An Optimisation Model for a Two-Node Router Network

Nalan Gülpınar, Peter Harrison, Berç Rustem
Department of Computing, Imperial College London
South Kensington Campus, London SW7 2AZ, UK.
{ng2,pgh,br}@doc.ic.ac.uk

Louis-Francois Pau
Rotterdam School of Management, Erasmus University
F 1-19, Po Box 1738, NL 2000 DR Rotterdam, Netherlands.
Ericsson, Network Core Products Division, Sweden.
lpau@fbk.eur.nl

Abstract

Architectural designs for routers and networks of routers to support mobile communication are analysed for their end-to-end performance using a simple Markov model. In view of the diverse design options, such models have many adjustable parameters and choosing the best set for a particular performance objective is a delicate and time-consuming task. We introduce an optimisation approach to automate this task, illustrated in a two-node, tandem network of routers with finite capacity and recovery buffers. We minimise the mean end-to-end delay subject to an upper limit on the rate of losses, which may be due to either full buffers or corrupted data. Losses at a full buffer are inferred by a time-out whereas corrupted data is detected immediately on receipt of a packet at a router, causing a N-ACK to be sent upstream. Recovery buffers hold successfully transmitted packets so that on receiving a N-ACK, the packet, if present, can be retransmitted, avoiding an expensive resend from source. Hence, a critical parameter that affects both loss rate and transmission time is the ratio of arrival-buffer size to recovery-buffer size. We develop a queueing model of this network and present graphs showing how end-to-end delay varies with certain parameter combinations. The tedious nature of trying to find the best parameter values in this way motivates our formal optimisation which yields optimal parameter values directly from the model specification using standard software.

1. Introduction

Research on routing and data communications has been going on for many years after the seminal work by Kleinrock with the ARPANET and CYCLADES projects but in those days no one imagined extending the routing capability to elements of the communication path that could be corrupted and/or subject to mobility.

The real research challenge today is to evaluate the possibilities of carrying out, at sufficiently high sustainable data rates, packet-based communication wirelessly, with mobility, in public networks. This in turn hinges on a second research challenge which is to support the packet routing directly inside mobile terminals. So called ad-hoc networks, and other types of networks where the terminals are also relay nodes, represent just a special case of these two challenges.

Very few papers have addressed the architectural imperfections, such as randomness in traffic and errors [7], in a direct coupling with router design [6]. This is why it is important to research both static and dynamic optimisation of routers and router networks, in order to allow for their reconfiguration and achievement of the highest possible, sustained, non-corrupted data rates in highly heterogeneous architectures.

The first steps in this direction have been taken in a number of specialised router architectures, called "mobile routers" [1], which have designs, architectures, protocol suites, handover processes and resilience quite different from the common routers built for best-effort local area network (LAN) environments and the best-effort backbones. They are placed inside the access networks or at gateways to the still circuit-switched transmission of packets over the air interfaces. Another evolution in the same direction has been taken by some wireless terminal platform designs which, thanks to dual IPv6/IPv4 stacks, in effect can be made to behave as routers with static routing tables and with mobility support.

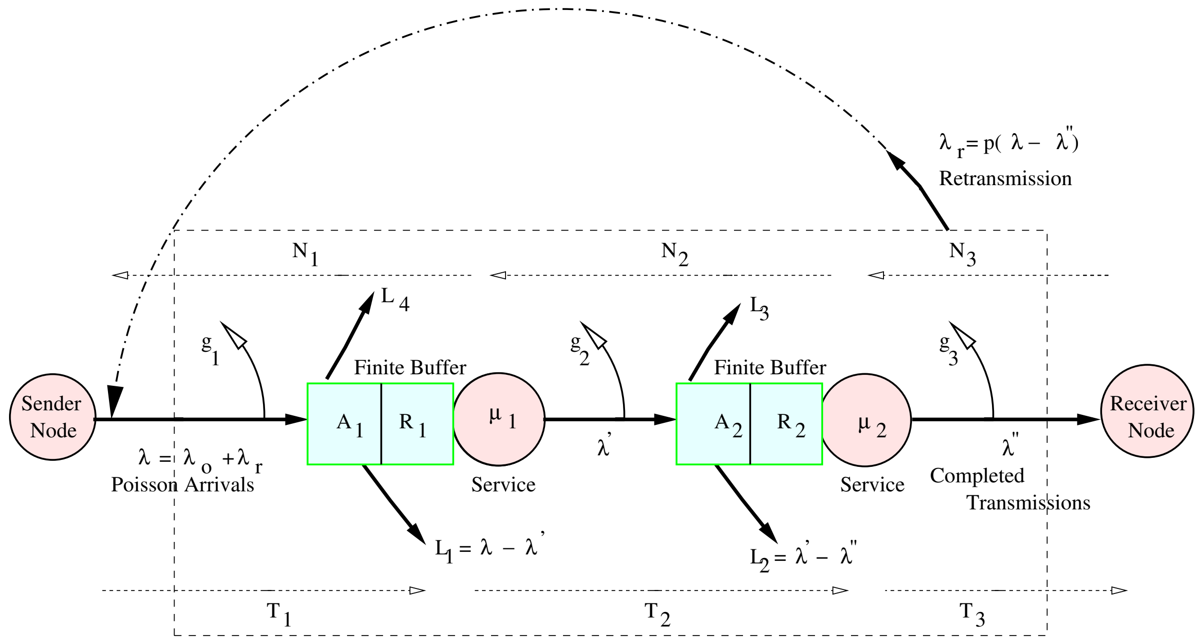

Figure 1. A two-node router network.

