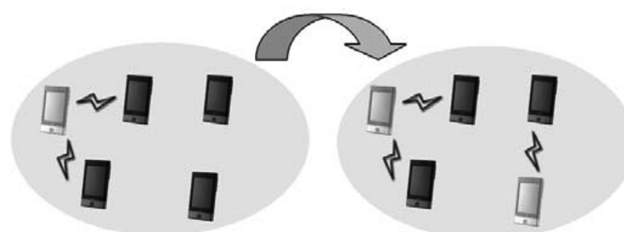


Fig. 1. Change of application usage when a susceptible user purchases the application.

The devices form an ad hoc network, in which the dark devices depict susceptible users, while the light ones depict infected users. In this example, the leech limit is two, so two susceptible users can peer to the only infected user, while the other two have to wait (the connection symbol represents application level peering). After one of them purchased the application, they can also use it, as shown in the right side of Fig. 1.

B. User Types

Beyond the basic communications, we distinguish three different user types based on the users' behavior. $Type_A$ users are interested in using the given application, therefore, they are its potential buyers even without trying it out. $Type_B$ users are motivated in purchasing the application only if they realize that the application is popular in their environment. $Type_C$ users also purchase the application very likely, but they will do it with a given intensity, only if they cannot find a seed from time to time which they can connect to.

In our CQN model, we use only $Type_A$ users, since we additionally introduce the other user behaviors in our SPN models, namely $Type_B$ users in the basic and $Type_B$, $Type_C$ users in the extended SPN model. Of course, additional user types can be introduced, too. However, the more user types we capture the more complex model we get, which can make the handling of the model difficult.

[1] After the terminology of BitTorrent [17].

IV. MODELING WITH CLOSED QUEUING NETWORKS

In self-organized networks, where spontaneous communication takes place, the network topology can change rapidly due to the high degree of mobility. These topology changes can be modeled by stochastic processes [18]. We can appropriately describe a stochastic process in a closed population, which is interesting from our point of view, using Closed Queuing Networks [1]. Moreover, ordering transition intensities to the state changes we can capture the time behavior of the application spreading process, too.

A. Spreading Model

Assuming a homogeneous user population with simple user behavior (only Type_A users) we propose the CQN depicted in Fig. 2 to model the application spreading process. CQN models are not appropriate to handle complex conditions, but they can be used as a first approach in simple situations providing quick results.

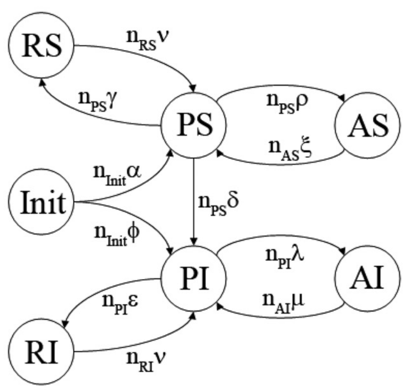


Fig. 2. The proposed CQN model.

The users are represented by the different model states in Fig. 2. Each user is in a given state depending on his current user class. The resistant users are in state *Init*. They possess neither the trial nor the full version of the application. We call also resistant the users, who have already lost the interest in using the application, however, they possess either its trial (state *RS*) or its full version (state *RI*). The susceptible users are in state *PS* and *AS*, depending on that they are currently using the application (Active Susceptibles, *AS*) or not (Passive Susceptibles, *PS*). Similarly, the active and passive infected users are in state *PI* and *AI*, respectively.

The Greek letters in Fig. 2 denote transition intensities regarding to a single user. The transition intensity is a real number illustrating how many times a transition expectedly takes place in a given time interval. n_x represents the number of users in state x, so the transition intensity of a given state transition is proportional to the number of users in the source state. The state transitions of the model are described in Table I.

B. Usage of the Spreading Model

We can unambiguously describe the state of the system with the user distribution (n_{Init}, n_{PS}, n_{AS}, n_{PI}, n_{AI}, n_{RS}, n_{RI}). The transition intensities (α, γ, δ, ε, ϕ, λ, μ, ν, ρ and ξ) regarding to a single user are the system parameters, which are hard to be determined theoretically. In this paper, we set the system parameters based on common sense. The parameter setting can be fine-tuned experimentally, what is beyond the scope of this paper.

TABLE I

STATE TRANSITIONS OF THE PROPOSED CQN MODEL

Transitions	Description
PS → AS	A susceptible user starts to run the application and tries to connect to a seed in the network. If he cannot find one, he has to wait.
AS → PS	A susceptible user stops running the application.
PI → AI	An infected user starts to run the application, then either he tries to connect to a seed, or will be a seed himself to which leeches can connect.
AI → PI	An infected user stops running the application.
Init → PS	A resistant user downloads the trial version of the application.
PI → RI	An infected user becomes resistant losing the interest in using the application.
PS → RS	A susceptible user becomes resistant losing the interest in using the application.
Init → PI	A resistant user purchased the application without trying it out.
PS → PI	A susceptible user becomes infected by purchasing the application.
RS → PS	It is possible that a resistant user, who lost the interest in using the trial version, wants to use the application again after a while. If so, his state becomes susceptible again.
RI → PI	Similarly, if a resistant user possessing the full version of the application wants to use it again, his state changes to infected.

In each system state, we can generate the holding time h (the time that the system is expected to spend in a given system state) as an exponentially distributed random variable[2] in the following way:

$$h = \frac{-lnRND}{\sum_{\forall state\ x} out_x} \quad (1)$$

where $0 \leq RND < 1$ is a pseudo-random number, and out_x denotes the sum of the intensities for each transition with source state x. We must compute h after each state change, because the user distribution changes when a transition takes place. We can generate the next system state based on the ratio of the current transition values. After we generated the transition that takes place, we move one user from its source to its destination state, and compute the holding time of the new system state, and so on.

At the beginning, each user is in state *Init*, which is the initial state of the system. The users will leave this state and change their states from time to time. After a while, a user will lose the interest in the application usage (reaches state *RS* or *RI*), but it does not mean that he cannot be interested again later on. Thus, the state transitions *RS* → *PS* and *RI* → *PI* are also enabled, however, we allow them only with low intensity values. Therefore, we will reach a system state (final system state) sooner or later, in which each user is either in state *RS* or

[2] Using exponentially distributed holding times is a usual modeling simplification in this context [12], [19].

PS. The final system state is not the steady state, however, our investigation will stop here. Taking into account the asymmetry of this system (if someone purchases the application, he will never lose it), each user will be in one of state *RI*, *PI* or *AI* by reaching the steady state (even product-form solution exists). However, it is meaningless to consider this state in the model investigation, since after reaching the final system state the holding times become extremely large, so the system changes very slowly. Hence, our investigations are always transient and consider only a time period which is interesting from the merchant's point of view.

We can determine how many pieces of the application were sold, upon reaching the final state, by summing the number of $PS \rightarrow PI$ and $Init \rightarrow PI$ transitions. Running simulations and evaluating the results, we can see the time characteristics of the spreading process, too.

C. *Results Derived from CQN*

In this section, we recall some analytical results from [5] and validate them via simulations. In contrast to [5], we use a single user type in this model, and apply such a parameter setting which makes possible to compare the results to the SPN models.

Analytical solutions of CQNs can be obtained, e.g., with the well-known Mean Value Analysis (MVA) [20]. However, we investigate the transient behavior of the system, in which case this method is not feasible.

After a while, each user will leave the initial state, since we do not take the uninterested individuals into consideration. Based on the intensity value of transition $Init \rightarrow PI$ and $Init \rightarrow PS$, we can determine how many users will expectedly purchase the application without trying it out (direct purchases, *DP*) in the following way:

$$DP = n_{Init} \cdot \frac{\phi}{\alpha + \phi} \tag{2}$$

where n_{Init} denotes the initial number of users in state *Init*, i.e., the total population size. All nodes (users) that did not purchase the application without trying it out will change their state to susceptible. Susceptible states are state *PS* and *AS*, and there are two possibilities for the users to leave these states: a user can become either (1) resistant (transition $PS \rightarrow RS$); or (2) infected (transition $PS \rightarrow PI$). Moreover, it is also allowed to return from state *RS*, but the intensity of transition $RS \rightarrow PS$ is very low, since losing the interest and being interested again after a while is not a typical user behavior. Therefore, we can estimate the number of indirect purchases (purchase after trying out) based on the ratio of transition $PS \rightarrow PI$ and $PS \rightarrow RS$ as follows:

$$IDP = n_{Init} \cdot \frac{\alpha}{\alpha + \phi} \cdot \frac{\delta}{\delta + \gamma} \tag{3}$$

The expected value of total purchases (*TP*) can be obtained by summing (2) and (3).

To validate the analytical results we ran simulations. Table II compares the analytical results to the average of

TABLE II
COMPARISON OF THE ANALYTICAL AND SIMULATION RESULTS USING THE CQN MODEL

Description	Analytically	By simulation
DP	4.95	4.95
IDP	165.02	164.99
TP	169.97	169.94

10000 individual simulation runs with the following parameters: $n_{Init} = 500$, $\alpha = 10^{-3}$, $\gamma = 210^{-3}$, $\delta = 10^{-3}$, $\varphi = 10^{-5}$. The other parameters do not influence *DP* and *IDP*, however, they have an effect on the time behavior of the spreading process. This can be investigated also via simulations.

V. MODELING WITH STOCHASTIC PETRI NETS

In this section, after introducing the fundamentals of Stochastic Petri Nets [2], to be able to handle also Type_B and Type_C users we present the description and the analysis of our two SPN models, the basic and the extended one.

Since Type_A users are present in all of our models (including the CQN model), we can compare the models from the viewpoint of this simple user behavior, see Section VI.

A. *SPN Formalism*

The continuous-time Stochastic Petri Net can be defined as a 6-tuple

$$(P, T, I, O, \lambda, M_0)$$

where $P = \{p_i\}$ is the set of *places*; $T = \{t_i\}$ is the set of *transitions*; $I, O : T \times P \rightarrow \mathbb{N}$ are the *input* and *output* functions that define the arcs of the net with their multiplicities; $\lambda : T \rightarrow \mathbb{R}$ are the functions that assign *firing intensities* to each transition and M_0 is the *initial marking* of the net.

A transition t is enabled in marking M if $M(p) \geq I(t, p)$ holds for all places p. In other words, a transition is enabled if each of its input places contains at least the amount of tokens defined by the input function *I*. Only enabled transitions can fire, and if one does, we remove tokens from the input places and add tokens to the output places of the firing transition. Formally, the new marking M' is given as $M'(p) = M(p) + O(t, p) - I(t, p)$ for all $p \in P$.

The transitions fire after a random delay that can be described with an exponentially distributed random variable. Its parameter depends on the given transition's firing intensity and the current marking. Our model uses the infinite server approach, thus the firing intensity is increasing with the increase of the tokens' number in the enabling places. This concept is formally captured by the definition of enabling degree. Namely, the enabling degree $ed(t, M)$ of a transition t in the marking M is d iff $\forall p \in P$, $M(p) \geq dI(t, p)$ and $\exists p \in P : M(p) < (d + 1)\, I(t, p)$.

For further details on SPNs, see [2] or [21].

B. *Basic SPN Model*

Here we describe our basic SPN model, and present a mean field based methodology for analyzing SPNs. Then using this methodology, we show the transient analysis of our basic SPN model.

Model Description

Our basic SPN model is depicted in Fig. 3. The rectangles illustrate the transitions of the SPN, while the circles represent the places. We assume the presence of 500 Type$_A$ and 500 Type$_B$ users, and initially each user is in the passive susceptible states *PASS_A* (Type$_A$ users) and *PASS_B* (Type$_B$ users). Seeds can also be passive (*PASS_S*), however, there are no seeds in the network initially. Similarly to the CQN model, we call a user active if he is currently using the application. We keep count of the number of active Type$_A$ users (*ACT_A*), Type$_B$ users (*ACT_B*), available seeds (*FREE_S*) and the total number of active users, including seeds (*ACT_USERS*). Since the leech limit is one, the available seeds show how many susceptible users can start the application in a given time. We also keep count of the number of application purchases either by Type$_A$ users (*PURCHASES_A*) or by Type$_B$ users (*PURCHASES_B*). After a while, users lose the interest in using the application in this model, as well. If so, they change their state to resistant (*LOST_INT_A*, *LOST_INT_B*, *LOST_INT_S*).

The transitions of the model and their values regarding to one user with which we analyzed the net are described in Table III.

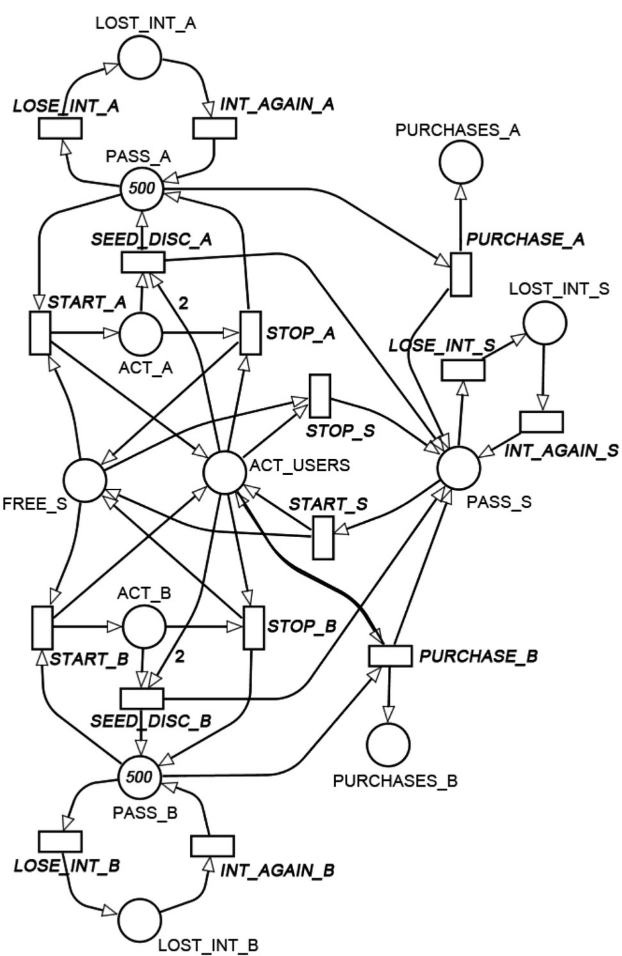


Fig. 3. The basic SPN model.

TABLE III
STATE TRANSITIONS OF THE BASIC SPN MODEL

Transitions	Description
START_A / START_B / START_S	A Type$_A$ user / Type$_B$ user / seed starts to run the application. They can fire only if there is at least one available seed in the network ($4 \cdot 10^{-2}$ in each case).
STOP_A / STOP_B / STOP_S	A Type$_A$ user / Type$_B$ user / free seed stops running the application ($9 \cdot 10^{-1}$ in each case).
SEED_DISC_A / SEED_DISC_B	A seed to which a Type$_A$ / Type$_B$ user had been connected stopped running the application. The state of the connected leech node becomes passive ($9 \cdot 10^{-1}$ in each case).
LOSE_INT_A / LOSE_INT_B / LOSE_INT_S	A Type$_A$ user / Type$_B$ user / seed loses the interest in using the application (10^{-3} in case of a seed, $2 \cdot 10^{-3}$ otherwise).
PURCHASE_A / PURCHASE_B	A Type$_A$ user / Type$_B$ user purchases the application (10^{-3} in each case).
INT_AGAIN_A / INT_AGAIN_B / INT_AGAIN_S	It is possible that a resistant user, who lost the interest in using the application, wants to use the application again after a while. If so, his state becomes susceptible or infected again, depending on his previous state (10^{-5} in each case).

Our basic SPN model does not contain inhibitor arcs, thus we can apply the mean field based methodology in its analysis. If we have to build a more complex model enabling also the use of inhibitor arcs, its analytical handling is not possible anymore, so we must run simulations to investigate the model's behavior (see Section V.C).

Mean Field Based Methodology

The standard approach for analyzing SPNs is to construct the continuous time Markov chain (CTMC, for details see, e.g., [22]) corresponding to the underlying stochastic behavior of the SPN and perform the steady state or transient analysis analytically [23] or by simulation. However, this approach becomes unfeasible due to the size of the state space if we consider a network composed of a large number of mobile components.

In the following, we describe the mean field approach, which is a fluid approximation method for model evaluation. Applying this method, the analysis will terminate within a few seconds, even when the state space explodes due to the high number of tokens. The following definition and theorem are based on [3], while we presented it in [6] in a form that is directly related to the applied definition of SPN. In that paper, we provided a formal relation between the CTMC and its fluid approximation, too.

Definition: A parametric family of Markov chains, $X_\upsilon(t)$ with $\upsilon \in \mathbb{N}$, with state spaces $E_\upsilon \subset \mathbb{Z}^k$, is called density dependent if and only if there exists a continuous function $f(x, l)$, $x \in \mathbb{R}^k$, $l \in \{L(t_1), \ldots, L(t_m)\}$, such that the non-diagonal entries of the infinitesimal generator corresponding to $X_\upsilon(t)$ can be written in the following form:

$$q_{k,k+l} = \upsilon f\left(\frac{k}{\upsilon}, l\right), l \in \{L(t_1), \ldots, L(t_m)\} \quad (4)$$

and the initial state of the chain is υx_0, $x_0 \in \mathbb{Z}^k$, with probability 1. Let $X(t)$ denote the solution of the ODEs:

$$\frac{dX(t)}{dt} = \sum_{l\in\{L(t_1),\ldots,L(t_m)\}} f\,(X(t), l) \quad (5)$$

with initial condition $X(0) = x_0$.

