Table 1: Summary of combined KF and CMA algorithm

<table>
<thead>
<tr>
<th>Process</th>
<th>Input(s)</th>
<th>Output(s)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tracking channel</td>
<td>$d(t)$, $x(t)$</td>
<td>$h(t)$</td>
<td>KF channel estimator with variable stepsize</td>
</tr>
<tr>
<td>Initialise CMA equalisers</td>
<td>$h_0(t)$</td>
<td>$w_0(t)$</td>
<td>$i = 0, ..., CL + EL - 2$</td>
</tr>
<tr>
<td>Equalise received signal</td>
<td>$x(t)$, $w(t)$</td>
<td>$y_{i}(t+1)$</td>
<td>CMA equalisers with variable stepsize</td>
</tr>
<tr>
<td>Select and direct decision</td>
<td>$y_{i}(t+1)$</td>
<td>$d_{i}(t+1)$</td>
<td>Select output whose amplitude is closest to expected value</td>
</tr>
</tbody>
</table>

The measurement signal is equalised by the proposed algorithm. The estimated channel and equaliser length are two and four, respectively. To show the improvement in the performance of the proposed algorithm against that of the conventional CMA algorithm, the output of the proposed algorithm will be compared with that of the 4-tap conventional CMA equaliser.

![Fig. 3 Amplitudes of outputs of equalisers](image)

*Fig. 3 Amplitudes of outputs of equalisers*

<table>
<thead>
<tr>
<th>a CMA</th>
<th>b CMA &amp; KF</th>
</tr>
</thead>
</table>

![Fig. 4 Difference of transmitted and reconstructed signals](image)

*Fig. 4 Difference of transmitted and reconstructed signals*

<table>
<thead>
<tr>
<th>a CMA</th>
<th>b CMA &amp; KF</th>
</tr>
</thead>
</table>

References:


Discrete-time buffers with variable-length train arrivals

S. Wittevrongel

An infinite capacity single server queueing system is considered, to which messages consisting of a variable number of fixed-length packets arrive at the rate of one packet per slot ('train arrivals'). Assuming geometrically distributed message lengths, an exact closed-form expression for the probability generating function (pgf) of the buffer contents is obtained.

**Introduction:** Asynchronous transfer mode (ATM) networks are expected to support a wide variety of telecommunication services. If the ATM networks are to be used as transport networks for currently existing data services, it is necessary to segment the large external data frames at the edge of the ATM network into small internal fixed-length cells, which are then transmitted through the network. A consequence of this segmentation is the presence of correlation in the cell arrival streams.

To study the impact of this correlation, we consider in this Letter a discrete-time queueing system with an infinite capacity buffer and one single output line. Data arrive at the buffer in the form of variable-length messages consisting of one or more fixed-length packets. In the model, the generic term 'message' denotes the external data format, whereas the term 'packet' is used for the internal format. It is assumed here that the message lengths (in terms of packets) are independent geometrically distributed random variables with common probability mass function (pmf):  \[ p(k) = (1 - \alpha)^{k-1} \alpha, \quad n \geq 1 \] (1)

Also, the time axis is assumed to be divided into fixed-length slots, such that one slot suffices for the transmission of exactly one packet from the buffer. The transmission of packets always starts (and ends) at slot boundaries. Messages enter the buffer at the rate of one packet per slot, which is denoted by the term 'train arrivals'. Note that a message of length $n$ causes packet arrivals in the buffer during $n$ consecutive slots, which results in a correlated packet arrival process. Finally, the numbers of new messages generated during the consecutive slots are modelled as i.i.d. random variables with common probability generating function (pgf) $M(z)$.

The above queueing system has been studied before in [1]. However, the analysis method described in [1] only leads to expressions for the mean and the variance of the buffer contents. In this Letter, an analytical technique is presented to obtain the pgf of the buffer contents, from which, besides the moments, the tail distribution of the buffer contents -- a very important performance measure -- can also be derived. For a survey of various related
arrival models the reader is referred to [1]. Some results of [1], which are needed for a proper understanding of the present Letter, are summarised below.