This paper is a first step towards an analytical, static optimisation-based design of routers for use in wireless communications. In view of the diverse design options possible, supporting models have many adjustable parameters and choosing the best set for a particular performance objective is a delicate and time-consuming task. Losses occur either because packets are rejected when they arrive at a full (arrival) buffer or else when corrupted packets are detected on receipt at a router. In the former case a full retransmission is initiated by the sender after a time-out, a significant increase in the end-to-end delay. In the latter case, the losses due to errors are signaled in the reverse direction in the network by N-ACKS and do not require a full retransmission if the clean packet remains in the *recovery* buffer of an upstream node. Packets successfully transmitted by a node are kept in its recovery buffer specifically for this purpose. Thus the choice of ratio of recovery-buffer size to arrival-buffer size in a fixed capacity node is critical to optimal performance: too small a recovery buffer and response times will be larger due to more retransmissions from source; too large and the arrival buffers will be too small and so response times will again be large due to network congestion arising from the retransmissions caused by losses.

The rest of the paper is organised as follows. In section 2, we describe a two-node network of routers with finite capacity and recovery buffers and introduce an optimisation problem which minimises the mean end-to-end delay. Numerical results are presented in section 3. The paper concludes with a summary of our contributions and discussion of research in progress.

2. Model of two-node router network

Consider a two-node, tandem network of internal routers with a sender (source) node and receiver (destination) node at the ends; see the model depicted in Figure 1. At each node there are arrival (for queueing) and recovery buffers. We assume that packets arrive at the server as a Poisson process with mean arrival rate λ_0 (number of arrivals per unit time) and wait if the node is busy and there is space in the buffer. In this initial model, the net arrival process is assumed Poisson. There are many situations where this approximation is reasonable; for example, a superposition of sparse, independent renewal processes is asymptotically Poisson and the net arriving traffic is likely to come from many sources. An enhancement of the model, which represents more typical Internet workloads with bursty, correlated traffic is discussed in section 4. However, the present assumption effectively supports our optimisation strategy as well as being of value in its own right.

The packets are served when they reach the front of the arrival queue. Service time, which includes the time to look up router tables, manage header information and dispatch the packet so that the next can be processed, is assumed to be an exponential random variable. Although another approximation, this assumption does at least reflect the commonly observed phenomenon that there are more shorter service times than long and has also been seen to be robust when estimating mean quantities in many queueing network applications. Extension to general service times is straightforward when the arrivals are Poisson and this too is discussed in section 4. The departure process from the queue is also approximated as Poisson; of course, assuming exponential service times, this assumption is exact if node-capacities are infinite and a good approximation if losses are rare.

In a Poisson stream, successive customers arrive after intervals which are independent and exponentially distributed [3]. Customers' service times are independent of each other and of the arrival process and the service rates (reciprocals of the means of their exponential distributions) at nodes 1 and 2 are μ_1 and μ_2, respectively. Any arrival packet is either transferred successfully or else is lost. Losses arise due to either a *full buffer* whereupon there is no acknowledgement (ACK) or *corrupted data on arrival at a buffer*. Losses due to a full buffer at each node, L_1, L_2, will cause an additional transmission delay by a timeout of duration much greater than the per-link transmission delay and typically a multiple of the estimated end-to-end delay. Losses due to corrupted data on arrival at buffer 1 and 2 are denoted by L_3, L_4 in Figure 1. We assume that garbage is not buffered but that a negative acknowledgement (N-ACK) is sent upstream instead. Losses might also occur due to incorrect routing or a wrong entry or late update in the routing table at a node. The effect on the sender (and receiver) is the same as a full-buffer loss, however, and we make no distinction in our model.

Packets are stored in the recovery buffer when any arrival (not overflow or garbage) to node 1 or node 2 has been successfully processed and sent on. If full, the oldest packet is deleted from the recovery buffer, which therefore operates in a round-robin, cyclic fashion.

Strictly speaking, a node has a third buffer, the output buffer, into which packets ready for downstream transmission are placed. A separate process manages the output buffer so that packets can be output concurrently with internal node processing (routing table look-up etc). Since we represent link transmission times separately from node service times (see the next section), delays associated with the output buffer can either be combined with link delays, when operation of the output buffer process is simultaneous with the node process, or else incorporated into the node's service time. We therefore do not consider the output buffer explicitly in this paper.

The definitions of our model's parameters are presented in Table 1. The symbol $*$ denotes data indexed by node $i = 1, 2$. Whilst many such queueing models have been considered before, for example [8, 9] which consider networks with finite buffers and augmented with additional non-queueing features, we believe our tailored investigation, involving the use of recovery buffers for example, is novel.