Theorem: Under mild conditions of function f (for details see [3]), the following relation holds between function $X(t)$ and a trajectory of the CTMC $X_\upsilon(t)$:

$$\forall\delta > 0: \lim_{\upsilon\to\infty} P\left\{sup_{s\le t}\left|\frac{1}{\upsilon}X_\upsilon(s) - X(s)\right| > \delta\right\} = 0 \quad (6)$$

The interpretation of (6) is the following. Consider a CTMC modeling the interaction of k quantities with $\mathbb{Z}^k$ state space. If we observe a sequence of CTMCs with increasing initial state, and this increase gives rise to a sequence of infinitesimal generators corresponding to the form in (4), then as υ is increased, the behavior of the CTMC converges to the solution of the ODEs in (5). It means that the probability of finding any difference between the trajectory of the CTMC and the solution of the ODEs in a finite time horizon (0, t) is zero.

It has already been shown in [3], that for large population sizes, the corresponding ODEs provide a good approximation of the system's behavior in case of density dependent CTMCs. It is straightforward to show that the basic Petri net we use for modeling application spreading is density dependent, therefore, we can approximate the behavior of this Petri net with high number of tokens by solving the ODEs. Moreover, we presented in [6] that the approximation works even with a lower number of tokens.

Transient Analysis

In the following, we illustrate the usage of the mean field approach through the transient analysis of the above mentioned basic SPN model.

The solution of the ODEs is a good approximation of the average behavior of the model. Fig. 4 depicts the cumulative expected value of the number of Type_A and Type_B users' purchases after a given time, while Fig. 5 shows the cumulative expected value of the number of users who lost the interest in using the application as time elapses.

In Fig. 4 and Fig. 5, we can see that mostly Type_A users purchased the application, and the interest in using the application is approximately limited to the first 4000 hours. However, the parameter settings are critical, since the number of Type_B users' purchases depends on the activity of the users. The higher the activity is, the more Type_B users will purchase the application.

C. *Extended SPN Model*

In this section, we describe our extended SPN model and its transient simulation.

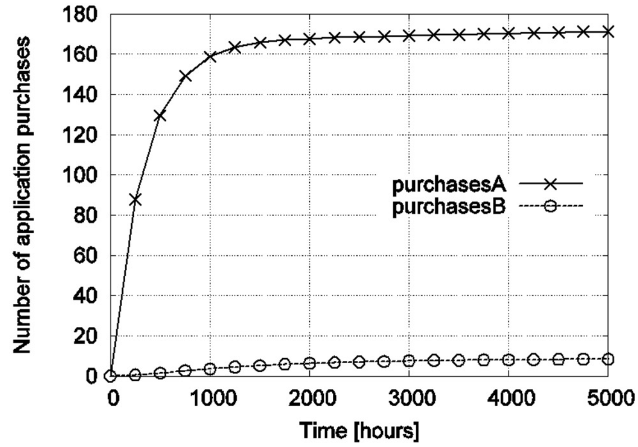


Fig. 4. Expected value of the number of application purchases as the function of elapsed time.

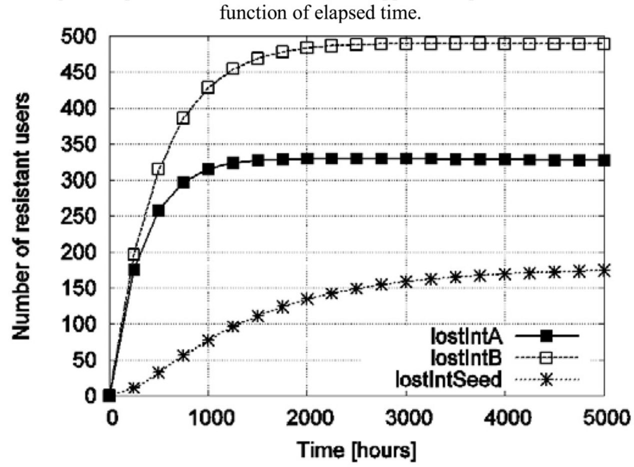


Fig. 5. Expected value of the number of users who lost the interest in using the application.

Model Description

In our extended SPN model, we can also handle more sophisticated user behaviors, like the third user type (Type_C). This requires the use of inhibitor arcs, too.

Apart from this, the extended model is very similar to the previous one. To handle the new user type we added four places (*PASS_C*, *ACT_C*, *LOST_INT_C* and *PURCHASES_C*) and six transitions (*LOSE_INT_C*, *INT_AGAIN_C*, *PURCHASE_C*, *SEED_DISC_C*, *START_C* and *STOP_C*) to the basic model. As we described in Section III.B, Type_C users purchase the application only if they cannot find an available seed which they can connect to. Therefore, the transition *PURCHASE_C* is enabled only if there is no token in place *FREE_S*. This relationship is denoted in the model by an inhibitor arc with a circle on its head. Beyond that, Type_C users' behavior is similar to the others'.

Fig. 6 shows our extended SPN model. The newly added part is marked by grey background. As earlier, we set 500 users from each user type in the initial marking.

Transient Simulation

Since our extended SPN model contains also inhibitor arc, we cannot use the fluid approximation method to investigate the model's behavior, rather we have to run simulations. There exist many tools for modeling with SPNs, e.g., the ones

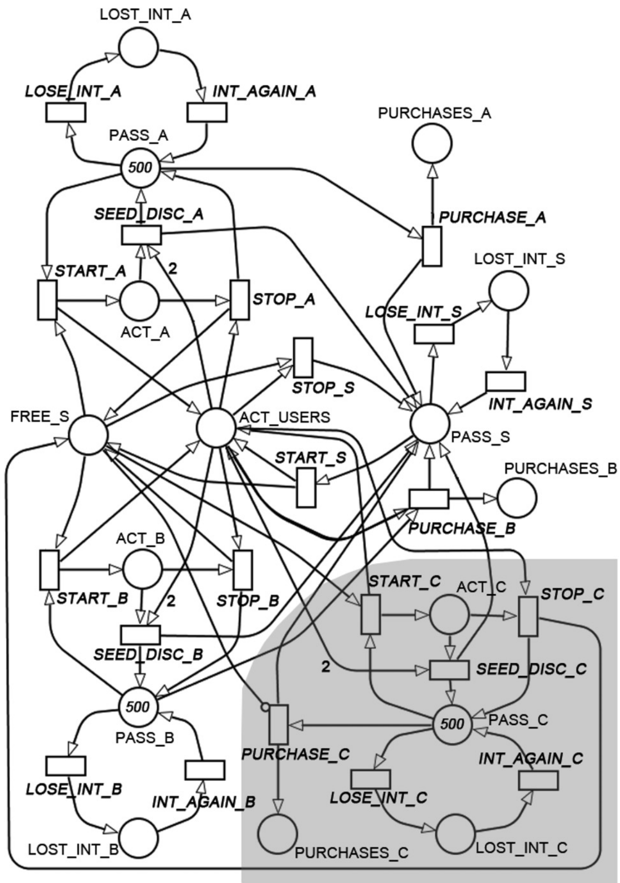


Fig. 6. The extended SPN model.

presented in [24]-[26]. We used the transient simulation method of TimeNET [24].

In our investigations, we used the same transition intensities for the existing transitions as in Section V.B, while the new transitions were set to the corresponding transition intensity values of the previous model. For example, transition *START_C* has the same intensity value as *START_A* and *START_B*.

The simulation results reflect 95% confidence level and 5% maximal error rating. Fig. 7 shows the cumulative expected value of the number of application purchases with regard to the different user types, while Fig. 8 depicts the cumulative expected value of the number of users who lost the interest in using the application as time elapses.

These results show similarity to the previous investigation, namely, approximately one third of the $Type_A$ users and 2% of the $Type_B$ users purchased the application. On the other hand, the $Type_C$ users' purchase depends on the number of free seeds in the network (and certainly on the initial parameter setting). According to our simulation results, 5% of the $Type_C$ users purchased the application, which ratio is much higher than the purchases of $Type_B$ users.

VI. Conclusion

In this paper, we presented two techniques for modeling application spreading aided by direct communication between the users' mobile devices.

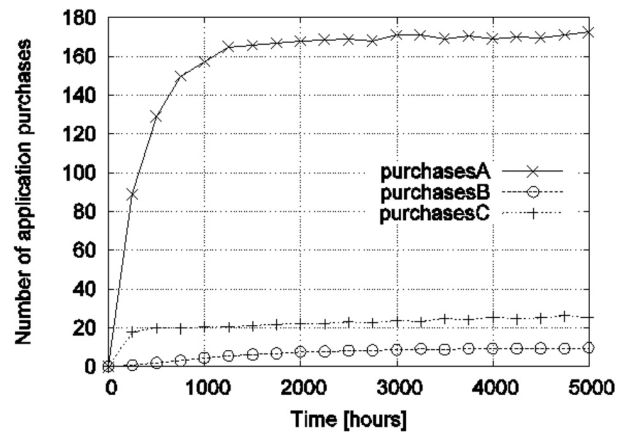


Fig. 7. Expected value of the number of application purchases as the function of elapsed time.

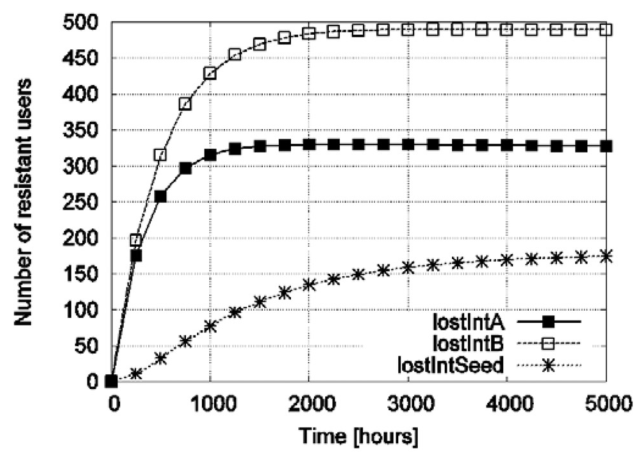


Fig. 8. Expected value of the number of users who lost the interest in using the application.

First, we presented our CQN model by which we can investigate the application spreading process assuming a homogeneous user behavior. This simple model allows a service provider to quickly calculate how much profit can be expectedly realized from application purchases.

Overcoming the limits of CQNs and being able to handle more complex user behaviors, we proposed two SPN models. In the basic one, we used the mean field based methodology to obtain an analytical approximation of the Petri net, which can derive results in the order of seconds. Then we presented an extended version of this basic Petri net that can accommodate an even more realistic user behavior for the price of using inhibitor arcs in the model. Unfortunately, the mean field approach cannot handle inhibitor arcs. Therefore, we investigated the extended Petri net model via simulations paying the fee of long runtime to produce results.

As we set the initial number of users from a given user type to the same (500) in every investigation, we can compare the results regarding to $Type_A$ and $Type_B$ users in the different models. The expected value of the number of $Type_A$ users' purchases was approximately the same in all models (169.97, 171.43 and 172.49), while this value of $Type_B$ users' purchases was also close to each other in the SPN models

(8.37 and 9.61). Therefore, we can consider the SPN models as extensions of the CQN model for cases when we have to handle more sophisticated user behaviors. On the other hand, we have to keep in mind that analytical results can be derived in a much faster way than producing results via running simulations, which can have also influence on selecting the model to be used.

In our models, with appropriate experience to set the model parameters a service provider can estimate his profit from application purchases in case of simple or even more complex scenarios. Moreover, he can learn the time behavior of the spreading process, by which he can realize additional gain, such as refining his marketing strategy.

Since the parameter setting is critical in this work, we plan to compare the outcome of our models to real data and refine the models accordingly as a future work.

References

[1] T. G. Robertazzi, "Computer networks and systems: Queuing theory and performance evaluation," *New York: Springer-Verlag*, 1994.

[2] G. Balbo, G. Conte, S. Donatelli, and S. Franceschinis, "Modelling with generalized stochastic Petri nets," *John Wiley & Sons*, 1995.

[3] T. G. Kurtz "Solutions of ordinary differential equations as limits of pure jump Markov processes," *Journal of Applied Probability*, vol. 1, no. 7, pp. 49–58, 1970.

[4] Á. Horváth, and K. Farkas, "Modeling Self-Organized Application Spreading," *In Proc. of the 5th International ICST Conference on Access Networks (ACCESSNETS 2010)*, Budapest, Hungary, 2010.

[5] Á. Horváth, and K. Farkas, "Modeling Application Spreading using Mobile Ad Hoc Networks," *In Proc. of the Third Joint IFIP Wireless and Mobile Networking Conference (WMNC)*, Budapest, Hungary, 2010.

[6] M. Beccuti, M. De Pierro, A. Horváth, Á. Horváth, K. Farkas, "A Mean Field Based Methodology for Modeling Mobility in Ad Hoc Networks," *In Proc. of the 73rd IEEE Vehicular Technology Conference (VTC2011-Spring)*, Budapest, Hungary, 2011.

[7] Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos, "Epidemic Spreading in Real Networks: An Eigenvalue Viewpoint," *22nd International Symposium on Reliable Distributed Systems (SRDS03)*, pp. 25-34, Florence, Italy, 2003.

[8] R. Pastor-Satorras, and A. Vespignani, "Epidemic Spreading in Scale-Free Networks," *Phys. Rev. Lett.*, 86:3200, 2001.

[9] F. Fu, L. Liu, and L. Wang, "Information Propagation in a Novel Hierarchical Network," *46th IEEE Conference on Decision and Control*, New Orleans, USA, 2007.

[10] A. Khelil, C. Becker, J. Tian, and K. Rothermel, "An Epidemic Model for Information Diffusion in MANETs," *5th ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems*, Atlanta, Georgia, USA, 2002.

[11] O. Sekkas et al, "Probabilistic Information Dissemination for MANETs: the IPAC Approach," *20th Tyrrhenian Workshop on Digital Communications*, Pula, Italy, 2009.

[12] V. Karyotis, A. Kakalis, and S. Papavassiliou, "Malware-Propagative Mobile Ad Hoc Networks: Asymptotic Behavior Analysis," *Journal of Computer Science and Technology*, 23(3) pp. 389-399, 2008.

[13] E. Huang, W. Hu, J. Crowcroft, and I. Wassel, "Towards Commercial Mobile Ad Hoc Network Application: A Radio Dispatch System," *9th Annual International Conference on Mobile Computing and Networking*, San Diego, California, USA, 2003.

[14] F. Bai, and A. Helmy, "A Survey of Mobility Modeling and Analysis in Wireless Adhoc Networks," *Book Chapter in the book "Wireless Ad Hoc and Sensor Networks"*, Springer, October 2006, ISBN: 978-0-387-25483-8.

[15] S. Gowrishankar, T. G. Basavaraju, and S. K. Sarkar, "Effect of random mobility models pattern in mobile ad hoc networks," *International Journal of Computer Science and Network Security*, vol. 7, no. 6, pp. 160-164, 2007.

[16] M. K. J. Kumar, and R. S. Rajesh, "Performance Analysis of MANET Routing Protocols in different Mobility Models," *IJCSNS International Journal of Computer Science and Network Security*, p22, VOL. 9 No. 2, February 2009.

[17] B. Cohen, "Incentives Build Robustness in BitTorrent," *1st Workshop on Economics of Peer-to-Peer Systems*, UC Berkeley, California, USA, 2003.

[18] C. Bettstetter, "Mobility Modeling in Wireless Networks: Categorization, Smooth Movement, and Border Effects," *ACM Mobile Computing and Communications*, vol. 5, no. 3, pp. 55-67, July 2001.

[19] V. A. Karyotis, M. Grammatikou, and S. Papavassiliou, "A Closed Queueing Network Model for Malware Spreading over Non-Propagative Ad Hoc Networks," *In Proceedings of the 3rd ACM workshop on QoS and security for wireless and mobile networks (Q2SWinet '07)*, New York, USA, October 2007.

[20] M. Reiser, and S. S. Lavenberg, "Mean-Value Analysis of Closed Multichain Queuing Networks," *Journal of the ACM*, 27.2: 313-322, 1980.

[21] F. Bause, and P. S. Kritzinger, "Stochastic Petri Nets – An Introduction to the Theory," *Vieweg*, 2nd edition, 2002.

[22] G. G. Yin, and Q. Zhang, "Continuous-Time Markov Chains and Applications – A Singular Perturbation Approach," *New York: Springer-Verlag*, 1991.

[23] W. J. Stewart, "Introduction to the Numerical Solution of Markov Chains," *Princeton University Press*, 1995.

[24] A. Zimmermann, and M. Knol, "TimeNET 4.0: A Software Tool for the Performability Evaluation with Stochastic and Coloured Petri Nets: User Manual," *Technical Report, Technical University Berlin*, Real-Time Systems and Robotics Group, TR 2007-13, 2007.