Preliminary results: Let $c_k$ denote the total number of packets arriving in the buffer during slot $k$, and let $m_k$ be the number of newly generated messages during slot $k$. Then the following relationship exists:

$$c_k = m_k + \sum_{i=1}^{\alpha_k-1} c_i$$

The $c$s are independent Bernoulli random variables, which are equal to 0 or 1 with probabilities $1-\alpha$ or $\alpha$, respectively, owing to the assumption of geometrically distributed message lengths. The quantity $m_k$ in eqn. 2 has pgf $M(c)$.

Next, let the random variable $s_k$ represent the system contents, i.e. the total number of packets stored in the buffer including the possible packet in transmission, at the start of slot $k$. Then the following system equation holds:

$$s_{k+1} = (s_k - 1)^+ + c_k$$

where $(\cdot)^+ = \max(\cdot, 0)$. From eqns. 2 and 3, it is clear that the set $(\bar{c}_1, \bar{s}_1)$ forms a two-dimensional Markov chain. It is assumed that the queuing system can reach a steady state and $(\bar{c}_s, \bar{s}_s)$ is used to denote the steady-state version of $(\bar{c}_s, \bar{s}_s)$. To analyse the buffer behaviour, the joint pgf of the random variables $\epsilon$ and $s$ is defined as

$$P(x, z) = E[x^z \cdot s^z]$$

where $E[\cdot]$ is the expected value of the argument between square brackets. Using standard techniques for $z$-transforms, it is obvious that the following fundamental functional equation has been established for $P(x, z)$:

$$P(x, z) = \frac{M(z)}{z} \left\{ \frac{P(1-\alpha + \alpha x z, z) + (z-1)(1-\rho)}{z} \right\}$$

where $\rho$ denotes the mean number of packet arrivals per slot. Furthermore, the pgf $E(x)$ of the number of packet arrivals per slot has been obtained in [1] as

$$E(x) = \sum_{i=0}^{\infty} M(1-\alpha^i + \alpha^i x)$$

Eqn. 5 was used in [1] to derive the mean and the variance of the system contents $s$. In this Letter, we shall show how the functional eqn. 5 can be used to arrive at a closed-form expression for the pgf $S(z)$ of $s$.

Pgf of the system contents: The pgf $S(z)$ of the system contents $s$ at the start of an arbitrary slot in the steady state is given by $P(1, z)$. To derive $S(z)$, we first select only those $(x, z)$-values for which the arguments of the $P$-functions on both sides of eqn. 5 are identical, i.e. for which $x = 1 - \alpha + \alpha x z$, or $x = (1-\alpha)(1-\alpha z)$. This choice leads to a linear equation for the function $P(1-\alpha + \alpha x z, z)$, which has the following solution:

$$P \left( \frac{1-\alpha}{1-\alpha^z}, z \right) = \frac{(z-1)(1-\rho)M \left( \frac{1-\alpha z}{1-\alpha^z} \right)}{z - M \left( \frac{1-\alpha}{1-\alpha^z} \right)}$$

Next, we define the functions $R_i(z)$ by means of the following recursive equations:

$$R_i(z) = 1 - \alpha + \alpha R_{i-1}(z) \quad i \geq 1$$

where $R_0(z) = 1$. By means of induction it is possible to show that

$$R_i(z) = \frac{1-\alpha}{1-\alpha z} + (\alpha z)^i \frac{(1-z)}{1-\alpha z} \quad i \geq 0$$

By successively applying the functional eqn. 5, taking into account the recursion formulas (eqn. 8) and the fact that $\lim_i R_i(z) = (1-\alpha)/(1-\alpha z)$, for $|z| < 1/\alpha$, we obtain

$$S(z) = P(1, z) = \frac{M(z)}{z} \left\{ \frac{P(R_i(z), z) + (z-1)(1-\rho)}{z} \right\}$$

Substitution of the result (eqn. 7) in the above equation finally yields

$$S(z) = \frac{(z-1)(1-\rho)\tilde{F}(z)}{z - M \left( \frac{1-\alpha z}{1-\alpha^z} \right)}$$

where the function $\tilde{F}(z)$ is defined as

$$\tilde{F}(z) = M \left( \frac{1-\alpha z}{1-\alpha^z} \right) \sum_{i=0}^{\infty} \frac{M(R_i(z), z)}{z}$$

![Fig. 1 Tail distribution of system contents prob[s > S] against S for \( \rho = 0.8 \) and various values of mean message length m](image)