Table 1. **Table of model parameters.**

Fixed Parameters	
μ_*	service rate
λ_0	mean arrival rate
$C_* = A_* + R_*$	total buffer capacity
g_1, g_2, g_3	probability of corrupted data at links
T_1, T_2, T_3	transmission delays from sender to node 1, from node 1 to node 2 and from node 2 to receiver, respectively
N_1, N_2, N_3	N-ACK delays from node 1 to sender, from node 2 to node 1 and from receiver to node 2, respectively
Control Parameters	
p	probability of retransmission
L_r	maximum acceptable loss rate
Output Parameters	
A_*	arrival buffer size
R_*	recovery buffer size
L_*	loss rate
W_*	average response time
U_*	server utilisation
λ	total mean arrival rate
λ_r	mean rate of retransmissions
$\lambda^{'}$	throughput from node 1
$\lambda^{''}$	rate of successful transmissions
f	probability of a full buffer

2.1. Model structure and parameters

The queueing network model determines the mean end-to-end transmission time (also called response time) in terms of its pre-specified parameters. Optimal choices can be found by running the model with various choices of the variable parameter values (like ratio of arrival to recovery buffer sizes in a node with given total buffer capacity) and identifying minima in a painstaking process, some results of which we show in section 3.

The model's design, in relation to its main fixed parameters, control (i.e. user defined) parameters and output (performance) parameters, is described in the following subsections. However, note that as far as the model is concerned, the distinction between fixed and control parameters is immaterial: they are all just input parameters used to compute an objective function defined in terms of the output parameters that the model computes.

2.1.1 Fixed parameters

As already remarked, external arrivals from various user input streams have total rate λ_0. These merge with retransmissions, which have rate λ_r, the net rate at which packets are resent to the sender node due to losses of all types. The total rate of arrivals, including retransmissions, is therefore $\lambda = \lambda_0 + \lambda_r$.

The probabilities that a packet is lost because of corruption that leads to garbage data arriving at node 1, node 2 and the receiver are g_1, g_2 and g_3, respectively. Such garbage is detected on arrival at a node, before joining the queue, and causes a N-ACK to be sent upstream to the sender. We assume that acknowledgements are processed fast, at high priority and do not incur queueing delays. More precisely, N-ACKs (and ACKs) are generated fast by a process running on a node's arrival buffer and an ACK packet is sent back to the previous node's *output* buffer. Because of the high priority of ACKs and N-ACKs and the fact that they do not have to pass through arrival buffers, they suffer no queueing delays, although they may place a small overhead on the nodes. Consequently the simplified representation of N-ACK transmission in our model should not be unduly inaccurate.

N_1, N_2 and N_3 are the N-ACK delays from node 1 to the sender, from node 2 to the node 1, and from the receiver to node 2, respectively. Hence, referring to Table 1 and Figure 1, the N-ACK delay, i.e. the time for the N-ACK to arrive back at the sender, is $N_1, N_1 + N_2$ and $N_1 + N_2 + N_3$ when corruption is detected at node 1, node 2 and the receiver node, respectively.

Losses due to full arrival buffers are influenced by the total capacities C_i of nodes $i = 1, 2$. These capacities are partitioned into fixed size arrival and recovery buffers, as discussed in the next section.

Finally, T_1, T_2 and T_3 are the transmission delays for packets passing over the links between the sender and node 1, node 1 and node 2, and node 2 and the receiver node, respectively. Hence the transmission delay for successful packets, excluding the time spent in the nodes, is

$$T = T_1 + T_2 + T_3$$

2.1.2 Control parameters

Successful transmissions have throughput λ'' (defined in section 2.1.3) so that a packet is not lost on a given transmission attempt with probability λ''/λ. A packet that is rejected due to a full arrival buffer at either node will be delayed for a user-defined time-out period as follows. The sender recognises that a transmission has been successful when it receives a positive acknowledgement (i.e. an ACK), sent by the receiver on receipt of an uncorrupted packet, according to the transmission protocol. On transmitting a packet, the sender sets a timer and if the timer's value reaches the prespecified time-out value, the packet is deemed lost and a retransmission may be attempted. The time-out period is set to a multiple k of estimated mean transmission times, for some $k \geq 1$, set by the user.

The probability p of retransmission is also decided by the user, $0 \leq p \leq 1$. When $p = 1$, there is always a retransmission attempt and so there are no losses (assuming no catastrophic faults) and the number of retransmissions is unlimited. This causes extra load on the network which might result in congestion and hence significantly longer delays. In mobile networks, unlimited retransmission is not acceptable since it can block the network. However, calls are never abandoned and *circuit switched* routes are used as an alternative after a certain number of retries or certain total elapsed time. As far as the packet network is concerned, however, these packets are lost, corresponding to the case $p < 1$.

When $p = 0$, no retransmissions are allowed so losses are higher but congestion is lower. Hence, successful transmissions are, on average, faster.

Let f denote the probability of a time-out delay, caused by arrival at a full buffer. Now, the probability that a packet is successfully transmitted to the receiver is λ''/λ and so

$$g_1 + g_2 + g_3 + f = 1 - \frac{\lambda''}{\lambda}$$

For convenience, we also write $P = 1 - \lambda''/\lambda$. The loss probability f is obviously influenced by the size of the arrival buffers, which are constrained by the node capacities and the partitioning into arrival and recovery buffers. We denote the sizes of the arrival and recovery buffers at node $i = 1, 2$ by A_i and R_i, respectively, with the constraint that $A_i + R_i = C_i$. If R_i is set too small, transmission times will be large due to the need for more retransmissions from the sender. If it is too large, the arrival buffers will be too small and so transmission times will again be large due to losses and the resulting time-outs and retransmissions. It is an optimisation exercise to find the best ratio of recovery buffer to arrival buffer sizes.