[25] S. Baarir et al. "The GreatSPN tool: Recent Enhancements," *SIGMETRICS Perform. Eval. Rev.*, vol. 36, no. 4, pp. 4–9, 2009.

[26] P. Bonet, C. M. Llado, R. Puijaner, and W. J. Knottenbelt, "PIPE v2.5: A Petri Net Tool for Performance Modelling," *In Proceedings of the 23rd Latin American Conf. on Informatics (CLEI 2007)*, San Jose, Costa Rica, 2007.

Ádám Horváth received the M.Sc. degree in Computer Science from the Budapest University of Technology and Economics (BME) in 2007. After his M.Sc. he joined the Institute of Informatics and Economics at the University of West Hungary as a Ph.D. student under the supervision of Prof. Károly Farkas. His research interests cover security of wireless networks and mathematical modeling, mainly focused on Markovian processes.

Károly Farkas received his Ph.D. degree in Computer Science in 2007 from ETH Zurich, Switzerland, and his M.Sc. degree in Computer Science in 1998 from the Budapest University of Technology and Economics, Hungary. Currently he is working as an associate professor at Budapest University of Technology and Economics and at University of West Hungary, Sopron, Hungary. His research interests cover the field of communication networks, especially autonomic, self-organized, wireless and mobile ad hoc networks. He has published more than 50 scientific papers in different journals, conferences and workshops and he has given a plenty of regular and invited talks. In the years past, he supervised a number of student theses, participated in several research projects, coordinated the preparation of an EU IST research project proposal and acted as program committee member, reviewer and organizer of numerous scientific conferences. Dr. Farkas is a member of IEEE and fellow of the European Digital Media Academy (EADiM).

Autonomous Online Evolution of Communication Protocols

Endre S. Varga, Bernat Wiandt, Borbála K. Benkő, Vilmos Simon

Abstract—**In this paper we describe an approach for optimizing multi-hop broadcast protocols in ad-hoc mobile networks with an online, distributed machine intelligence solution. In our proposed framework not only runtime parameters of a predefined protocol are optimized, but the protocol logic itself also emerges dynamically. The model is based on genetic programming and natural selection: protocol candidates compete for being picked (natural selection), then survivors get combined with each other and/or mutated (genetic operators), forming the next generation of protocol instances. To achieve this we created (i) a genetic programming language to describe protocols, and (ii) defined a distributed, communication-wise non-intensive, stigmergic feed-forward evaluation and selection mechanism over protocol instances, and (iii) a budget based fair execution model for competing protocols. We show that the result of the online, autonomous protocol evolution outperforms traditional approaches, by adapting to the local situation, when used for multi-hop broadcast problem in ad-hoc mobile networks. Experiments confirmed 50% improvement with a random movement mobility pattern, and 66% improvement with a group based mobility pattern. The evolution also protected the system from the negative effects of initially present harmful protocols.**

I. INTRODUCTION

THE choice of communication protocol is always of high importance in telecommunication networks, typically a tradeoff between the messaging overhead and the transmitted information content needs to be found. While too chatty protocols waste resources such as bandwidth and processing power, unnecessarily tight-lipped communication strategies hinder the flow of information, and as a consequence, impede the effective operation of the system. The protocol selection problem becomes especially challenging in networks with highly dynamic structure and highly dynamic load characteristics, for example in opportunistic ad-hoc networks where mobile nodes are present. Recent studies indicate that while there is no clear answer for the protocol selection riddle in general, it makes sense to evaluate the goodness of communication protocols for a certain problem case [18], [7], [4], [1]. In this paper we focus our research on a specific subset of protocols, namely the multi-hop broadcast protocols where the aim is to efficiently distribute a set of messages to all of the nodes in the system.

Manuscript received 29 February, revised 10 April, accepted for publication 15 April, 2012.

E. S. Varga is a PhD student at the Budapest University of Technology, Department of Telecommunications 1117 Budapest (email: vendre@hit.bme.hu).

B. Wiandt is a PhD student at the Budapest University of Technology, Department of Telecommunications 1117 Budapest (email: bwiandt@hit.bme.hu).

B. K. Benkő was with the Budapest University of Technology, Department of Telecommunications 1117 Budapest (email: bbenko@hit.bme.hu).

V. Simon is an assistant professor at the Budapest University of Technology, Department of Telecommunications 1117 Budapest (email: vendre@hit.bme.hu).

The idea behind this article is to abandon the classic approach, where the communication protocol is a static building block of the system, and, instead, introduce the principle of autonomous protocol selection and protocol evolution:

• Protocol selection means that existing protocol instances compete for being selected for use.

• Protocol evolution means that new protocol instances get created systematically, by combining and modifying existing protocols, in order to enable the emergence of even more successful instances.

Our vision is a system with a high degree of distributedness and autonomy. Nodes of the system pick their protocols autonomously, based only on locally available information. Moreover, we introduce the idea of inverted decision, that enables decision making without straining the network with explicit information collection messages.

This is a drastic departure from traditional protocol engineering: clearly, the vision of communication protocols changing and shaping in an online manner during the system's normal operation is a drastic image, and rather different form the mainstream of the state of the art. However, we believe that the communication, as being a heart of the problem in many cases, is also a place for being autonomous. One advantage is that the evolution mechanism removes the burden of designing communication protocols manually. The use of machine intelligence not only reduces costs but with a suitable evolution and selection model also guarantees the emergence of successful protocols in the end. Another advantage is that the distributed, purely local selection mechanism adds flexibility and fault tolerance: enables nodes to adapt to the very local challenges, and the presence of multitudes of protocols in the system at each moment guarantee that there will always be (or emerge) suitable protocols for unseen situations.

In [14], as a precursor to the current work, we used natural selection to achieve self-adaptation of multi-hop broadcast protocols in ad hoc networks, through automatically selecting the optimal one from a predefined set of protocols, however, no protocol evolution was present in that work. In [17] we published an early version of the on-line genetic programming framework, and demonstrated that the global goodness of a protocol can be reliably approximated with pure local measurements under certain circumstances. In this paper we introduce our matured model and demonstrate its mechanisms by showing the situation dependency and dynamics of the evolution through excessive simulation.

The structure of the paper is as follows. In section II we discuss the background of the problem as well as related work. Section III describes the vision of autonomous protocol selection and its main ideas. In section IV we focus on the genetic programming framework created for the multi-hop broadcast problem. Section V presents the results of the experimental evaluation where we analyzed the course of evolution under different environmental conditions. In section VI we give our overall conclusions.

II. Background and Related Work

This section discusses the background and related work in connection to our problem. We define multi-hop broadcast and present the difficulties implementing such protocols, then we investigate the known directions for optimization.

A. Multi-Hop Broadcast

It is a common task in ad hoc networks to distribute messages globally to all, or almost all, participants. This is basically an extension of local broadcast, usually referred to as multi-hop broadcast. By nature, this kind of service consumes a significant amount of resources (channel usage, collisions), therefore optimization is of high importance.

Channel usage is just one of the difficulties that present themselves when one implements global scale broadcast protocols. One dangerous phenomenon is the so called Broadcast Storm [13] that happens when multiple nodes start forwarding a message simultaneously after receiving it from a common source node, leading to excessive collisions. The common presumption used in protocol design, that traffic patterns of neighboring nodes are uncorrelated with each other, is not valid for this case. In the case of multi-hop broadcast, several nodes may decide to transmit at the same time, and collide even after several backoff events. To avoid this outcome, protocols have means to de-correlate the traffic of neighbors, for example by waiting for a random time before forwarding the message.

Multi-hop broadcast algorithms typically exploit the local broadcast channel to reduce channel usage and the number of collisions in the system. This way, as one transmission may be overheard by multiple devices, it is possible to drastically reduce the amount of transmissions. The essence of this optimization is to identify or approximate a Minimal Connected Dominating Set (MCDS) [10], [7] in the network, and broadcast the message once per set.

A set for a graph $G(V, E)$ is a Connected Dominating Set if M is a connected subgraph of $G(V, E)$ and for each vertex either or there exists an edge so that . A Connected Dominating Set M is a minimal CDS (MDCS) if $|M|$ is minimal. An additional constraint in multi-hop broadcast is that if B is the set of vertices containing the nodes that initially possess the payload to be broadcasted, then must hold. If the vertices V of the graph $G(V, E)$ stand for nodes in the network and an edge $e = v, w$ represents that v and w are in radio range, then an MCDS gives the smallest set of nodes needed to accomplish a successful global broadcast.

The identification of an MDCS raises several questions. First of all, it is an NP-complete problem. Another problem is that the MCDS model describes a static scenario, where the connection between nodes do not change. Clearly, if the network is distributed and dynamic (nodes move, disappear or new nodes appear), and changes may occur much faster than they can be discovered, then a centralized model, such as a centralized MCDS solver is not practically possible. Instead of tackling with real MCDSs, broadcast protocols typically use some kind of approximation based on simple heuristics and local knowledge. These heuristics range in sophistication from simple counter based solutions to probabilistic methods and complex graph theoretic approximations [19], [4]. A common point in these approximations is that they concentrate on the closest part of the network rather than dealing with the whole topology. We also followed this approach in our system.

B. Protocol Optimization Directions

Various literature sources investigate possible protocols for multi-hop-broadcast and their performance characteristics, a few examples are [18], [7], [4], [1]. Authors in [9] also give a theoretical upper bound for the worst-case performance of their algorithm.

Results suggest that there is no general winner; instead, the performance of a protocol heavily depends on volatile attributes of the environment. These attributes include mobility patterns, node speed, node density, transmission technology, and traffic models. Selecting the suitable protocol, therefore, requires deep and exact knowledge about the actual environment. However, that is generally hard to acquire, given the complex factors involved, such as human behavior influencing the mobility pattern and the load characteristics. Worse, the environment will change over time, through appearance and disappearance of nodes, technology turnovers, or changes in the usage practice, i.e. human habits; therefore any static off-line design is just a compromise.

The issues above raise the question whether an automated, online, adaptive approach could solve the matter of obtaining the best protocols for a given situation. The use of online, adaptive techniques for protocol optimization (i.e. fine tuning of operational parameters on-the-fly) is a known, but not widely used practice. For protocols, even if machine learning is applied, this step typically happens during the manual design phase, and not as part of the operation of the actual system. An exception is [5], where authors used online machine learning to approximate the behavior of sophisticated broadcast algorithms and found that simple heuristics were able to reproduce the sophisticated decision with 87% accuracy. This result indicates that in practice small but powerful heuristics could provide good approximations instead of sophisticated calculations. Note that Colagrosso's work uses predefined (fixed) protocol bodies, and aims to optimize the runtime parameters of these protocols. Our approach goes one step further: in our work the protocol body itself is also an emergent, ever-changing element.

The idea of on-the-fly protocol selection or protocol switching has been present for many years in other areas, such as in cryptography. In the field of telecommunication protocols, [5] proposed the idea of using machine learning to switch between a small set of predefined protocols in order to accommodate to the recent topology changes. In [14] we proposed stigmergic communication and natural selection for online, automatic protocol replacement. Natural selection as a tool is not unheard of in this area, in [2] authors applied a form of natural selection for parameter optimization of ad-hoc network protocols, using explicit feedback from neighboring nodes. Our selection approach is different; our approach does not require explicit feedback, resulting in a significantly smaller communication overload, furthermore, the mechanism works with arbitrary broadcast protocol without modification.

C. Genetic Programming

The basis of our work is genetic programming. Genetic algorithms are biologically inspired random search algorithms directed towards a global optimum, based on generations of solution candidates (individuals). In each round, the algorithm evaluates individuals with a fitness function; then, the next generation gets produced by applying genetic operators (crossover and mutation) on the selected individuals of the current generation. Genetic programming (GP) is a form of genetic algorithm, where individuals are programs composed of instructions in a particular programming language. When using GP we generally distinguish on-line and off-line approaches. 'Off-line' means that solutions are generated during a design phase, and the result is then used unmodified in the operational phase of the system; while in the 'on-line' case the evolution itself is part of the system and new solution instances are generated continuously, during the system's normal operation. According to our knowledge, on-line genetic programming has not been applied in the area of broadcast communication protocols before.

The variety of challenges present in multi-hop broadcast protocols for mobile ad hoc networks, such as the unpredictability of the position or speed of the mobile node, changing topology and the diversity of devices, makes it an ideal target for genetic programming. Protocols need to cope with numerous and more or less distinct aspects of the problem; hence, these aspects can be freely blended by the genetic algorithm so that the blend is still likely to make sense. Human-designed protocols typically have distinct "sweet spots": working best under different conditions. By increasing the diversity of the protocols and enabling them to adapt freely to the current environmental conditions through evolution, it is possible to always maintain a protocol or a family of protocols that works well in the current environment. This approach can also be considered as an automated protocol design tool.

While it provides many benefits, the use of GP also has certain design consequences. Fine-tuning the system becomes problematic, as evolution tends to produce individuals that circumvent human imposed rules and design patterns. GP also generates a large amount of individuals, meaning that obtaining insight becomes harder (it is impossible to inspect all the generated individuals manually). However by using data mining techniques, getting an insight into the mapping relations becomes easier, as demonstrated later in this chapter.

The success of any genetic algorithm relies on a carefully designed fitness function. In ad hoc systems the unavailability of global, system-wide data demands a careful approach to fitness calculation, collecting any kind of performance metric may easily result in a prohibitive amount of channel usage.

III. THE MODEL OF AUTONOMOUS ONLINE PROTOCOL EVOLUTION

This section describes the model of online, automated protocol evolution, comprising of four main building blocks:

1) natural selection with decision inversion to select which protocols survive for the next round
2) a budget based execution model for the current protocol instances
3) a genetic programming language to a describe protocol instance
4) genetic operators in order to combine and mutate protocols.

A. The overall picture

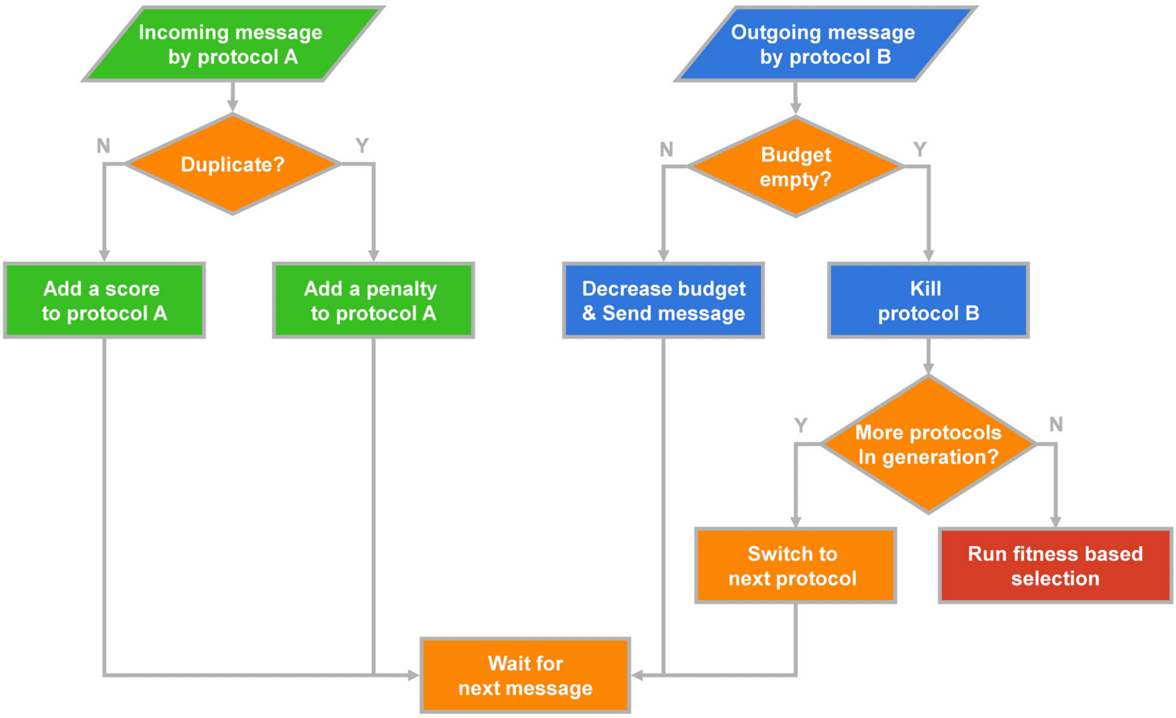
The protocol evolution is a fully distributed, asynchronous mechanism; each node selects and generates protocols on its own agenda. The core process is a loop, as visualized in figure 1b.

The evaluated protocols undergo a selection step, deciding which protocols survive and which end their life time. Once the surviving generation is selected, genetic operators, i.e. crossover and mutation, are used in order to introduce further new protocols by combining and/or modifying survivor instances. The program code of the new protocols gets compiled, resulting in a new generation of executable protocol instances. Then each of the protocols is executed, sequencially, using a budget-based execution scheme, giving equal opportunity to each individual of the generation to live. Finally, the loop starts over.