Tail distribution of system contents: To derive the tail distribution of the system contents from the expression in eqn. 11 for $S(z)$, we use the approximation method presented in [2]. Specifically, we approximate the tail distribution of the system contents by the following geometric form:

$$\text{prob}[s > S] \approx \frac{\theta}{\alpha_0 - 1} \left( \frac{1}{\alpha_0} \right)^{S+1}$$

where $\alpha_0$ is the pole of $S(z)$ with the smallest modulus and $\theta$ is the residue of $S(z)$ in the point $z = \alpha_0$. From eqn. 11, it follows that $\alpha_0$ is a real root of the equation $z - M(1-\alpha)/(1-\alpha z) = 0$. Hence, $\alpha_0$ can easily be calculated numerically, for instance by means of the Newton-Raphson iteration method. The residue $\theta$ is obtained from eqn. 11 as

$$\theta = \frac{(z_0 - 1)(1-\rho)\tilde{F}(z_0)}{1 - M \left( \frac{1-\alpha z_0}{1-\alpha^z} \right) \left( \frac{1-\alpha}{1-\alpha z_0} \right)^2}$$

where $\tilde{F}(z_0)$ can be derived from eqn. 12 as

$$\tilde{F}(z_0) = z_0 \sum_{i=0}^{\infty} \frac{M(R_i(z_0), z_0)}{z_0}$$

Note that $M(R_i(z_0), z_0)$ goes to $M(1-\alpha)/(1-\alpha z_0)$, if $i$ goes to infinity. Since $M(1-\alpha)/(1-\alpha z_0) = \alpha_0$, it follows that the factors $M(R_i(z_0), z_0)$ of the infinite product in eqn. 15 go to one, as $i \to \infty$. 

1720 ELECTRONICS LETTERS 3rd September 1998 Vol. 34 No. 18
goes to infinity. Consequently, $\tilde{F}(z_0)$ can be calculated numerically as
\[
\tilde{F}(z_0) \approx z_0 \left( \prod_{\ell=1}^{J} \frac{M(R_\ell(z_0)z_0)}{z_0} \right) \quad J \text{ sufficiently large}
\] 
(16)

Example: Suppose that new messages are generated according to a Poisson distribution with a mean value of $n$ messages per slot. In this case, the pmf $M(z)$ is given by $M(z) = e^{-z}z^k/k!$. In Fig. 1, the tail distribution of the system contents probability $S_{\rho}(z_0)$ is shown as a function of $S_{\rho}$ for $\rho = 0.8$ and various values of the mean message length $m = 1/(1-\rho)$. The Figure illustrates that, for a given $\rho$, the performance deteriorates as $m$ increases. In other words, for a given mean arrival rate $\rho$, a smaller number of, hence longer, messages leads to higher buffer occupancies than a larger number of shorter messages.

Acknowledgment: The author wishes to thank the Fund for Scientific Research - Flanders (Belgium) (F.W.O.) for support of this research.

© IEE 1998
Electronics Letters Online No: 19981248
S. Witteveen (Stochastic Modeling and Analysis of Communication Systems (SMACS) Research Group, Laboratory for Communications Engineering, University of Ghent, Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium)

References

Distributed multicast routing for delay-sensitive applications

Yongjin Im and Yanghee Choi

A distributed multicast algorithm for constructing minimum cost multicast trees with delay constraints is proposed. The proposed algorithm can always construct a delay-constrained multicast tree, if one exists, and exhibits superior tree cost performance compared to existing algorithms.

Introduction: In data networks, in order to handle a large number of multicast sessions, it is important to minimise session resource requirements, while meeting their end-to-end delay requirements. Most of the previous delay-constrained multicast algorithms have been centralised ones [1]. Here, we are interested in a distributed minimum cost multicast algorithm that satisfies the end-to-end delay requirement. Multicast algorithms that perform cost optimisation are known to be NP-complete [2]. The proposed algorithm is therefore a heuristic one.

The simplest distributed algorithm is the shortest path tree (SPT). The multicast SPT consists of the least delay path from the source to all destinations in the multicast group. While SPT minimises the delay, it does not try to minimise the total cost of the multicast tree.