Under our assumptions, the probability of finding a full buffer at node $i = 1, 2$ is equal to the probability of finding A_i customers in the corresponding M/M/1/A_i queue. Since we are assuming Poisson arrivals which encounter queue lengths with the equilibrium probability distribution (by the Random Observer Property), we estimate f by $(1-\rho_1)\rho_1^{A_1}/(1-\rho_1^{A_1+1}) + (1-\rho_2)\rho_2^{A_2}/(1-\rho_2^{A_2+1})$ where $\rho_1 = \lambda/\mu_1$ and $\rho_2 = \lambda'/\mu_2$. An approximation for the loss probability at a given node $i = 1, 2$, which is accurate at very low loss probabilities, is simply the probability of finding at least A_i customers in the corresponding M/M/1/∞ queue, i.e. $\rho_i^{A_i}$ and then $f = \rho_1^{A_1} + \rho_2^{A_2}$.

During transmission, losses should not exceed a specified maximum *loss rate* L_r, measured conventionally as a *fraction* of the external arrival rate. This implies

$$(1 - p)(\lambda - \lambda'') \leq \lambda_0 L_r$$

2.1.3 Output parameters

From the input parameters (fixed and control) the model computes the following performance measures as output:

- the utilisation of each node;
- the throughput of the network, i.e. the rate at which successful transmissions are received;
- the mean 'response time' of successful packets, i.e. the mean end-to-end delay from the instant that the packet was first sent to the instant of its successful arrival at the receiver;
- the rate of losses due to either corruption or full buffers.

The server utilisations at nodes 1 and 2, U_1 and U_2, are simply the ratios of their throughputs to service rates:

$$U_1 = \frac{\lambda'}{\mu_1}$$
$$U_2 = \frac{\lambda''}{\mu_2}$$

We assume Poisson arrivals at both queues, as discussed already, with raw arrival rates λ at node 1 and λ' (the throughput of node 1) at node 2. The throughputs of successful transmissions from nodes 1 and 2 are then obtained by direct analysis of the M/M/1 queue as

$$\lambda^{'} = \frac{\rho_1\left(1-\rho_1^{A_1}\right)\mu_1}{1-\rho_1^{A_1+1}}$$
$$\lambda^{''} = \frac{\rho_2\left(1-\rho_2^{A_2}\right)\mu_2}{1-\rho_2^{A_2+1}}$$

The expected response times at nodes $i = 1, 2$ are found as

$$W_i = \frac{1}{\mu_i(1-\rho_i)(1-\rho_i^{A_i})}(1-(A_i+1)\rho_i^{A_i} + A_i\rho_i^{A_i+1})$$

The total expected response time W, i.e. the combined time spent in the two nodes on a successful transmission attempt, is the sum of the expected response times at each node, $W = W_1 + W_2$. Thus, the mean transmission time MTT for a packet that is successful on its first attempt is $MTT = W + T$.

2.2. Total mean transmission time

Failed packets, due to either corrupted data or loss at a full arrival buffer, retry a number of times given by the retry-probability p. Because each retry is made independently of previous attempts, this number of attempts is a geometric random variable with parameter p. The overhead incurred by a failed transmission, i.e. the elapsed time between the start of an attempt that subsequently fails and the start of the next attempt, consists of:

- the time elapsed to the instant that the failure is detected—i.e. arrival at a node with garbage or at a node that is full;
- the time-out delay of $k(W + T)$ (k mean successful transmission times) for packets lost due to a full buffer;
- the time required for a N-ACK to be transmitted back to the sender.

We can express this overhead, L say, in terms of the probabilities of the different causes of failure as follows:

$$\begin{aligned} L &= k(W+T)f + g_1(N_1+T_1) + g_2q_1(T_2+N_2) \\ &+ g_2(1-q_1)(W_1+T_1+T_2+N_1+N_2) \\ &+ g_3(1-q_2){q'}_1(T_3+N_3+T_2+N_2+W_2) \\ &+ g_3(1-q_2)(1-{q'}_1)(W+T+N) \\ &+ g_3q_2(T_3+N_3) \end{aligned}$$

where q_1, q_2, q_1' are the probabilities of finding a clean copy of a corrupted packet in the upstream recovery buffer at node 1 (when the corrupted packet was detected at node 2), node 2 (when it was detected at the receiver) and node 1 (when it was detected at the receiver), respectively.

For an eventually successful transmission, the mean transmission time conditional on the number of attempts $m = 1, 2, \cdots$, together with the corresponding probability of that number of attempts, is shown in Table 2.

The total mean unconditional transmission time is now calculated as

$$\begin{aligned} MTT &= (W+T)\frac{\lambda''}{\lambda}\sum_{m=0}^{\infty}\left[p\left(1-\frac{\lambda''}{\lambda}\right)\right]^m \\ &\quad +pL\frac{\lambda''}{\lambda}\sum_{m=1}^{\infty} m\left[p\left(1-\frac{\lambda''}{\lambda}\right)\right]^{m-1} \\ &= \frac{\lambda''(W+T)}{\lambda(1-p)+p\lambda''} + pL\frac{\lambda''}{\lambda}\frac{1}{\left[1-p\left(1-\frac{\lambda''}{\lambda}\right)\right]^2} \\ &= \frac{\lambda''}{\lambda(1-p)+p\lambda''}\left[(W+T) + \frac{p\lambda L}{\lambda(1-p)+p\lambda''}\right] \end{aligned}$$

2.3. Probabilities of recovery

We now estimate the probabilities q_1, q_1', q_2 of a clean copy of a corrupted packet remaining in the recovery buffer at node 1 or 2 when a N-ACK arrives from downstream.