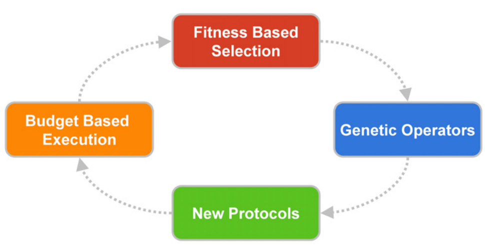
Protocol evolution runs on each node of the network, in parallel, without any explicit synchronization with other nodes. No global clock is assumed. On the other hand, an implicit synchronization is indeed present in the system. Neighbors, as part of the inverted decision making mechanism, discover each other's protocols; thus, a successful protocol instance may spread over the network from hub to hub. This fully distributed scheme also brings fault tolerance to the system.

A distributed protocol evolution model needs to answer the following questions.

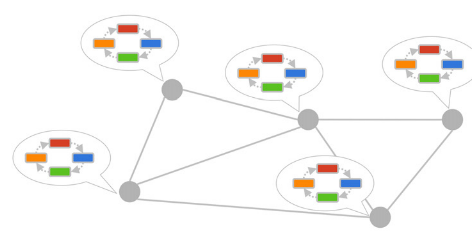
- How to measure the goodness of a protocol. The decision must be made locally, without the aid of global or wider area help, so only local or nearly-local metrics are acceptable. We defined a set of locally available evaluation factors to estimate the overall goodness of a protocol. In [17] we showed that these local metrics are linearly dependent, strongly associated with the global metrics, and are very good approximators for it.
- How to avoid the communication overhead of the measurement. We propose an inverted decision making model



(a) Inverted selection mechanism executed at the receiver node, assuming that the sender at the moment executes protocol A, and the receiver executes protocol B.



(b) Flow of protocol evolution at a node.



(c) Network-wide view of protocol evolution.

Figure 1: The Model of Autonomous Online Protocol Evolution

which avoids the sending of feedback messages, saving bandwidth and also helping the system stay asynchronous.

- How to describe protocols in a way that is robust enough even for crossover and mutation operators. We do not only need to avoid syntactic errors, but also need to facilitate the emergence of presumably sensible new protocols (semantic correctness). We propose a genetic programming language.
- How to evaluate a set of protocols at runtime, during the system's normal operation. Clearly, the evaluation should happen in a fully online manner, i.e. by sending real messages rather then through some kind of internal simulation. We propose a budget based daisy chain model to give equal chance to each protocol to show its strength.

B. Natural Selection with Decision Inversion

Natural selection is a fundamental principle in evolutionary systems. In our case, when the system is equipped with several protocol candidates, in order to select the most suitable ones, we need to measure the performance of each candidate and push the system towards the use of them.

To realize a lightweight but efficient natural selection mechanism is highly non-trivial in our case. Protocols, in order to be successful, need to address conflicting requirements, i.e. maximal coverage versus minimal duplication count. These two requirements not only make the choice of evaluation metrics hard, but also pose measurability problems, such as: (1) Only the sender node is able to reliably measure the real cost of a successful message transmission. Lost messages, by definition, could not be seen by other nodes, so the amount of work (total sent messages) is only known by the sender (2) Only the receiver node is able to reliably measure the number of redundantly received messages (we call these *duplicates*). (3) Each receiver node can measure the number of local duplications (i.e. the duplications they personally received), but they can not measure the number of total duplications in the system. (4) In order to collect the measurement results at a designated place, message passing is needed over the same channel as normal (payload) messages use. Measurement messages, just like payload messages, may get lost.

To get a clearer picture of the measurement problem, consider the case when a node sends out the same message ten times. Although naively we could consider this as ten duplicates, in reality this is not necessarily the case. If during the timeframe of the broadcasting of the ten messages ten nodes pass by, each hearing the message only once, then there are no duplicates at the receivers. Now consider the case, when the sender sends out exactly one message, with ten other nodes in transmission range. If all the neighbor nodes already received this message in the past then this transmission alone generates ten duplicates at the receiver nodes. From these examples it should be clear, that the amount of copies sent and the amount of duplicates are not the same, and the former is known by the sender, and the latter is known collectively by the neighbor nodes. To centralize all this data at a designated place, further communication is needed. Unfortunately this generates a large overhead, especially if the reliability of this procedure is considered.

The above factors imply [14] that implementing a centralized (even locally centralized) protocol selector criterion is impractical, because the reliable collection of performance data is both technically challenging and wasteful in terms of channel usage. Instead, we propose a feed-forward selection method using stigmergy and natural selection.

The solution is based upon the idea of decision inversion. The naive implementation of natural selection would be that every sender collects its performance metrics from the surrounding receivers and creates the next protocol generation according to this information. However, as explained above, this can not be efficiently implemented in general. Instead of trying to select locally at the sender, we delegated the task of decision making to the receivers, because they are in the optimal position to observe the performance of a protocol. Nodes score and evaluate all protocols that successfully sent messages to them. Then, the next generation of the node's protocols will be comprised of the highest scoring sender protocols. This essentially means, that no performance metrics are disseminated back to the sender, instead, the evaluation is delegated to the set of receivers, collectively.

To make this possible, receivers must know the protocol that generated the given payload message. This is achieved by senders attaching a representation of the protocol itself to every payload. Such compound packets act as a virtual seeds where the "nutritional" part of the seed is the payload and the genetic material is the code of the sender protocol. Nodes (as receivers) collect seeds from surrounding nodes and assign scores to the protocol instances they carry. Every payload that is useful to the receiver node means a score for the sender protocol. Every unnecessary message (duplicate) means a penalty to the sender protocol. Seeds with useful nutritional parts (high scores) will survive on the receiver node. The main advantage of the inverted selection is that performance results do not need to travel back over the network to the sender: instead, the receiver will utilizes them during the creation of its own next protocol generation. Of course in the next round, the sender may meet the offsprings of its own good protocol. With this approach, both the measurement overhead and the need for synchronization is minimal.

It is important to see, that this procedure optimizes protocol instances locally, using local information. This is similar to the way how static, human-made protocols work, but instead of only optimizing protocol parameters, our framework is capable to "invent" new heuristics as well.

C. Budget Based Protocol Execution Model

As the selection of new algorithms happens at the receiver it is impossible to implement explicit cost calculation without expensive control overhead. To overcome this, we adopted a stigmergic solution: by assigning a limited transmission budget (called *quota*) to each protocol instance at the sender, protocols are forced to make good use of the channel resources. Any lost or duplicate message is a lost opportunity for reproduction; therefore transmission has an implicit cost function, even if it is not expressed directly. Similarly, we added a timer that upon firing, removes the current protocol, and replaces it with the next instance, forcing protocols to use the available time efficiently, we call this the *time limit* of a protocol

D. A Genetic Programming Language for Protocols

Natural selection implemented by decision inversion, along with the budget based execution mode, answers the question how protocols should be executed, evaluated and selected for survival in a distributed fashion. In this section we introduce our choice protocol representation and the tools used in protocol composition.

In order to make Genetic Programming and evolution possible, the representation of protocols needs careful consideration. As the protocols in the system are no longer engineered by humans, a lightweight, flexible and robust formal description is needed which suits genetic operators.

In a GP environment protocols must be represented by their program code. However, to use a general purpose programming language as the protocol representation would be problematic because the application of genetic operators could result in frequent syntactic errors and uninterpretable code. To avoid these issues several GP specific languages were designed. We selected the Push language [15] as a starting point for the design of our own language, called GPDISS.

Push is a natural choice, as it is a widely known and used language in GP related research, and Push code is relatively easy to interpret by humans. Artificial chemistries were also considered [8], particularly the Fraglets language by [16], which was used to conduct experiments in protocol evolution in [11]. However, artificial chemistry based languages are notoriously hard to analyze manually, therefore we decided to exclude these languages from our experiments.

Push is a stack based language, meaning, that its instructions do not have explicit arguments, instead, they are taken from the corresponding stack. For example, an 'if' instruction takes its argument from the top of the Boolean stack. This enables a mutation and crossover friendly, flexible syntax.

When defining our custom programming language, GPDISS, we borrowed ideas from Push, but augmented it with new instructions and introduced new concepts in order to provide a better fit for our particular problem. Our modifications include:

- Extended instruction set in order to match the multi-hop broadcast problem.
- A new data type (relation type) to describe complex data relationships. This is particularly useful for modeling graph-like structures
- The concept of typed event handlers that simplify the crossover of message handling routines

1) Extended instruction set: The most important instructions of GPDISS are shown in table I. The set of operations include control instructions, basic arithmetic and logic operators, control flow, messaging, timers, and the handling of complex data using relations. Note that this particular instruction set was designed for the multi-hop broadcast problem. For a different problem a slightly different instruction set may be required but without affecting the basic principles.

Table I: Most important instructions in GPDISS.

Stack	Instruction	Description
*	dup, drop, swap,rotate, hold, release	Common instructions available on all stacks implementingcommon stack manipulation operators.
number	add, div, mult, random	Simple floating point arithmetic and random number generation.
bool	and, or, not, if, while	Boolean logic (usually for control flow).
list	additem, nth, remove_first, delete_duplicates	Typed list handling. Common operations are available.
relation	addpair, union, join, remove_first, invert, intersect	Typed relations are like two column tables. They can be filtered, joined, intersected, etc.
messages	send, sender	Common instructions for handling all types of messages.
timers	id, start_timer	Timers can be used to schedule different tasks at different points in time.
-	else, endif, do, return	Control flow constructs.

2) Relation type: A new data type, called relation type, and a set of accompanying instructions were devised in order to enable efficient calculations on graph structures, tabular data, or trees, as these are common in broadcast protocols. On the relation stack ordered, typed pairs of objects can be stored. Relations can be imagined as two column tables, the first column being the key, and the second column being the value. The key and the value columns are typed, which means they receive their content from the appropriate typed stack. Relations give a new dimension to stored data by describing relationships between the objects on the stacks. Relations are immutable: every operation creates a new instance on the relation stack. With the help of the operations defined on relations sophisticated data manipulations are possible, such as filtering by key or value, intersecting, joining, subtracting or creating the union of two relations. The code snippet in Figure 4 creates a new relation containing our direct neighbors, supposing that we already have a neighbor map (node-node pairs) of the network in our vicinity on top of our relation stack.

Table II: Event handlers in GPDISS

Message type	Event handler
data	The common wrapper for all data messages in the system. Data messages contain payload originating from a node with the goal of reaching all nodes in the system.
neighbor	Neighbor messages contain neighbor information of the sender nodes. This can be used to locally map the connectivity graph of the network
timer	Internal event handler. A protocol receives a timer event when one of its timers fire. One can use it to implement features such as Random Assessment Delay (a transmission delay used in broadcast protocols)
init	Internal event handler. Protocols receive this event once upon initialization, before the first message arrives to them.

3) Event handlers: Instead of defining programs monolithically, we defined multiple hooks, called event handlers, that protocols may use to implement their logic. Therefore, the code of a protocol is a list of assembly-like instructions grouped into event handlers.

Each message type has its own event handler (illustrated in table II), which gets activated when a message of that type arrives. The activation is controlled by an underlying meta-protocol, shared by all nodes. The purpose of this meta-protocol is to ensure that the result of the evolution, i.e. the messages, remain interpretable for all possible receiver nodes. The meta-protocol defines how to decode and interpret the messages sent by other nodes. The possible most basic meta-protocol simply defines the format of messages, and ensures that the incoming message is forwarded to the corresponding event handler. In GPDISS, the meta-protocol defines the format of payload messages, and a few control messages, although it does not define the ordering, timing of them. This choice reflects our conservative approach to GP, restricting the protocols to use control primitives we already know and consider to be useful. It is currently impossible for protocols to "grow" their own custom messages. It is important to note, however, that the meta-protocol does not restrict the order of messages, nor does it enforce messages to get processed (event handlers may skip any message); therefore there is still a large degree of freedom for protocols to explore. Furthermore, with a different meta-protocol, we could easily enable the emergence of new message types, as well.

E. Genetic Operators

Event handlers are the basic units used by genetic operators. The crossover operator mixes the bodies of two event handlers; while the mutation operator changes the instruction sequence of a single handler. Implementing a protocol via event handlers is a natural and practical choice because it enables the genetic operators to modify corresponding parts of a protocol by mixing code snippets of similar task or extending protocols by adding new event handlers (or removing ones).

Although syntax is maintained, the semantic correctness of a program can not be guaranteed after the application of genetic operators. If an instruction is impossible to execute (such as pop on an empty stack), it defaults to a no-op instruction, which does nothing, and the execution of the code continues undisturbed. This results in a quasi-linear [3] genetic programming language.

IV. Genetic Programming Framework for Multi-Hop Broadcast

This section demonstrates how the introduced described protocol evolution model can be used for implementing multi-hop broadcast. Note that multi-hop broadcast is especially suitable for genetic programming as it has several, more or less independent aspects, and several good and combinable solution strategies.

First we describe the initial population of protocols used as the starting point of the evolution; then, we elaborate on the details of the execution and selection mechanism and the used genetic operators. Finally, we discuss the metrics we used to investigate the course of evolution.

A. Initial Protocol Population

The initial population was selected from a small set of well-known protocols that are simple enough to be the starting point of evolution.

- Adaptive Periodic Flood (APF) is an optimization of blind flood. An APF node periodically transmits all the messages it possesses to all neighbors it encounters, after a random waiting period. However, when it detects that there is another node sending the same message, it increases the period of broadcasting to reduce the total channel usage.
- Gossiping (Gos). A gossiping node forwards the received message to its neighbors with a given probability. Gossiping is easy to analyze mathematically, as neighboring nodes have minimal effect each other's operation.
- Density sensitive adaptive gossiping (AGos). In adaptive gossiping the probability of propagating the message depends on some condition. We used a density sensitive model, where the probability decreases as the number of neighbors gets higher.
- Aggressive flood (AgrF). In case of aggressive blind flood the node propagates each received message to its neighbors n times, with a certain waiting time between the repetitions. We used two different repetition amounts to analyze how the evolution can protect the system from aggressive attackers, trying to propagate their codes by sending messages in an aggressive manner.

The criteria for the choice of protocols in the initial generation were to include simple but versatile algorithms to see if evolution is able to improve them and to include a proven 'dangerous' algorithm to see if the evolution process can eliminate it by the use of strictly local metrics. Highly effective, but overly complex protocols were excluded from our experiments, as they are usually not good candidates for Genetic Programming.

Note that AgrF is a particularly dangerous protocol from the viewpoint of this stigmergic feed-forward evolution model, as it attempts to spread its seeds at the highest possible rate.

B. Selection

Every protocol generation is created from the previous locally available protocol generation and those non-local protocols that were discovered in the previous round. We used SUS (Stochastic Universal Sampling) with a score function that gives priority to better performing individuals [12]. SUS provides zero bias and minimum spread, meaning that the actual and expected probabilities of selecting an individual are equal and the range in the possible number of trials that an individual can achieve is minimal. SUS is a variant of the roulette wheel selection. The steps are as follows. (1) Order the individuals by their fitness score in non-increasing order. (2) Allocate slices on the wheel proportional to the fitness of the individuals. (3) Calculate the step-width. For example if we want to select n elements the step-with is (sum of fitness values)/n. (4) Choose a starting point on the wheel between 0 and step-width. The corresponding protocol is selected for the next generation. (5) Make n-1 step-width wide steps, and always select the protocol assigned to the given slice. Selecting an individual means making a copy of it and adding that copy to the new generation.

The new generation is then shuffled, individuals are coupled into pairs, and with a certain probability the crossover and mutation operators are applied. Crossover is applied on the pairs, and mutation on the instances.

C. Crossover and Mutation

A modified one-point crossover is used for combining two event handlers. Given that the protocol pair (A, B) is selected for crossover, the algorithm is the following:

- Choose an event handler from A randomly. If B has no such event handler, then return.
- Select a cutting point randomly in A's handler, and another point in B's handler. Cut the handlers along the cutting points, resulting in four fragments: A-head, A-tail, B-head, and B-tail.
- With 0.5 probability exchange the head and the tail fragment of the original handlers.
- Glue fragments together forming two new handlers, an (A-head, B-tail) and a (B-head, A-tail). To protect handlers from growing indefinitely, we limited the maximal size of event handlers; bodies above the limit were chunked.

We use constant parameter mutation, meaning that instead of modifying instructions in the event handler body, the mutation affects the constants, i.e. the runtime parameters of the algorithm. For example such a runtime parameter is the message propagation probability in the Gos. When a parameter with current value x is mutated, the new value is chosen from the $(0, 2x]$ range with a Gaussian distribution, favoring fine-tuning but also allowing larger changes.

D. Execution and Parallelism

We used a simulation with completely independent node entities, each running its own Virtual Machine executing GPDISS code and implementing our meta-protocol. The clocks of the nodes were not synchronized, and the VMs had slightly differing opcode execution times.

A new payload unit was generated periodically and nodes had the task of broadcasting it over the network.

E. Measurement Metrics

During simulations we collected various metrics about the protocol instances that appeared in the system. Recording of measurements happens whenever a protocol finishes its execution, either because it depleted its messaging quota or because the time limit expired.