A network is represented by a directed graph $G = (V, E)$, where $V$ is a set of nodes and $E$ is a set of links. Weighting functions for cost and delay, respectively, exist on each link: $C(e): E \rightarrow R$, $D(e): E \rightarrow R$. A multicast group $G \subseteq V$ is a set of nodes participating in the same multicast session. The problem of finding a multicast tree satisfying the end-to-end delay bound can be formulated as follows. Given $G = (V, E)$, $C(e)$, $D(e)$, source node $s$, multicast group $G$ and a delay bound $\Delta$, minimise $\Sigma_{v \in G} C(v)$, where $T_S$, $G$ is a multicast tree rooted at $s$ and spanning all of the nodes in $G$ such that for each node $g$ in $G$, $\Sigma_{v \in G, \delta(v,g) \leq \Delta}$, and $P(s, g)$ is the unique path in $T(s, G)$ from $s$ to $g$.

Proposed algorithm: We call the proposed algorithm the DDCMT (distributed delay-constrained multicast tree). In this algorithm, every node $v \in V$ must maintain the following information during the route computation: a delay vector and a cost vector. The delay vector at node $v \in V$ consists of $|V|-1$ entries, one entry for each other node $w$ in the network. The entry for node $w \in V$ holds the following information: end-to-end delay of the least delay path $P_v(v, w)$ from $v$ to $w$; $D(P_v(v, w))$, cost of $P_v(v, w)$; $\Sigma_{P_v(v, w)}$, and next hop node on $P_v(v, w)$ and $\Pi_{P_v(v, w)}$. The entry for node $w \in V$ in the cost vector contains the following information: end-to-end delay of the least cost path $P_v(v, w)$ from $v$ to $w$; $D(P_v(v, w))$, cost of $P_v(v, w)$; $\Sigma_{P_v(v, w)}$, and next hop node on $P_v(v, w)$ and $\Pi_{P_v(v, w)}$.

The source node $s$ sends TEST_METRIC message to nodes already included in the tree. Upon receiving TEST_METRIC, each node $v$ verifies its cost vector and delay vector to see if there exist delay-constrained paths to the destination nodes not included in the tree. At first, node $v$ checks eqn. 1 for each destination node $w$ not included in the tree. If eqn. 1 is satisfied, $\text{COST}_{\text{SPT}}(v, w)$ is set to $\Sigma_{P_v(v, w)}$. Otherwise it checks eqn. 2 and if it is satisfied, $\text{COST}_{\text{SPT}}(v, w)$ is set to $\Sigma_{P_v(v, w)}$.

\[
\sum_{e \in P_v(v, w)} D(e) + D(P_v(v, w)) < \Delta \quad (1)
\]
\[
\sum_{e \in P_v(v, w)} D(e) + D(P_v(v, w)) < \Delta \quad (2)
\]

If both eqns. 1 and 2 are not satisfied, $\text{COST}_{\text{SPT}}(v, w)$ is set to $\infty$. After node $v$ finds $\text{COST}_{\text{SPT}}(v, w)$ for all destinations not included in the tree, it selects the one with minimum $\text{COST}_{\text{SPT}}(v, w)$, and the related information is carried back to the source node $s$ in a TEST_METRIC_ACK message. In the source node, TEST_METRIC_ACK messages are collected from the children nodes, and the one with the least $\text{COST}_{\text{SPT}}(v, w)$ is selected. The node that sent the message is notified by a PATH_FIND message. Upon receiving the PATH_FIND message, node $v$ checks eqn. 1 is satisfied. If it is satisfied, node $v$ sends PATH_SETUP message to $\Pi_{P_v(v, w)}$. Otherwise it checks eqn. 2 and if it is satisfied, node $v$ sends PATH_SETUP message to $\Pi_{P_v(v, w)}$. Upon receiving the PATH SETUP message, the next hop node executes the same procedure as was performed by $v$ when receiving the PATH_FIND message. The destination node $w$ replies by sending a PATH SETUP ACK message when it receives a PATH SETUP message. The nodes on the path to the added destination $w$ are also included in the multicast tree, and participate in the subsequent tree computation steps. The source node repeats the whole procedure until there are no destinations left.

![Fig 1 Tree cost relative to BSM4](image_url)

109 nodes. $\beta_{\text{SPT}} = 100$ Mbit/s, $\beta_{\text{DDCM}} = 120$ Mbit/s. $\Delta = 30$ ms

DDCM

SPT

Performance evaluation: In the simulation, a random graph generator based on Waxman’s generator [3] was used to create net-