Table 2. **Mean transmission time conditional on number of attempts and corresponding probability.**

Attempts	Conditional Mean Response Time	Probability
1	$W+T$	$\frac{\lambda''}{\lambda}$
2	$W+T+k(W+T)\frac{f}{P}+(N_1+T_1)\frac{g_1}{P}$ $[(1-q_1)(W_1+T_1+T_2+N_1+N_2)+q_1(T_2+N_2)]\frac{g_2}{P}$ $\left[(1-q_2)(1-q_1^{'})(W+T+N)+q_2(T_3+N_3)\right]\frac{g_3}{P}+$ $\left[q_1^{'}(1-q_2)(T_3+N_3+T_2+N_2+W_2)\right]\frac{g_3}{P}$	$\frac{\lambda''}{\lambda}\left(1-\frac{\lambda''}{\lambda}\right)p$
$\cdots$	$\cdots$	$\cdots$
m	$W+T+(m-1)k(W+T)\frac{f}{P}+(m-1)(N_1+T_1)\frac{g_1}{P}$ $(m-1)\left[(1-q_1)(W_1+T_1+T_2+N_1+N_2)+q_1(T_2+N_2)\right]\frac{g_2}{P}$ $(m-1)\left[(1-q_2)(1-q_1^{'})(W+T+N)+q_2(T_3+N_3)\right]\frac{g_3}{P}+$ $(m-1)\left[q_1^{'}(1-q_2)(T_3+N_3+T_2+N_2+W_2)\right]\frac{g_3}{P}$	$\frac{\lambda''}{\lambda}\left(1-\frac{\lambda''}{\lambda}\right)^{m-1}p^{m-1}$
$\cdots$	$\cdots$	$\cdots$

These probabilities will depend on the size of the recovery buffer—the bigger the buffer, the higher the recovery probability—and the time elapsed between a corrupted packet being detected and the resulting N-ACK arriving at the node holding the clean packet—the longer this delay the more likely the clean packet will have been replaced by a packet transmitted later.

Consider first departures from node 1, which take time T_2 to reach node 2 where they arrive corrupted. The N-ACK to node 1 takes time N_2 and so q_1 is defined as the probability that there are less than R_1 departures from node 1 in time T_2+N_2. Thus,

$$q_1 = \sum_{k=0}^{R_1-1} a_{1k}(T_2+N_2)$$

where

$$a_{1k}(t) = \Pr(k \text{ departures from node 1 in time } t) = \frac{[\lambda' t]^k}{k!}e^{-\lambda' t}$$

Since T_2+N_2 is constant, we compute $a_{1k}(T_2+N_2)$ as $\lambda'^k(T_2+N_2)^k e^{-\lambda'(T_2+N_2)}/k!$ for $k=0,1,\cdots,R_1-1$.

Similarly, q_2 is defined as the probability of less than R_2 departures from node 2 in time T_3+N_3:

$$q_2 = \sum_{k=0}^{R_2-1} a_{2k}(T_3+N_3)$$

where $a_{2k}(T_3+N_3)=\lambda''^k(T_3+N_3)^k e^{-\lambda''(T_3+N_3)}/k!$ for $k=0,1,\cdots,R_2-1$.

To calculate q_1', we need the probability of k arrivals in a time which is a sum of random variables which are not all constant. Indeed, for networks with more than two nodes, the number of hops from a node detecting garbage to an upstream node holding a clean copy could be several. We therefore prove the following result.

Proposition 1 *Let the independent random variables $V_1,\cdots,V_n$ have probability density functions $v_1(t),\cdots,v_n(t)$ with respective Laplace transforms $v_1^*(s),\cdots,v_n^*(s)$. Then the probability that there are k arrivals of an independent Poisson process with rate γ in time $V_1+\cdots+V_n$ is*

$$a_k(V_1,\cdots,V_n)=\frac{(-\gamma)^k}{k!}\frac{d^k}{d\gamma^k}\left[v_1^*(\gamma)v_2^*(\gamma)\cdots v_n^*(\gamma)\right]$$

Proof

Let the random variable $V=V_1+\cdots+V_n$ have probability density function $v(t)=v_1(t)*\cdots*v_n(t)$, where $*$ denotes convolution, with Laplace transform

$$v^*(s)=v_1^*(s)\cdots v_n^*(s)$$

$$\begin{aligned} a_k(V_1,\cdots,V_n) &= \int_0^\infty e^{-\gamma t}\frac{[\gamma t]^k}{k!}v(t)dt \\ &= \frac{(-\gamma)^k}{k!}\int_0^\infty \left[\frac{d^k}{d\gamma^k}e^{-\gamma t}\right]v(t)dt \\ &= \frac{(-\gamma)^k}{k!}\frac{d^k}{d\gamma^k}\left[\int_0^\infty e^{-\gamma t}v(t)dt\right] \\ &= \frac{(-\gamma)^k}{k!}\frac{d^k}{d\gamma^k}\left[v_1^*(\gamma)\cdots v_n^*(\gamma)\right] \end{aligned}$$