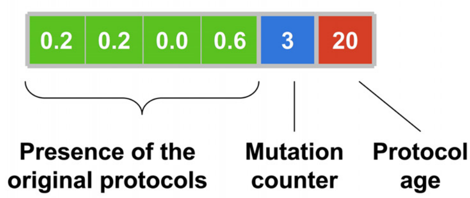


Figure 2: Protocol fingerprint for the case of 4 initial protocols. This protocol instance is the result of 20 evolution rounds, with 3 mutations. The original protocols are present in it in the amounts of 0.2, 0.2, 0, and 0.6.

The first metric, the protocol fingerprint is a vector, describing the genetic constitution, mutation count and the age of the protocol instance, as shown in figure 2. The first segment, the genetic constitution part, approximates the presence of the code parts of the initial protocols in the current instance. For example, a pure $0th$ initial protocolfor example APF, has a $[1, 0, 0, 0]$ genetic constitution part, while the mixture of the $0th$ and 1st initial protocols is represented as $[0.5, 0.5, 0, 0]$. The length of the constitution part equals to the number of initial protocols, and the sum of the constitution values is always 1. The second section of the fingerprint is the mutation counter. The third section is an age indicator, containing the local generation number at the node that created the instance.

In addition, we measured a set of local and global metrics for each protocol instance. In [17] we showed that the aggregate of certain local metrics could be very efficient approximators for the global ones, so we will not differentiate between local and global scores in the current analysis. The list of measured metrics is the following:

- GEN: Generation of a protocol, i.e. the value of the local evolution counter on the generating node. Note that as the system is asynchronous, new generations get produced by different rates at the different nodes
- SCORE: Score of the protocol instance, particularly, the number of useful messages minus the number of duplicates it generated.
- TIME_SLICE: Global time divided into epochs. The total simulation time was divided into 20 equal time segments, assigning the numbers $0..19$ to each.
- PROTOCOL_ID: Protocols were assigned identifiers in the order of their deaths. There were approximately $100,000$ observed protocols per experiment

It was a deliberate decision not to measure which protocol instances survive and which does not. This is because the actual performance of the system in the current round depends on the executed protocol instances and not on the surviving ones.

V. Experimental Evaluation

We executed several simulations to evaluate our protocol evolution model in the context of the multi-hop broadcast problem. First we describe the environment used in our experiments including the different scenarios used. Then, the results of the measurements using different mobility models are discussed. Finally, we illustrate the asynchrony of the system.

A. Simulation Environment

For the experiment we used a custom created discrete-event simulation engine, written in the Scala language [6], and our implementation of the GPDISS language and VM in Java. The compiler for the GPDISS was created with the ANTLR compiler generator.

Experiments were conducted on a laptop PC with $2x2$ GHz processor and 6 GByte memory, out of which 1 GByte was used for the simulation.

The general settings were the following:

- Mutation probability: 0.1
- Crossover probability: 0.2
- Node count: 500
- Simulation time: $7000s$
- New information message to be broadcasted over the network is generated every $20s$
- Maximum age of broadcasted payload: $20s$
- Maximum time budget for protocol instances: $7s$
- Messaging quota for a protocol: $5000byte$
- Average size of a payload message: $250byte$
- Average size of an instruction: $4byte$
- Channel bandwidth: $1Mb/s$
- Initial population: 5% AgrF, and equal proportions of the other three protocols.

B. Scenarios

The attributes common for all scenarios:

- Nodes move within a geographic area (800 x 1200 m) according to a mobility pattern. Nodes within $50m$ see each other, i.e. are neighbors. The movement of the nodes cause the connectivity graph to change over time
- The nodes of the system are independent entities, without any global knowledge or synchronization
- Initially, each node starts with a single, randomly selected protocol. This single protocol is the local initial population.

- A new payload message is injected in the system periodically. The overall goal is to broadcast the payload with the possible highest total coverage and lowest duplicate count before the payload expires.
- Nodes execute the decision inversion algorithm locally. By sending messages, protocols have the chance to spread over the network.

C. Mobility patterns

We used two highly different mobility patters for the evaluation: a simple random movement pattern (M1) and a scale-free group pattern (M2). The initial distribution of the nodes over the area was random in bothy cases.

1) M1: Random direction mobility pattern: In M1, nodes randomly choose a direction and distance and move with $1m/s$ speed until they reach the desired distance. After reaching the destination, the process starts over.

2) M2: Competing groups pattern: In M2, nodes have colors assigned. Each node tends to join nearby matching-color groups with a given probability. Groups, when a new member joins, relocate towards the new mass center. Nodes with different colors repel each other, and nodes with matching colors attract each other (until reaching a minimum distance), using a spring model. When the group becomes stable, i.e. no new member joins, the whole group tries to get nearer to a point of interest (top left corner of the area). Groups compete for this area.

We used 10 colors, $5m$ as a minimum distance between matching-color nodes, $30m$ as maximum distance from the group center, $30m$ as desired minimum distance between non-matching nodes, 0.5 probability for solitary nodes to join a nearby group, and 0.02 probability for a grouped node to leave its current group.

The dynamics of the model is shown in figures 3a and 3b, visualizing the two phases of the pattern: (i) group expansion and (ii) competition between groups. Note that even though the group expansion phase results in large movements and a high amount of topology changes, the second phase, the competition of groups for the point of interest even exaggerates that by causing massive-size local changes (e.g. the collision of two dense groups). Groups may temporarily even tear up due to a collision, controlled by to the spring model. Also note that some nodes do not join groups at all.

D. Aggressor transmission rates

We used two different AgrF repetition values in different simulation scenarios: $n = 1$ and $n = 3$. The 3-fold repetition makes the protocol more robust against transmission failures at the cost of significantly reduced performance.

E. Experiments using the M1 mobility pattern

The first set of experiments analyses the random movement based M1 mobility pattern. The goal to examine was twofold: (i) whether the evolution works, i.e. produces better scores than the situation without evolution, and (ii) to analyze how the genetic constitution of protocols changes with time. An interesting question here is whether AgrF, the aggressive protocol manages to survive or the evolution purges it. We analyzed the scores and the constitution of protocols with time.

1) Scores: Figure 4a shows the score chart of the $n = 1$, M1 mobility pattern setting without evolution. This setting serves a reference for comparison. The variance of the scores is very high, so a moving average line (light blue) was added to emphasize the trend. The average does not change significantly over time, it stays around the initial -20 value meaning that for example for 25 useful messages typically also 45 duplicates got produced.

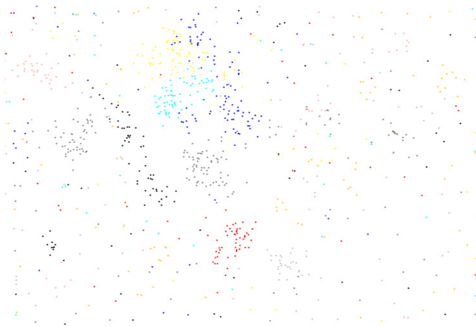
Figure 4b visualizes the same setting with evolution enabled. The variance of the scores is still high, but lower than in the no-evolution case. Also, a significant improvement in the scores can be observed; both the trend line and the minimal score rise, while the maximal achieved score stays constant. At the end of the simulation, the moving average is around -10, compared to the initial -20, which is a 50% improvement. Figure 4c displays the two previous charts in one figure. The gap between the two trends lines confirms that the evolution works.

The results produced by the more aggressive, $n = 3$ setting are summarized in figure 4d. The positive effect of the evolution is also confirmed here, with an even higher difference; because the no-evolution case produces a slightly decreasing trendline due to the more aggressive message propagation of the 5% AgrF instances. The trendline of the with-evolution case suggests that the mechanism starts producing good results after an initial struggling phase. This is because it takes time to get known with all protocols (initially, each node knows only one), and also to find a good blend. However, once a good combination is there, it is likely to produce further good offsprings.

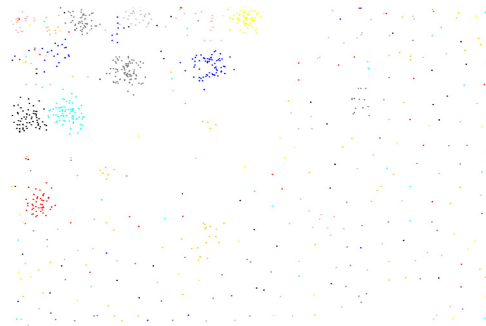
2) Genetic constitution: The next set of figures visualizes how the constitution of protocols changes with time. We quantized the constitution part of the fingerprint into 5 ranges: $0 - 20\%$, $20 - 40\%$, $40 - 60\%$, $60 - 80\%$ and $80 - 100\%$. For example, a $[0.1, 0.35, 0.55, 0]$ fingerprint constitution is quantized into a [“$0-20$”, ”$20-40$”, ”$40-60$”, ”$0-20$”] signature. Then, these signatures were summarized by algorithm type and time slice.

Figure 5a displays the participation chart of the APF protocol over time. Columns represent the summary of a time slice. For example, in the time slice 10, APF was present in $0-20\%$ amount in 53% of the protocol population, in $20-40\%$ amount in 26% of the protocols, in $40 - 60\%$ in the 15% of the population, and in $60 - 80\%$ and $80 - 100\%$ amount in $3-3\%$ of the population, respectively. The curve suggests that APF is a surviving constituent; at the end of the simulation it was present in more than half of the protocols in at least 20%. On the other hand, APF is not a dominant part in the successful offsprings, only 3% of the final population contains APF in more than 80%, and only $3 + 5 = 8\%$ contains it in more than 60%.

The same chart for Gos is shown in figure 5b. Gos is a much more dominant survivor; in the first 10 time slices it manages to remain a $80\%+$ constituent in $35 - 45\%$ of the

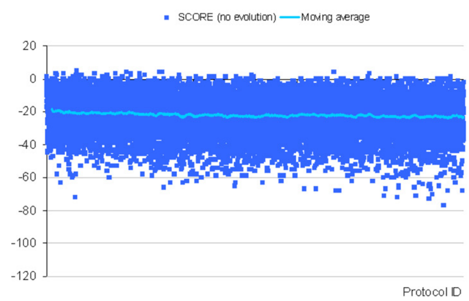


(a) M2 movement model: Group expansion phase.

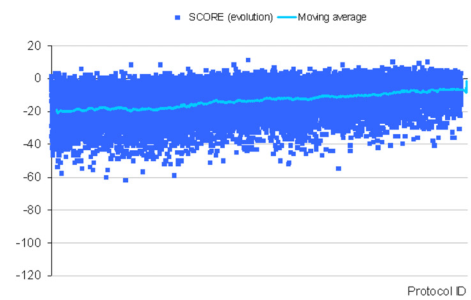


(b) M2 movement model: Groups competing to approach the point of interest in the top right corner.

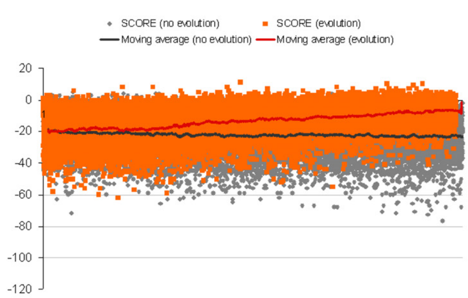
Figure 3: M2 mobility model



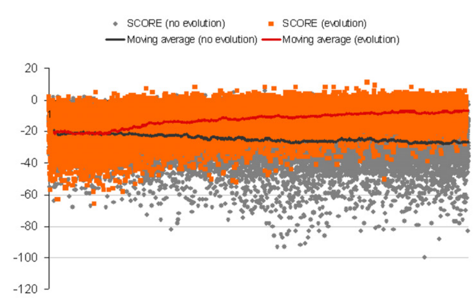
(a) Scores with M1, no evolution. 500 nodes, AgrF $n = 1$, M1



(b) Scores with M1, evolution on. 500 nodes, AgrF $n = 1$, M1



(c) Scores and moving averages with and without evolution. 500 nodes, AgrF $n = 1$, M1



(d) Scores and moving averages with and without evolution. 500 nodes, AgrF $n = 3$, M1

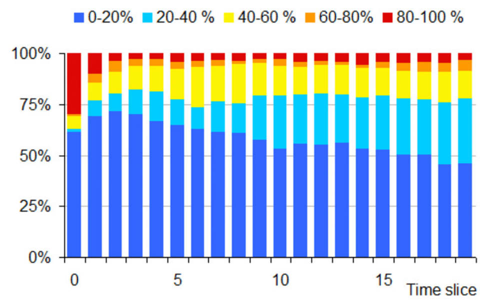
Figure 4: Scores using the M1 mobility pattern

population. At the end of the simulation, Gos still has a visible dominant presence (nearly 25% of the protocols is built from it in 80%+), but also starts blending with other protocols. Note that the presence of the protocol does not mean that it is present in an unmodified form; mutation i.e. fine tuning of constants is likely to increase the success of a protocol. This is exactly what happens to Gos, evolution optimizes the propagation probabilities to the actual neighbors.

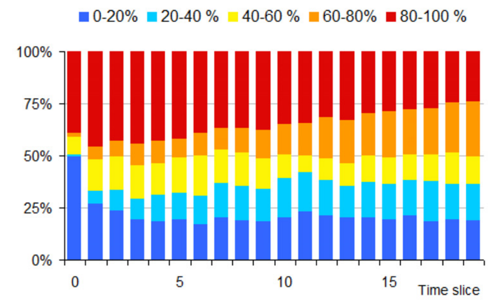
AGos offsprings, as shown in figure 5c, are less likely to be present in offsprings with time than the two previous protocols. There are a few successful crossover-generated offsprings, built partially from AGos, they are successful at certain locations but are not widely spread over the network.

The participation chart of AgrF in figure 5d, shows that in case of the M1 movement pattern this protocol is completely purged from the network in 10 time slices.

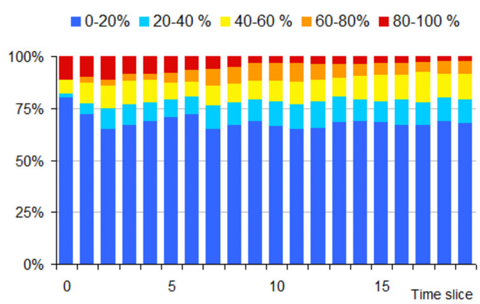
Figures 6a to 6d display the same charts for the $n = 3$ repetition case. Unexpectedly, AgrF here is not purged from the system, although suppressed quite fast. AgrF serves as a $20 - 40\%$ constituent in 3% of the protocols, constantly, in a non-increasing manner. Another difference is that the other three protocols manage to blend better with this setting, resulting in rarer 80%+ presence, and more $20 - 80\%$ presence. The higher ratio of blends means that the offsprings generated by crossover are successful.



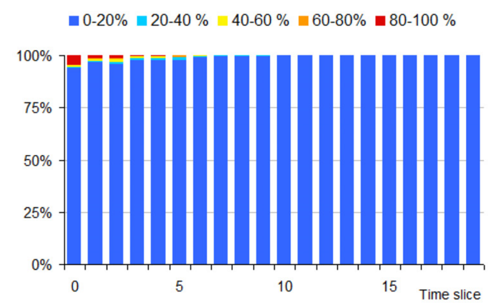
(a) Participation of APF offsprings in protocols over time. 500 nodes, AgrF $n = 1$, M1



(b) Participation of Gos offsprings in protocols over time. 500 nodes, AgrF $n = 1$, M1

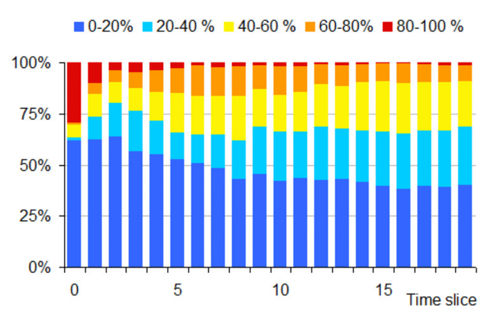


(c) Participation of AGos offsprings in protocols over time. 500 nodes, AgrF $n = 1$, M1

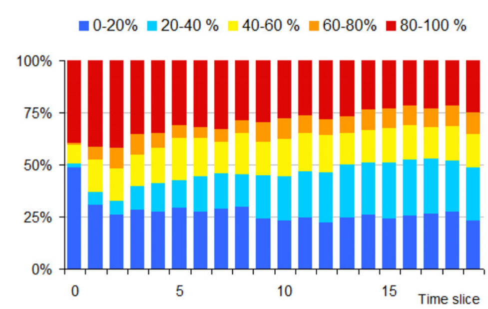


(d) Participation of AgrF offsprings in protocols over time. 500 nodes, AgrF $n = 1$, M1

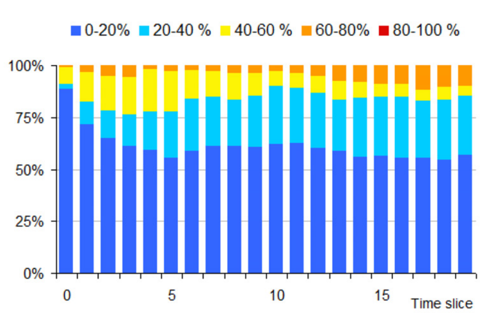
Figure 5: Genetic constitution of protocols using the M1 mobility pattern, $n = 1$



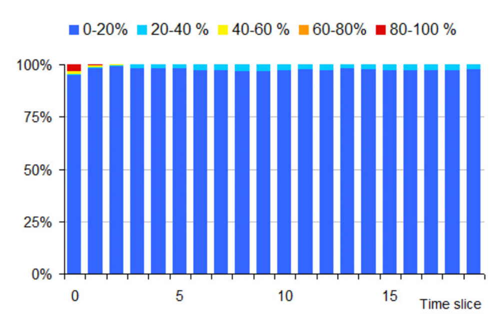
(a) Participation of APF offsprings in protocols over time. 500 nodes, AgrF $n = 3$, M1



(b) Participation of Gos offsprings in protocols over time. 500 nodes, AgrF n=3, M1



(c) Participation of AGos offsprings in protocols over time. 500 nodes, AgrF $n = 3$, M1



(d) Participation of AgrF offsprings in protocols over time. 500 nodes, AgrF $n = 3$, M1

Figure 6: Genetic constitution of protocols using the M1 mobility pattern, $n = 3$

F. Experiments using the M2 mobility pattern

The second set of experiments examines how the use of an intrinsically different mobility pattern changes the results of the evolution. Note that in this case the mobility is group based, meaning that once a node joins a group, a dense neighborhood is likely to be present for it for the remainder of the simulation. On the other hand, the competition between groups, hence collisions with other groups may highly influence the neighboring topology, also changing what kind of propagation strategies work well. Scores and the constitution charts were used for the evaluation, just like in the previous case.

1) Scores: Figure 7a shows the score curve of the $n = 1$ setting with and without evolution with the M2 mobility pattern. The main direction of the trend is the same as in the M1 case, however, the dynamics of the improvement is slightly different. The variance of the scores remains high with M2 throughout the simulation, beacsue M2 causes heavy but localized changes in the topology while groups compete for the area around the POI. The evolution is more successful with M2 than it was with M1, the moving average of the score ends at -6.5, meaning a 66% improvement compared to the non-evolution case.

The score chart of the $n = 3$, M2 case is shown in figure 7b. The improvement in means of average is also around 66% here. Again, in the non-evolving case, the aggressive flood produces a somewhat decreasing score curve and very high variance. Note that the worst score of the evolving population at the end of the simulation is basically the same as the average score of the non-evolving population, showing that the difference is really significant. Another interesting phenomenon observable in the figure is that the initial effect of the evolution is slightly negative, the score curve decreases in the beginning, before turning upwards.

Detailed analysis of the score pointed out that, as figure 7c demonstrates, the number of useful messages sent by the protocols remained the same throughout the experiment, while the number of duplicate messages declined with time. The same trend was visible in all experiments.

2) Genetic constitution: Constitution charts for the M2 case show different blending proportions than M1 did; the difference is especially articulated with the $n = 3$ setting.

With $n = 1$ (figure 8) an unexpected result is that AgrF does not get purged, instead, at the end of the simulation in 4% of the protocols it is still a 100% constituent. This is because AgrF can be a beneficial protocol in certain topologies when only a small portion of nodes use it.

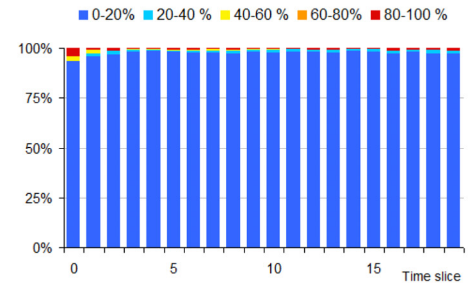


Figure 8: Participation of AgrF offsprings in protocols over time. 500 nodes, AgrF $n = 1$, M2

The charts of the $n = 3$ case are shown in figures 9a to 9d. Protocols here, with M2, blend more than they did with the M1 mobility pattern; $80\%+$ constituents are very rare, altogether 13%, at the end of the experiment, compared to the 32% of the same setting with M1.

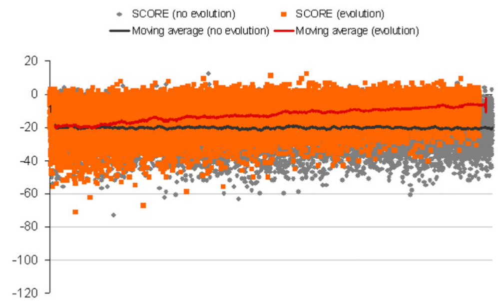
An interesting wave effect can be observed in the charts of Gos (figure 9b) and AGos (figure 9c), showing how the shift from the group expansion phase to the group competition phase influences the usefulness of these protocol as constituents. The shift occurs around slice $8 - 9$. With AgrF (figure 9d), the effect of the shift is also visible; AgrF becomes useful as a $20 - 40\%$ constituent when large and stable groups emerge (around slice 7), and manages to maintain and even improve this position until the end of the simulation. In the end, AgrF is present as $20 - 40\%$ constituent in 5% of the protocols.

G. Comparison with a homogeneous case

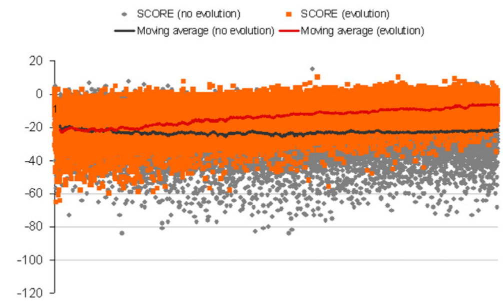
For comparison, we examined how the system (without evolution) would perform if only one non-aggressive protocol family was present, instead of the three tackled with in the experiments before. We kept the 5% AgrF presence for this setting too, as it would not make sense to compare a 100% homogeneous case with the previously discussed settings, where an aggressive, harmful protocol was present. Figure 10a visualizes the measured score values over time. The trend shows no improvement. This confirms that the comparisons used in the previous experiments, i.e. the use of a four-protocol setting, were valid. The use of several protocols did not introduce general bias or disturbance that is not present in a homogeneous case.

H. Execution metrics

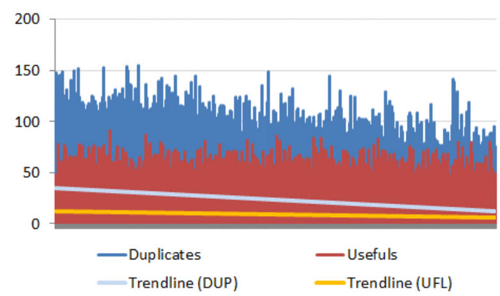
Finally, we demonstrate the asynchrony of the system. Figure 10b shows the age of each protocol (the value of the local evolution counter at the generator node) in the order of measurement, i.e. in the order of the protocols' death. The line-like shape suggests that the evolution speed, although not being synchronized explicitly, is more or less the same in most nodes. Deviations from the average line suggest a



(a) Scores and moving averages with and without evolution. 500 nodes, AgrF $n = 1$, M2

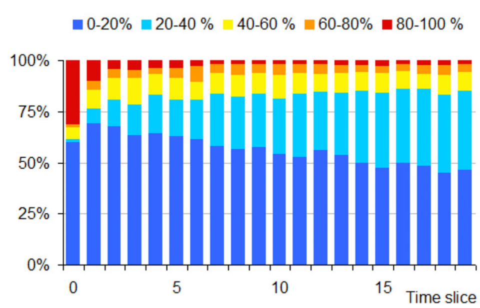


(b) Scores and moving averages with and without evolution. 500 nodes, AgrF $n = 3$, M2

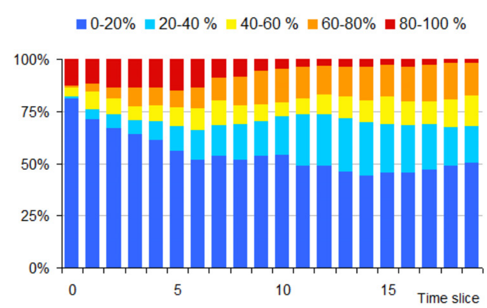


(c) Number of duplicates and useful messages over time. With evolution, 500 nodes, AgrF $n = 3$, M2

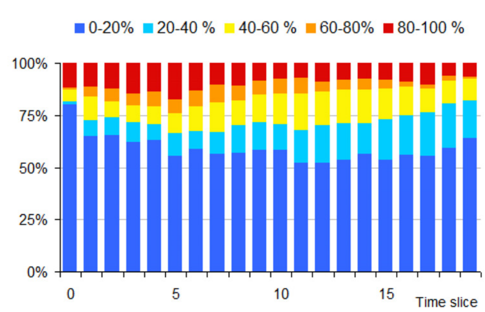
Figure 7: Scores using the M2 mobility pattern



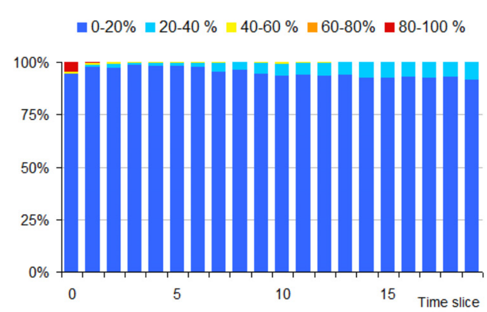
(a) Participation of APF offsprings in protocols over time. 500 nodes, AgrF $n = 3$, M2



(b) Participation of Gos offsprings in protocols over time. 500 nodes, AgrF $n = 3$, M2

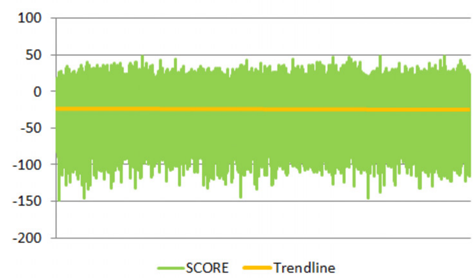


(c) Participation of AGos offsprings in protocols over time. 500 nodes, AgrF n=3, M2

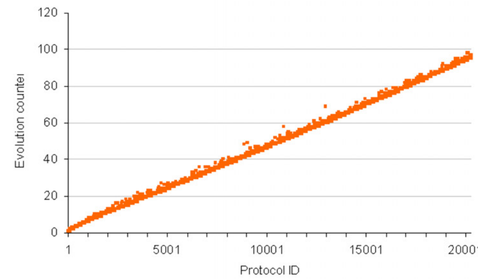


(d) Participation of AgrF offsprings in protocols over time. 500 nodes, AgrF $n = 3$, M2

Figure 9: Genetic constitution using the M2 mobility pattern, $n = 3$



(a) Score over time, homogeneous case (95% APF + 5% AgrF). 500 nodes, AgrF $n = 3$, M1



(b) Protocol age (local evolution counter) over time. AgrF $n = 3$, M2

Figure 10

local difference in the speed. Deviations typically occur in the positive direction, i.e. the rounds are executed faster at some nodes. This happens when a protocol empties its messaging budget before the time limit expires.

VI. Conclusions

Simulations confirmed that the autonomous, online protocol evolution model is a promising approach for using to optimize and self-adapt multi-hop broadcast networks. In our experiments, evolution produced $50-66\%$ gain in terms of achieved average score, and also significantly improved the worst-case score and the variance of scores.

Our model using evolution and natural selection was able to neutralize the negative effects of a malicious protocol present in the system. However, AgrF was not simply eliminated from the system, but instead, parts of its code got incorporated into good-performing offsprings in some cases. In one scenario, a small and non-increasing amount of pure AgrF instances persisted among the survivors, however they represented a small percentage.

The evolution resulted different survivor constitutions depending on the mobility pattern and on the set of initial protocols (rigorously, AgrF $n = 1$ and AgrF $n = 3$ are two different original protocols). The feed-forward selection mechanism was able to offer enough adaptivity to the changing requirements, as demonstrated with the M2 mobility pattern around the phase shift.

Our results affirm our belief, that the demands for the new forms of networking infrastructure can be effectively addressed by bio-inspired solutions. Our focus was on presenting an evolutionary framework for the family of multi-hop broadcast protocols in ad hoc networks, where it is usually impossible to find a single absolute candidate, as the optimal protocol choice always depends on the actual environment and application conditions. We introduced a novel idea in this field: instead of human engineered static protocols, autonomous evolutionary methods were applied to achieve dynamic emergence of new ones, driven by the current needs and environment of the communicating nodes. For this purpose we have introduced an evolutionary model built upon a low-overhead, feed-forward, fully distributed, stigmergy based natural selection mechanism, and a genetic programming language GPDISS incorporating some nonconventional concepts such and relation types and event handlers.

We showed that the proposed model of evolving protocols is applicable for the multi-hop broadcast problem in ad-hoc networks: with time, evolution results in better performance than that the initial, manually engineered, protocols could provide. The fitness function was defined so that it used only local and quasi-local input, resulting in a model that is applicable for fully distributed systems such as ad-hoc sensor networks. Also, the feed-forward nature of the evaluation and selection process eliminated most of the communication overhead needed for the calculation of fitness values. Additionally, the process was carried out in an online manner, that is, the evolution of protocols happened continuously during the normal operation of the system. This is a significant feature, as the process is able to continuously search for new and better protocols without interfering with the normal operation of the system.

An important limitation of the model is that being based on a quasi-random search, it cannot provide any quality guarantee on the short term; for example, we cannot claim that the next generation of protocols will always improve the current one. While guarantees do not exist for the quality of protocol individuals, the overall performance of the system, especially for longer time windows, improves with high probability.

Online protocol evolution is a research subject that is in its infancy at this point. According to our knowledge, there have not been any initiative that resulted in a fully distributed, low-overhead, on-line, genetic programming based protocol evolution framework. Our ideas may be of interest to other researches for theoretic and practical reasons. It may provide insights to the solution of similar problems, especially in the area of ad-hoc, sensor and peer-to-peer networks. Theoretically, the idea is a fertile area for further research, and we hope that numerous interesting aspects and derivations will be introduced in the future.

References

[1] Ahmad Al Hanbali, Mouhamad Ibrahim, Vilmos Simon, Endre Varga, and Iacopo Carreras. A survey of message diffusion protocols in mobile ad hoc networks. In *Proceedings of the 3rd International Conference on Performance Evaluation Methodologies and Tools*, ValueTools '08, pages 82:1–82:16, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[2] Sara Alouf, Iacopo Carreras, Daniele Miorandi, and Giovanni Neglia. Embedding evolution in epidemic-style forwarding. *2007 IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems*, pages 1–6, 2007.
[3] Markus F. Brameier and Wolfgang Banzhaf. *Linear Genetic Programming*. Genetic and Evolutionary Computation. Springer, New York, USA, 2007.
[4] Xiuzhen Cheng, Xiao Huang, Deying Li, and Ding zhu Du. Polynomial-time approximation scheme for minimum connected dominating set in ad hoc wireless networks. *Networks*, 42:2003, 2003.
[5] Michael D. Colagrosso. Intelligent broadcasting inmobile ad hoc networks: three classes of adaptive protocols. *EURASIP J. Wirel. Commun. Netw.*, 2007:25–25, January 2007.
[6] Vincent Cremet. *Foundations for SCALA: Semantics and Proof of Virtual Types*. PhD thesis, EPFL, Lausanne, Switzerland, 2006.
[7] Fei Dai and Jie Wu. Performance analysis of broadcast protocols in ad hoc networks based on self-pruning. *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS*, 15:1027–1040, 2004.
[8] Peter Dittrich, Jens Ziegler, and Wolfgang Banzhaf. Artificial chemistries - a review. *Artif. Life*, 7:225–275, June 2001.
[9] Rajiv Gandhi, Arunesh Mishra, and Srinivasan Parthasarathy. Minimizing broadcast latency and redundancy in ad hoc networks. *IEEE/ACM Trans. Netw.*, 16:840–851, August 2008.
[10] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20:374–387, April 1998.
[11] Yamamoto Lidia, Schreckling Daniel, and Meyer Thomas. Self-replicating and self-modifying programs in fraglets. BIONETICS, Budapest, Hungary, 2007.
[12] K.F. Man, K.S. Tang, and S. Kwong. Genetic algorithms: concepts and applications [in engineering design]. *IEEE Transactions on Industrial Electronics*, pages 519–534, 1996.
[13] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, MobiCom '99, pages 151–162, New York, NY, USA, 1999. ACM.
[14] Vilmos Simon, Márton Bérces, Endre Varga, and László Bacsárdi. Natural selection of message forwarding algorithms in multihop wireless networks. In *Proceedings of the 7th international conference on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, WiOPT'09, pages 16–22, Piscataway, NJ, USA, 2009. IEEE Press.
[15] Lee Spector and Alan Robinson. Genetic programming and autoconstructive evolution with the push programming language. *Genetic Programming and Evolvable Machines*, 3:7–40, March 2002.
[16] Christian F. Tschudin. Fraglets - a metabolistic execution model for communication protocols. In *AINS'03*, AINS'02, Menlo Park, USA, 2003.
[17] E. S. Varga, B. Wiandt, B. K. Benko, and V. Simon. *Biologically Inspired Networking and Sensing: Algorithms and Architectures*. IGI Books, 2012.
[18] Brad Williams and Tracy Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '02, pages 194–205, New York, NY, USA, 2002. ACM.
[19] Jie Wu and Hailan Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*, DIALM '99, pages 7–14, New York, NY, USA, 1999. ACM.