♠

The density function of the constant random variable $X = c$ is the Dirac delta function $x(t) = \delta(t - c)$, which has Laplace transform $x^*(s) = e^{-sc}$. Thus, for constant random variables $X_i = c_i, 1 \le i \le n$, we obtain

$$a_k(X_1,\cdots,X_n) = \frac{(-\gamma)^k}{k!}\frac{d^k}{d\gamma^k}e^{-\gamma c} = \frac{(\gamma c)^k}{k!}e^{-\gamma c}$$

where $c = c_1 + c_2 + \cdots + c_n$. This yields our results for $a_{1k}(t)$ and $a_{2k}(t)$.

The following results now follow immediately.

Corollary 1 *Let $X_1,\cdots,X_n$ be constants and V be a random variable. Then*

$$a_k(X_1,\cdots,X_n,V) = a_k(X_1+\cdots+X_n,V)$$

Corollary 2 *Let V_α, V_β be exponential random variables with distinct parameters α, β, respectively and let Z be any random variable. Then*

$$a_k(Z,V_\alpha,V_\beta) = \frac{\beta}{\beta-\alpha}a_k(Z,V_\alpha) - \frac{\alpha}{\beta-\alpha}a_k(Z,V_\beta)$$

Proof

$$\begin{aligned} v_\alpha^*(\gamma)v_\beta^*(\gamma) &= \frac{\alpha\beta}{(\gamma+\alpha)(\gamma+\beta)} \\ &= \frac{\beta}{\beta-\alpha}v_\alpha^*(\gamma) - \frac{\alpha}{\beta-\alpha}v_\beta^*(\gamma) \end{aligned}$$

and so the result follows from proposition 1. ♠

In fact, we can expand a product of Laplace transforms of exponential densities in partial fractions similarly, to obtain the following more general result.

Corollary 3 *Let $V_{\alpha_1},\cdots,V_{\alpha_n}$ be exponential random variables with distinct parameters $\alpha_1,\cdots,\alpha_n$ respectively and let Z be any random variable. Then*

$$a_k(Z,V_{\alpha_1},\cdots,V_{\alpha_n}) = \sum_{i=1}^{n} c_i a_k(Z,V_{\alpha_i})$$

for certain constants c_i.

We can now easily compute q'_1, which is the probability of less than R_1 departures from node 1 in time $W_2 + N_3 + T_3 + N_2 + T_2$, i.e.

$$q'_1 = \sum_{k=0}^{R_1-1} a_k(N_3+T_3+N_2+T_2,W_2)$$

where W_2 is approximated by an exponential random variable with rate $\alpha = \mu_2 - \lambda''$ and $N_3 + T_3 + N_2 + T_2$ is a constant, call it v. Here, the $M/M/1$ response time approximation for W_2 is reasonable because overflows are very rare in practice and the impact of the finite buffer is negligible. Hence we estimate:

$$\begin{aligned} a_k(v,W_2) &= \alpha\frac{(-\lambda')^k}{k!}\frac{d^k}{d\lambda'^k}\left[\frac{e^{-\lambda' v}}{\alpha+\lambda'}\right] \\ &= \frac{\alpha\lambda'^k e^{-\lambda' v}}{(\alpha+\lambda')^{k+1}}\sum_{n=0}^{k}\frac{(\alpha+\lambda')^n v^n}{n!} \end{aligned}$$

2.4. The optimisation problem

The optimisation model minimises the mean end-to-end delay subject to an upper limit on the rate of losses. For the case of non-zero recovery buffer size, the optimisation problem is stated as follows:

$$NZRB: \quad \min_{A_1,A_2,R_1,R_2,\lambda,\lambda_r,\lambda'',f} MTT$$

subject to

$$\begin{aligned} \lambda &= \lambda_0 + \lambda_r \\ \rho_1 &= \frac{\lambda}{\mu_1} \\ \rho_2 &= \frac{\rho_1(1-\rho_1^{A_1})\mu_1}{(1-\rho_1^{A_1+1})\mu_2} \\ \lambda'' &= \frac{\rho_2(1-\rho_2^{A_2})\mu_2}{1-\rho_2^{A_2+1}} \\ \lambda_r &= p(\lambda-\lambda'') \\ \lambda_0 L_r &\ge (1-p)(\lambda-\lambda'') \\ 1-\frac{\lambda''}{\lambda} &= g_1+g_2+g_3+f \\ C_1 &= A_1+R_1 \\ C_2 &= A_2+R_2 \\ \lambda,\lambda_r,\lambda'',R_1,R_2 &\ge 0 \\ 0 \le A_1 &\le C_1 \\ 0 \le A_2 &\le C_2 \end{aligned}$$

where $\lambda_0, g_1, g_2, g_3, L_r, p, C_1$ and C_2 are pre-determined constants.

Such problems are routine for convex objective functions when the parameters are real numbers and often well approximated when some are integers—by allowing them to be real and choosing the nearest integers as candidate optima. We present our numerical examples in section 3.