E. S. Varga is a PhD candidate at the Budapest University of Technology and Economics. He received his MSc degree in computer science in 2006. His main interests are discrete-event simulations and complex systems.

B. Wiandt is a PhD student at the Budapest University of Technology and Economics. He received his BSc degree in computer science in 2010. His main interests are self-organizing networks and genetic algorithms.

B. K. Benkő was a research fellow at Budapest University of Technology and Economics. She received her MSc degree in computer science in 2003. Her research interests include data mining, machine learning, and distributed and self-organizing systems.

V. Simon is an associate professor at Budapest University of Technology and Economics. He received his MSc degree in 2003 and PhD degree in 2009. His research interests include self-organizing and adaptive networks, evolution of communication protocols, opportunistic and delay-tolerant networks and mobility management in 3G- 4G mobile systems.

Big Data: at the Crossroad of Technology, Business and Regulation

Roberto Saracco, *Senior Member, IEEE*

Abstract – **The technological evolution in the last 20 years has steadily increased the amount of data in digital form, sensed from the environment, produced by private companies or collected from individuals' activities. Communications becoming more and more pervasive make these data available to any application, as if they were in a single database. The real value comes from information and services derived from data correlation, tailored to specific interest. We claim that the availability of a Living Open Data Framework enables the exploitation of data through services, created by many parties, in several contexts. However, data correlation is fraught with perils, from the obvious of privacy breaching to more subtle ownership and value protection. These issues are of fundamental importance and have to be addressed to create a viable business based on data and require advances in technology and in the regulatory framework. This is what is being pursued by the Italian EIT ICT Labs Trento Node through the cooperation of several parties, including the Autonomous Province of Trento, Telecom Italia, and several universities and SMEs. The paper aims at reporting the directions and the results so far obtained.**

Index Terms — **Data Analysis, Data Correlation, Open Data**

I. INTRODUCTION

Last year researchers at Stanford published a study on the analyses of millions of medical records of USA patients indicating that the use of two drugs, one for lowering cholesterol level and the other fighting depression, led to higher blood glucose levels [1]. Spotting this was quite easy, while determining this effect through clinical studies would have been almost impossible.

In fact, how could a pharmaceutical company clinically test all possible outcomes from all possible use of drugs? There are thousands of them and the possible permutations of concurrent use go up in the millions.

In a nutshell, this is the power of Big Data: their correlation and analyses generate valuable information.

If the potential is evident, the way to exploit it is not straightforward: on the one hand we have more and more data being produced but not all of them are readily available nor through secure channels. This is mostly due to a cultural resistance of data producers to share their data because so far data ownership has been considered a source of control: it is true for Telco companies as well as for single individuals. On the other hand the power provided by correlation can be misused, can affect privacy of individuals, the value created is often unbalanced among the data chain stakeholders. The latters have recently raised new concerns about data ownership and value redistribution.

Manuscript received 13 March, revised 24 March, accepted for publication 31 March.

R. Saracco is with EIT ICTLabs Trento Node, 38123 Trento, Italy.
E-mail: roberto.saracco@ictlabs.eu

Technology is progressing rapidly, both in the creation and in the correlation of data, with distributed computing enabling the processing of huge amount of data, unthankable just five years ago. Riding many positive forecasts, business is endeavoring to find ways to exploit the potential offered but regulation is lagging behind.

In the following we provide a roadmap to address Big Data, describing the context and the activities carried out in the Trentino area within the EIT ICT Labs framework.

II. IT'S BIG, AND IT'S GETTING BIGGER

The amount of data being created every day is staggering. Only in Italy we have exceeded 3.5 billion data records that can be harvested every day, including call data records, digital power meters, mailing and banking transactions, health care prescriptions, traffic monitoring, security camera feeds, ambient sensors, etc.

This amount is growing every day, as more and more sensors are deployed, individuals track their actions in a digital form, the fabric of commerce turns to digital transactions (mobile payment substituting cash), health care becomes heavily personalized and requires continuous monitoring, safety and energy concerns move the focus from vehicles to infrastructure and so on.

Our expectation is that by the end of this decade Italy alone will be producing 10 billion "accessible" transactions a day.

This figure may not still considered Big Data, if one compares it to the TB generated by the LHC for a single experiment. But looking at all the correlations generated over those multi-source data, the amount of data can be compared in computational complexity with the ones of the LHC or the KAT-7 Radio Telescope.

The following are some application scenarios we envision, where big data and correlation of multi-data-sources go hand in hand.

A. Our life in bits

My Life in Bits was a research program initiated by Microsoft a few years ago. It assumed that our individual life could be recorded in digital form. At that time the focus was on the creation of devices that would capture the various moments of our life and on the storing and retrieval of those moments.

Today, capturing in real time what we see, what we hear, what we do is no longer a challenge. Products like Pivot Head let us wear a pair of glasses that embed a camcorder, so tiny that you won't even notice it, at a cost that compares to the one of some cool sun-glasses. Data storage is no longer a challenge either.

Sorting out and extracting knowledge from the collected data: this is now the challenge that research is taking up.

Recording our life in bits create a sort of black box that can be used for a variety of purposes, from helping to remember, to increase the effectiveness of education and training, to increase safety.

But the records of our life can also be used as the starting point for correlating other data and creating information. You've got a running nose. An analyses of the data recorded 2 days ago, correlated with data from the place you have been (temperature, other people health status, ambient particulate,...) may pinpoint the probable cause and single out a normal (annoying) cold from a potential serious disease at a stage when prompt actions reduce risks (and cost).

B. The rise of digital transport infrastructures

Cars are becoming a bunch of computers on wheels: entertainment of passengers, fuel consumption optimization software, safety control systems, etc., making use of tens of sensors and computers that continuously monitor the vehicle and sometimes share the data with the outside world.

Actually, much more in terms of safety and energy consumption can be done if data are shared and correlated to create a digital transport infrastructure. The infrastructure can talk to the car navigator to get information on the car destination (eventually it may become compulsory to use the navigator and input a "drive plan" as pilot are requested to input a "flight plan") and based on information received from all cars on the road and road conditions (man at work, accidents, weather hazard...) can instruct the engine to drive the car at a certain speed. No need to hurry to find yourself in a queue sooner!

In the future cars (people) may pay a variable road toll (like some "pay as you go" profile that insurance broker already offer), with higher rates charged for specific itineraries and offer of cheaper solution on the fly. They may also opt for even higher traffic fees if they want to get priority routing....

By the way, being aware of how much a trip would cost you at that particular time of the day (taking into account driving time, fuel consumption and taxes) may change our driving behavior.

C. Understanding a living environment

Many of today's products are born with several sensors embedded. Be it a toaster (heath sensor), a television (light sensor) a lamp (proximity sensor), an Mp3 player (accelerator sensor), a digital camera (light, accelerometer sensors) ... and more are coming. These sensors are used for "local" action (like detection of a smiling face for a camera or the dusk for a street lamp) but they may, in principle, provide data to the world, via Internet. Data are coming from sensors in orchards to warn on the presence of harmful bugs, they come from tunnels if temperature of a passing car is too high, from switches if it gets too cold and ice may become a danger, from water reservoirs, rivers and lake warning for pollutants...

Through these data and through their correlation we are getting a better understanding of the environment, and we can start to take action sooner, decreasing overall cost, increasing safety and environment protection.

D. Just in time production

The digitalization of the supply and delivery chains, along with the flexibility offered by robotized production is changing the way we think about factories. The growing percentage of software in products further contributes to a shift in the production paradigm.

Monitoring sales leads to change the production in real time mostly for local productions and for some kind of goods. If something sells better in blue, then that is the color for the next batch. These kinds of information are usually inferred by a posteriori analysis of reports and surveys that companies produce starting from their data. "Just in time" production requires the synchronization of many processes, and in turns the correlation of many data. Just few years ago these data where the result of guessing, now they are hard data reflecting what is going on.

A more complex and more efficient analysis that correlated multiple data source such as the mobile payment, aired ads, blogs and tweets, mood derived from security cameras, etc. let companies have a deeper understand of customers' needs and preferences in real time. This provides plenty of "hints" to design marketing strategies in real-time.

E. Society Well being

Defining and understanding the well-being degree in a society is not easy because it is often the result of multiple factors. Although poverty is not a basic recipe for happiness, wealth is not a guarantee at all: European northern countries are considered leaders of good services and social support, although often they are plagued with a high suicide rate, an indicator of a general discomfort.

The availability of multi-source Big Data enables the studies of the factors and the correlation that affect the well-being.

The supporting actions are focused on people with specific disabilities or diseases e.g. Autisms (or more generally ASD). This is essential but for every person with ASD you should take into account two parents, an educator (or more), colleagues and so on. It is this fabric, or social network, that actually has to be considered and there is a need to discover the relevant social network and work on improving its well-being.

Discovering social networks and sensing the mood of a social network is now within the realm of possibilities and much information can be derived from data analyses.

III. TECHNOLOGY EVOLUTION

Technology and Big Data interplay in many different ways.

Collection. Sensors of many kind transform changes in the environment into digital data, that can be "stamped" with origin (localization) time, identity; actions, like paying with your phone, writing a prescription, generate data; photographing, filming generate data. All of this is getting easier and cheaper thanks to technology evolution.

Transport & Storage. Data have to be transported from the point of generation to the point of storage, and pervasive communications is making this ever more effective. Storage can, to a certain extent, be considered an integral part of the communications technology, particularly after the rising of cloud technologies.

Mining. Stored data need be retrieved and processed. Here, Data Mining and modeling technologies play the lion's share. Data mining had its first presence in the spot light in the nineties, in conjunction with artificial intelligence studies. Today, after some time in the twilight Data Mining again is taking the forefront with a radically different approach. It is no longer based on Artificial Intelligence but on statistical analyses. Clearly this is made possible by Big Data, and, in turns, stimulate the attention on Big Data.

Visualization. The data visualization plays a crucial role in our understanding the hidden semantics of Big Data. Availability of graphic displays, and in perspective of haptic interfaces, let us immerse in data, feel them and physically interact with them. A good part of these progresses is enabled by increased processing capacity and lower cost.

Control. Data, and the derived information, are fraught with potential mismanagement. Privacy is clearly at the forefront of concerns every time we deal with personal data. Correlation makes these concerns even bigger by circumventing normal safeguards, like anonymity. Ownership comes second. If data can generate value shouldn't their owner be entitled to some sort of compensation? Digital rights management didn't work very well for music, it is even more difficult in case of data management and correlation. What if the sources of data are public (freely available) and their correlation is able to generate significant value. Is this value to be reaped by the one who made the correlation or should it extend to the ones whose data made correlation possible?

Technology evolution helps addressing these issues, although it often creates new issues, since more sophistication in correlation worsen many of the above listed issues.

IV. TRENTINO OPEN LIVING DATA (TOLD) FRAMEWORK

As it emerges from the above statements, exploiting the potential of Big Data raises a lot of issues, that technology can help to fix, but alone it is not enough. What we need is a "framework" that, relaying on technology and innovative solutions for big data management, takes into account the related process innovation opportunities, as well as the stakeholders' role and constraints, the business potential and so on.

We talk about *Living* and *Open* data: Living data because they are real-time, geo-localized and Open data because free to use, reusable and distributable. For sure, technology evolves faster than the regulatory framework, and the differences among different frameworks are exploited by players in this field, particularly those that are subject to lower constraints.

The institutions of the Trentino Region have taken the decision of managing the territorial data in an open fashion and have created an Open Living Data Framework, not unique but surely one of the first to approach this issue and develop a "policy" on open data.

This goes along with the statement given on December 12th, 2011, by the European Vice President Neelie Kroes "The best way to get value from data is to give them away". Indeed, Open Public Data are expected to boost EU's economy by 40 billion euros each year and the EU is investing 100 million euros in the 2011-2013 period to fund research on data handling technologies. And these estimated are referring just to data owned by public administrations. Just imagine the boost to the economy if also private companies would make their data available.

This is actually the aim of the Open Living Data Framework set up by the Trento Region: take the lead to steer private companies to jump on the bandwagon of Open Data by releasing their own data to get back value from information and services deriving it from multi-source data correlation. This should overcome the general difficulty of other Open Data initiatives to get momentum due to a scarce involvement of private subjects, to the lack of a defined semantics of the available data and the need of common uniformed format of data exchange. Since Trento satisfies the requisites for this framework, the idea is to use this (relatively small) pilot area to apply and verify the approach and then extend it at Country level.

The regulation provides the legal framework onto which data can be opened, and in parallel funding is given to research technologies to solve issues related to the opening of data. The Telecom Italia research lab, SKIL, is part of this global approach.

At the time of writing, April 2012, a first core of Companies have joined this initiative making available (part of) their data. Among them Telecom Italia, providing billions of anonymized call data records every month, the local highway management company, providing in and out of vehicles through its highway system, the local public transportation authority, providing hundreds of thousands of trip data per month, the Autonomous Province Public Authority, providing ambient sensors data, environmental geo data and safety related statistical data, the Area Electric Power Company, providing individual smart metering data every day, the National Postal Service, providing money transaction and goods delivery data.. The first result has been achieved: having the commitment of private entities to share their data within this initiative means having broken the first cultural reluctance that made most of similar

Data are going to be collected through several channels, each provided in its own format and subject to the constraints decided by the owner. That makes them difficult to use and requires the system to bring them in a common shared format.

In the Open Living Data framework SKIL [2], the aforementioned Telecom Italia Research Lab focused on semantic annotation and knowledge extraction, will be delegated to convert the raw data stream into a semantically annotated data stream and to correlate the data streams. The generated correlations are made accessible by the open API, thus decoupling the original raw data from the actual use.
This is one of the tricks being used to neutralize data, not just at the point of origin (this is done by the data provider) but also at the point of usage. This neutralization is a fundamental part of the Open Living Data Framework.

Correlation is also based on other nation-wide data provided by ISTAT, the Italian Institute of Statistics, a government organization.

The Trentino Open Living Data Project is moving the first steps towards the direction of exploring how to get value from multi-source Big Data and how this can be leveraged by the different stakeholders. Such kind of initiatives need to be "socialized" with the Territory, in order to let the citizens provide feedbacks and be familiar with the importance of open data. We planned to achieve this by visualizing the results of a number of correlations on big screens in places where people are likely to roam.

Example visualizations are thematic maps, such as those on car accidents, where people can easily detect the most dangerous areas, understand the underlying factors and eventually see the actions taken by the public administration to decrease the risk.

Other maps could show cultivated areas, their yield and the increase in productivity made possible by understanding negative factors impacting the products; others showing the spikes of some diseases and correlation points out probable causes.

V. CREATING BUSINESS OPPORTUNITIES

The framework provided by Trentino Open Living Data will be proposed as the starting point to generate new business through SMEs (Small Medium Enterprises). This is where the EIT ICT Labs [3], of which Trento is one of the six nodes.

The ICT Labs is a European Initiatives aiming at stimulating European competitiveness by fostering innovation and helping SMEs to put successful products into market. This is done using in an auto-sustaining way the tools of education, research and business innovation.

More specifically, in the context of Big Data, there is a need for research to create the required tools for data neutralization, security, privacy, authentication, accountability and value tracking. There is a need to support SMEs to manipulate those date and create services that are conforming to European and local regulation, to submit patent and to commercialize the product. This latter is usually a component in a complex network and through its wide reach ICT Labs can be a winning partner for these enterprises.

Finally, there is a need for high level education in this area, engineers need to understand what wealth can be leveraged out of Big Data and what are the issues related to that.

It is often voiced the concern of an imbalanced situation once comparing the strict rules usually imposed on European Companies when dealing with data, compared with the much looser ones imposed in other world areas. This generates, it is claimed, unfair competition. However, the more stringent rules applied in Europe can be turned to market competitive advantage, since they provide a context for better security and trust and these have commercial value.

SMEs can work at different level in the Open Living Data Value Chain; actually they usually operate at the Open Data Ecosystem level, by acting independent of one another and all together increasing the overall value and biz opportunities.

Data are pervasive and support many application areas, from Digital Cities to Smart Spaces, from Health Care and Well Being to Intelligent Transportation Systems. These are all areas addressed in the EIT ICT Labs. Furthermore, the area Security, Privacy and Trust is right on the spot to help tackling and leveraging Big Data, acting as a catalyst to speed up innovation in the afore mentioned areas.

Innovation is expected both in correlation (finding meaning in the data) and in the presentation of correlation result (visualization on a variety of media, big screens, cell phones, television, kiosks...) and each innovation is basically a service that can generate an attractive biz for an SME.

VI. CONCLUSIONS

We have presented a case for working on Big Data, by saying that technology evolution has already provided us with huge amount of data and this amount will increase in the future. Technology is also providing us means to deal with correlation, and this is the place where value is generated, and to deal with issues connected to data, like privacy, ownership, accountability.

We have also indicated that a strong steering from an institutional and political entity is required and this is what is happening in the Trentino Region.

Furthermore, the Big Data is a platform to sustain a new economy where the wealth is mined from data analyses and usage, a wealth that requires both a "data infrastructure" usually provided by big companies, and many SMEs creating and delivering services.

What we did not say, and we are now doing, is that Big Data leverage results in better life for each of us, for the citizen of the Information Society.

Correlation makes us aware of phenomena that are normally hidden, explains behavior in terms of causes and let us take corrective actions when needed.

The Information Society delivers at a much lower cost than the Industrial Society (that was in turn delivering at a lower cost than the agricultural and artisan society). This lower cost means lower barriers to enter business, hence more people entering the business as entrepreneurs.

We need to respond to these opportunities by preparing our students to become entrepreneurs, education is a fundamental stepping stone. Research is another fundamental building block, particularly research that can act as catalyst for the development of business.

EIT ICT Labs ambition is to be a main player in this context. It is in its infancy, since it has started in 2011 (the Trento node started on January 1st, 2012) so the jury is still out. For sure the commitment and enthusiasm is very high.

ACKNOWLEDGEMENT

The contribution of ideas and text revision by Fabrizio Antonelli, Director of the SKIL Lab is gratefully acknowledged.

REFERENCES

[1] http://www.technologyreview.com/business/38484/?nlid=nldly&nld=2011-09-19

[2] http://skil.telecomitalia.com/

[3] http://eitictlabs.eu

Open Data:
at the Crossroad of Technology, Business and Regulation

Roberto Saracco is the President of EIT Italy (European Institute for Innovation and Technology) and Node Director of EIT ICTLabs Italy, based in Trento, one of the 6 European Labs set up by the European Commission to transform research results in the Information and Telecommunication areas into market innovation. Up to December 2011 he was the Director of the Future Centre in Venice, responsible for innovative telecommunications architectures and scientific communications reporting directly to the Strategy Officer of Telecom Italia. Roberto chaired the Visionary Group (1996-1997) on Super Intelligent Networks to steer the co-operative research at the European Union (EU) level beyond the year 2000. He has recently served as member of the Internet 2020 Strategy Group and European Research Network (GEANT) expert group. In the eighties, Roberto led research in Telecommunications Management in CSELT, and actively participated in standardization activities at CCITT, and in a number of international standardization organizations including OSI, ETSI and T1M1. His leadership includes chairing an EU-level group for planning, leading European research activities in the area of software technologies, and the EURESCOM group in designing the framework for European co-operation on TMN. He has published over 100 papers in journals and magazines, six books – including "The Disappearance of Telecommunications," which was published in the USA by IEEE press, He has also delivered speeches and keynotes at many international conferences. He currently lectures at the Turin Polytechnic on the aspects of multimedia and telecommunications.
He is a senior member of IEEE that he joined over 20 years ago. In the last 15 years he has held several leading roles and conducted a number of DLTs and DSPs. Currently he is the Director of the Sister and Related Societies of COMSOC. Previously he has served as VP of Member Relations, Director of Marketing, Chair of the CNOM and Enterprise Management TCs.

Call for Papers – Special Issue on Quantum Communications

Although quantum computers are going to be the applications of the far future, there are already a few algorithms to solve problems which are very difficult to handle with traditional computers. Quantum computing is based on various quantum effects in physics and offers revolutionary solutions for different problems e.g., prime factorization, searching in unsorted database, key distribution and information coding. The power of quantum parallelism allows us to solve classically complex problems, and the quantum entanglement leads to quantum communication algorithms like teleportation and super dense coding. The quantum cryptography provides new ways to transmit information with unconditional security by using different quantum key distribution protocols. The special issue of Infocommunications Journal will focus on quantum communications with the following scope:

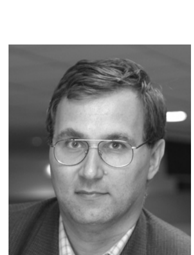
- Quantum Communications
- Quantum Interferometry and Quantum Sensors
- Entanglement as a Resource of Quantum Technology
- Quantum Cryptography
- Quantum Processors and Computers Design
- Quantum Programming Languages and Semantics

Authors are requested to send their manuscripts via electronic mail to László Bacsárdi at *bacsardi@hit.bme.hu* until 30 September, 2012. The manuscripts should follow the IEEE format with maximum length of 8 journal pages.

Guest Editors:

OZGUR B. AKAN (M'00–SM'07) received the B.S. and M.S. degrees in electrical and electronics engineering from Bilkent University and Middle East Technical University, Ankara, Turkey, in 1999 and 2001, respectively, and the Ph.D. degree in electrical and computer engineering from the Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, in 2004.
He is currently a Professor with the Department of Electrical and Electronics Engineering, and the Director of Next-generation and Wireless Communications Laboratory (NWCL), Koc University, Istanbul, Turkey. His current research interests include wireless communications, acoustic communications, nano communications, quantum communications and information theory. Dr. Akan is an Associate Editor for the IEEE Transactions on Vehicular Technology, the International Journal of Communication Systems (Wiley), the European Transactions on Telecommunications, and the Nano Communication Networks Journal (Elsevier). He served as an Editor for ACM/Springer Wireless Networks (WINET) Journal from 2004 to 2010, as an Area Editor for AD HOC Networks Journal (Elsevier) from 2004 to 2008, as a Guest Editor for several special issues. He currently serves as the General Co-Chair for ACM MobiCom 2012, General Co-Chair for IEEE MoNaCom 2012, and TPC Co-Chair for IEEE ISCC 2012. He is the Vice President of the IEEE Communications Society – Turkey Section. He is a Senior Member of the IEEE Communications Society (COMSOC), and a member of ACM. He is a COMSOC Distinguished Lecturer (2011–2012).
He received the IEEE COMSOC Outstanding Young Researcher Award for EMEA Region 2010 (as runner-up), the IBM Faculty Award twice in 2010 and 2008, and the Turkish Academy of Sciences Distinguished Young Scientist Award 2008 (TUBA-GEBIP).

LÁSZLÓ BACSÁRDI obtained his MSc degree in computer engineering from Budapest University of Technology and Economics (BME) in 2006. He holds an assistant professor position at the University of West Hungary. He wrote his PhD Thesis on the possible connection between space communications and quantum communications at the BME Department of Telecommunications. His current research interests are in mobile ad hoc communication, quantum computing and quantum communications. He is Secretary General of the Hungarian Astronautical Society (MANT), which is the oldest Hungarian non-profit space association founded in 1956. He is member of the board of a Hungarian scientific newspaper ('World of Nature') and he is the publisher of a non-profit Hungarian space news portal ('Space World'). Furthermore he is member of IEEE and HTE. He has joined the Space Generation Advisory Council (SGAC) as well, and is currently active as the Hungarian National Point of Contact.



SÁNDOR IMRE was born in Budapest in 1969. He received the M.Sc. degree in Electrical Engineering from the Budapest University of Technology (BUTE) in 1993. Next he started his Ph.D. studies at BUTE and obtained Dr.Univ. degree in 1996, Ph.D. degree in 1999 and DSc degree in 2007. Currently he is carrying his activities as a Professor and a Head of Department of Telecommunications at BUTE. He is a member of Telecommunication Systems Committee of the Hungarian Academy of Sciences. He participates in the Editorial Board of two journals: Infocommunications Journal and Hungarian Telecommunications. He was invited to join the Mobile Innovation Centre as R&D director in 2005. His research interests includes mobile and wireless systems. His main research interests and contributions are in the areas of various wireless access technologies, mobility protocols and reconfigurable systems.

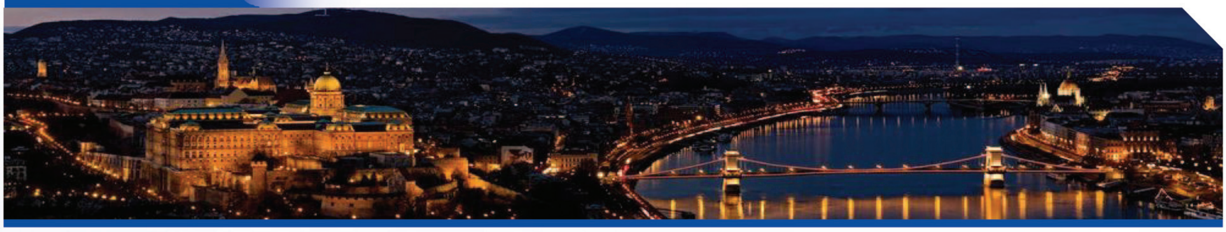
IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS

IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS ICC 2013

Bridging the Broadband Divide
9-13 June • Budapest, Hungary

IEEE | IEEE COMMUNICATIONS SOCIETY

WWW.IEEE-ICC.ORG/2013



CALL FOR PAPERS

The 2013 IEEE International Conference on Communications (ICC) will be held in the vibrant city of Budapest, Hungary from 9 – 13 June 2013. This flagship conference of IEEE Communications Society aims at addressing an essential theme on "Bridging the Broadband Divide." The conference will feature a comprehensive technical program including several Symposia and a number of Tutorials and Workshops. IEEE ICC 2013 will also include an attractive expo program including keynote speakers, various Business, Technology and Industry fora, and vendor exhibits. We invite you to submit your original technical papers, industry forum, workshop, and tutorial proposals to this event. Accepted and presented papers will be published in the IEEE ICC 2013 Conference Proceedings and in IEEE Xplore®. Full details of submission procedures are available at http://www.ieee-icc.org/2013.

To be published in the IEEE ICC 2013 Conference Proceedings and IEEE Xplore®, an author of an accepted paper is required to register for the conference at the full member or non-member rate and the paper must be presented at the conference. Non-refundable registration fees must be paid prior to uploading the final IEEE formatted, publication-ready version of the paper. For authors with multiple accepted papers, one full registration is valid for up to 3 papers. Accepted and presented papers will be published in the IEEE ICC 2013 Conference Proceedings and in IEEE Xplore®.

PLANNED TECHNICAL SYMPOSIA

Selected Areas in Communications Symposium

E-Health Area
Pradeep Ray, University of New South Wales, Australia

Power Line Communications Area
Andrea Tonello, University of Udine, Italy
Stephan Weiss, University of Strathclyde, UK

Smart Grids Area
Bahram Honary, Lancaster University, UK

Tactical Communications & Operations Area
Gabe Jakobson, Altusys, USA

Satellite & Space Communication Area
Hiromitsu Wakana, NICT, Japan

Data Storage Area
Tiffany Jing Li, Lehigh University, USA

Access Systems and Networks Area
Michael Peeters, Alcatel-Lucent, Belgium

Green Communication Systems and Networks
Athanassios Manikas, Imperial College London, UK

Wireless Communications Symposium
Zhaocheng Wang, Tsinghua University, China
Metha B. Neelesh, Indian Institute of Science, India
Hanna Bogucka, Poznan University of Technology, Poland
Fredrik Tufvesson, Lund University, Sweden

Wireless Networking Symposium
Azzedine Boukerche, University of Ottawa, Canada
Pan Li, Mississippi State University, USA
Min Chen, Seoul National Unversity, Korea

Communication Theory Symposium
David Gesbert, EURECOM, France
Angel Lozano, Universitat Pompeu Fabra, Spain
Velio Tralli, University of Ferrara, Italy
Sennur Ulukus, University of Maryland, USA

Signal Processing for Communications Symposium
Hai Lin, Osaka Prefecture University, Japan
Octavia Dobre, Memorial University, Canada
Saiid Boussakta, Newcastle University, UK
Hongyang Chen, Fujitsu Laboratories, Japan

Optical Networks and Systems Symposium
Xavier Masip-Bruin, Technical University of Catalonia, Spain
Franco Callegati, University of Bologna, Italy
Tibor Cinkler, Budapest University of Technology and Economics, Hungary

Next-Generation Networking Symposium
Malathi "MV" Veeraraghavan, University of Virginia, USA
Joel Rodrigues, University of Beira Interior, Portugal
Wojciech Kabacinski, Poznan University of Technology, Poland

Communication QoS, Reliability & Modeling Symposium
Tetsuya Yokotani, Mitsubishi Electric Corporation, Japan
Harry Skianis, University of the Aegean, Greece
Janos Tapolcai, Budapest University of Technology and Economics, Hungary

Ad-hoc and Sensor Networking Symposium
Guoliang Xue, Arizona State University, USA
Abdallah Shami, University of Western Ontario, Canada
Xinbing Wang, Shanghai Jiaotong University, China

Communication Software and Services Symposium
Jiangtao (Gene) Wen, Tsinghua University, China
Lynda Mokdad, University Paris-Est, France

Communication and Information Systems Security Symposium
Tansu Alpcan, TU Berlin, Germany
Mark Felegyhazi, Budapest University of Technology and Economics, Hungary
Kejie Lu, University of Puerto Rico at Mayagüez, PR

Cognitive Radio and Networks Symposium
Honggang Zhang, Zhejiang University, China
David Grace, University of York, UK
Andrea Giorgetti, University of Bologna, Italy

COMMITTEE

General Chair:
Christopher Mattheisen
Magyar Telekom, Hungary

Executive Chair:
Lajos Hanzo
University of Southampton, UK

Technical Program Chair:
Andreas Molisch
University of Southern California, USA

Technical Program Vice-Chairs:
Andrea Conti
University of Ferrara, Italy
Iain Collings
CSIRO ICT Centre, Australia

Workshops Co-Chairs:
Thomas Michael Bohnert
Zurich University of Applied Sciences, Switzerland
Christoph Mecklenbrauker
Vienna University of Technology, Austria
Christina Fragouli
EPFL, Switzerland

Tutorials Co-Chairs:
Marco Chiani
University of Bologna, Italy
Wei Chen
Tsinghua University, China

Panel Session Co-Chairs:
David Soldani
Huawei, Germany
Peter Rost
NEC Labs Europe, Germany

Publications Co-Chairs:
Dong In Kim
Sungkyunkwan University, Korea
Peter Mueller
IBM Zurich Research Laboratory, Switzerland

Conference Operations Chair:
Rolland Vida
Budapest University of Technology and Economics, Hungary

Patronage Chair:
Roland Jakab
Ericsson, Hungary

Keynotes Co-Chairs:
Doug Zuckerman
Telcordia, USA
Gerhard Bauch
Universität der Bundeswehr München, Germany

Finance Chair:
Peter Nagy
HTE, Hungary

Publicity Chair:
John Vig
IEEE, USA

Local Arrangements Chair:
Nandor Matrai
Asszisztencia, Hungary

IMPORTANT DATES

Paper Submission	Acceptance Notification	Camera-Ready	Tutorial Proposal	Workshop Proposal:	Business Forum Proposal
16 September 2012	27 January 2013	24 February 2013	7 October 2012	18 March 2012	8 April 2012

SCIENTIFIC ASSOCIATION FOR INFOCOMMUNICATIONS

Who we are

Founded in 1949, the Scientific Association for Infocommunications (formerly known as Scientific Society for Telecommunications) is a voluntary and autonomous professional society of engineers and economists, researchers and businessmen, managers and educational, regulatory and other professionals working in the fields of telecommunications, broadcasting, electronics, information and media technologies in Hungary.

Besides its more than 1300 individual members, the Scientific Association for Infocommunications (in Hungarian: Hírközlési és Informatikai Tudományos Egyesület, HTE) has more than 60 corporate members as well. Among them there are large companies and small-and-medium enterprises with industrial, trade, service-providing, research and development activities, as well as educational institutions and research centers.

HTE is a Sister Society of the Institute of Electrical and Electronics Engineers, Inc. (IEEE) and the IEEE Communications Society. HTE is corporate member of International Telecommunications Society (ITS).

What we do

HTE has a broad range of activities that aim to promote the convergence of information and communication technologies and the deployment of synergic applications and services, to broaden the knowledge and skills of our members, to facilitate the exchange of ideas and experiences, as well as to integrate and harmonize the professional opinions and standpoints derived from various group interests and market dynamics.

To achieve these goals, we…

- contribute to the analysis of technical, economic, and social questions related to our field of competence, and forward the synthesized opinion of our experts to scientific, legislative, industrial and educational organizations and institutions;
- follow the national and international trends and results related to our field of competence, foster the professional and business relations between foreign and Hungarian companies and institutes;
- organize an extensive range of lectures, seminars, debates, conferences, exhibitions, company presentations, and club events in order to transfer and deploy scientific, technical and economic knowledge and skills;
- promote professional secondary and higher education and take active part in the development of professional education, teaching and training;
- establish and maintain relations with other domestic and foreign fellow associations, IEEE sister societies;
- award prizes for outstanding scientific, educational, managerial, commercial and/or societal activities and achievements in the fields of infocommunication.

Contact information

President: **DR. GÁBOR HUSZTY** • *ghuszty@entel.hu*

Secretary-General: **DR. ISTVÁN BARTOLITS** • *bartolits@nmhh.hu*

Managing Director, Deputy Secretary-General: **PÉTER NAGY** • *nagy.peter@hte.hu*

International Affairs: **ROLLAND VIDA, PhD** • *vida@tmit.bme.hu*

Addresses

Office: H-1055 Budapest, V. Kossuth Lajos square 6-8, Room: 422.

Mail Address: 1372 Budapest, Pf. 451., Hungary

Phone: +36 1 353 1027, Fax: +36 1 353 0451

E-mail: *info@hte.hu*, Web: *www.hte.hu*