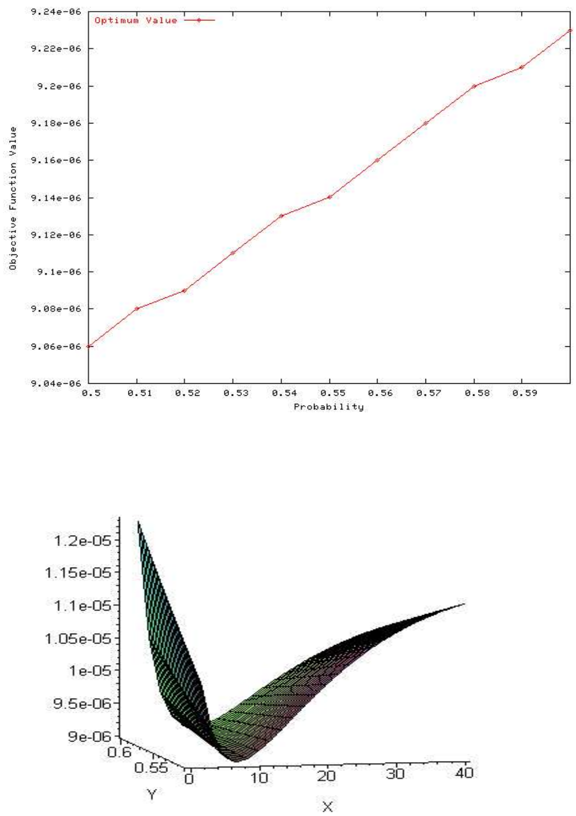

Figure 2. Mean transmission time plotted at various retry probabilities with and without optimisation.

3. Numerical results

The optimisation model described in the previous section was implemented and integrated with a software package called *basqueu*. *Basqueu* has been written in C++ and uses the nonlinear solver E04UCF, Nag Library [5], to optimise the transmission time. In our computational experiments, the parameters, which are input to the optimisation model, are as follows: the service rates at nodes 1 and 2 are $\mu_1 = \mu_2 = 1.0E + 06$, the probabilities of garbage data at the nodes are $g_1 = g_2 = 0.0002$ and $g_3 = 0.001$, the maximum loss rate is 1% of the arrival rate, the acknowledgement delays are $N_1 = N_2 = N_3 = \frac{1.0}{10.0 \times \mu_1}$, the link transmission delays are $T_1 = T_2 = T_3 = \frac{1.0}{10.0 \times \mu_1}$. All computational experiments were carried out on a 500 MHz Pentium III, running Linux with 256Meg of RAM.

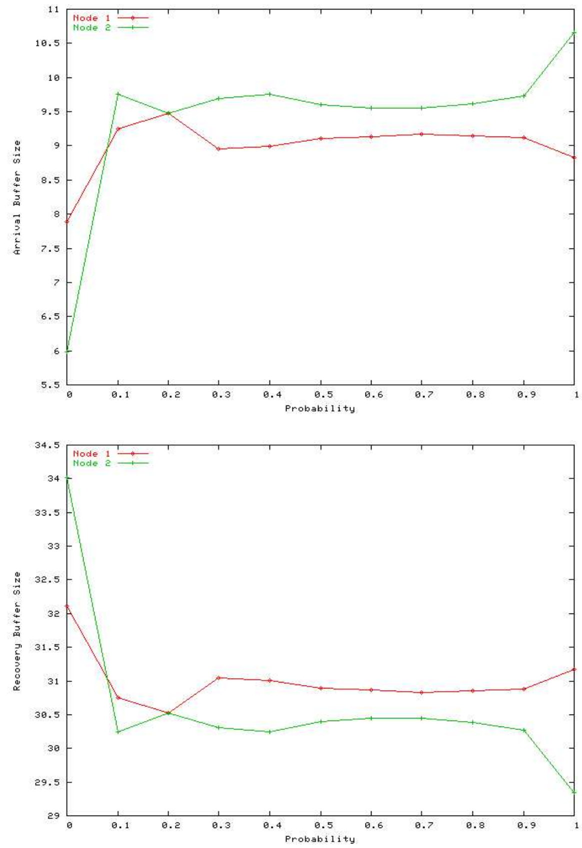

Figure 3. Optimal arrival and recovery buffers versus retry probabilities.

Partly in order to motivate the optimisation model, we considered the mean transmission time function over the range of retransmission probabilities, $0.5 \leq p \leq 0.6$ in Figure 2. The arrival buffer size at node 2 was fixed at 10 and the arrival rate was chosen to be 90% of the service rate of each node. The graph on the bottom in Figure 2 is a plot of the function MTT (the Z-axis) versus the user-defined probability p (the Y-axis) and arrival buffer size at node 1 A_1, (the X-axis). Although not realistic, this network being overloaded with excessively high accepted loss rate, these parameters were chosen to reveal interesting performance characteristics, in particular optimum operating points, to illustrate our approach.

The graph on the top in Figure 2 was obtained by *basqueu* software. The optimum buffer size at node 1 was obtained as 7.321 and the optimum objective value varies between $9.06E - 06$ and $9.23E - 06$. This clearly verifies that the minimum mean transmission time is obtained at the minimum arrival and recovery buffer size.

We next carried out experiments to illustrate the performance of the optimisation model and presented our results in terms of different parameters, varying the value of the user-defined retransmission probabilities, $0 \leq p \leq 1$. The optimum values of the arrival and recovery buffer sizes at nodes 1 and 2 can be seen in Figure 3.

When retransmission is not permitted, i.e. $p = 0$, the arrival buffer capacities at nodes 1 and 2 are kept at the minimum values (8 and 6), whereas the recovery buffer capacities are at their maxima (32 and 30). When $p > 0$, since the number of arrivals is increased due to retransmissions, the arrival buffer sizes at nodes 1 and 2 are also increased, the recovery buffer sizes being correspondingly decreased so that the constraints on the total node capacities are satisfied.

Figure 4 shows the packet arrival rate, throughput, retransmission rate and loss rate (i.e. $(1-p)(\lambda - \lambda'')$) versus p. This figure shows how, as p approaches 1, these rates all increase, as expected.

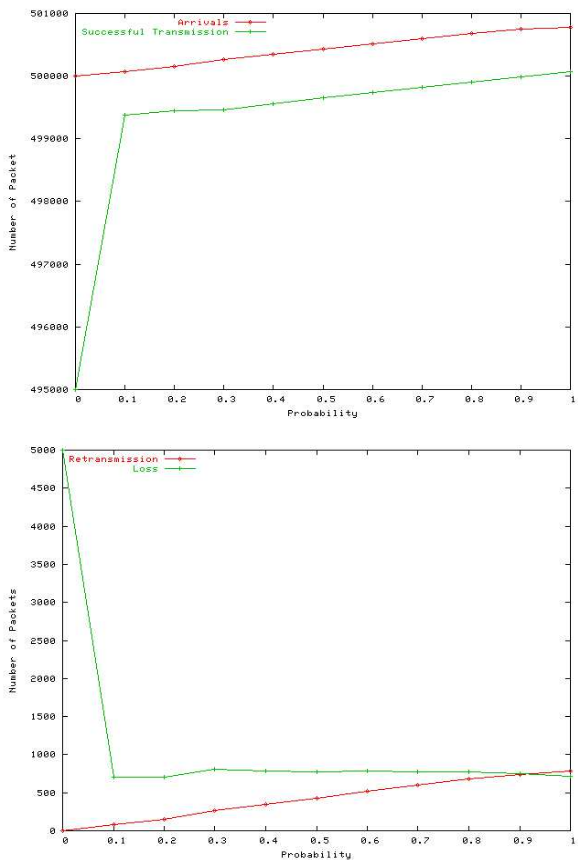

Figure 4. The number of packets versus retry probabilities.

The graph on the top in Figure 5 shows the optimum total mean transmission time (MTT) and the value of $W = W_1 + W_2$. The graph on the bottom in Figure 5 shows the probability of a full-buffer being encountered for $0 \leq p \leq 1$. Notice that the total transmission time as well as the value of W is also increased when the total arrival rate is increased due to a higher probability of retransmission.

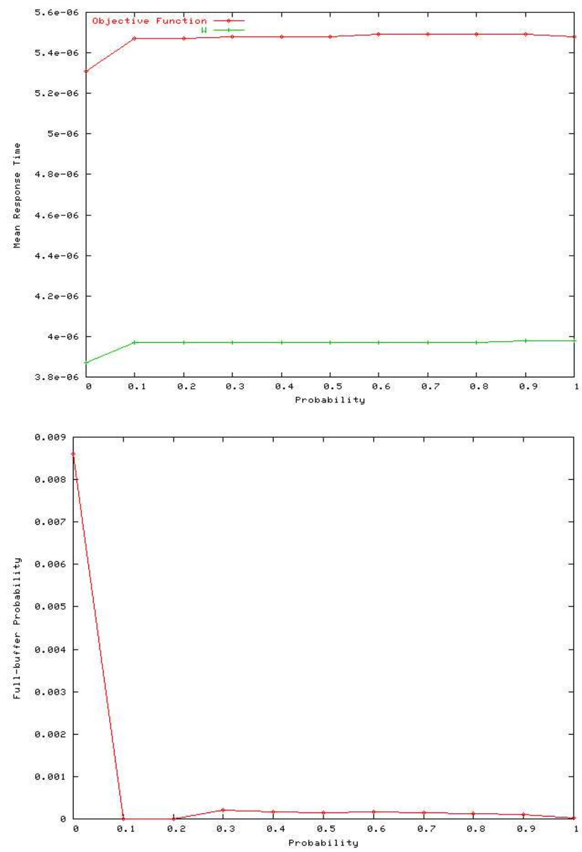

Figure 5. Optimum mean transmission time and probability of full buffer.

4. Conclusion

We have introduced a model of a two-node router network in order to demonstrate the optimisation of end-to-end performance, using a Markov model. The numerical results showed how end-to-end delay varies with various parameter combinations, from which optimal operating points could be deduced with considerable, tedious effort. This motivated our formal optimisation approach which finds these points automatically. A two-node network was chosen for its simplicity in illustrating our approach, but it is clear that arbitrary sized networks could equally well be accommodated at greater computational effort.

In the present model the traffic characteristics are not yet those of the bursty, correlated and lossy packet radio networks, but Poisson. Our forthcoming research will address time-dependent optimisation by stochastic control [7] and, even more interestingly, worst-case design approaches with design tradeoffs amongst minimal transmission times, maximum sustainable data rates and lowest possible packet loss. In addition, we will generalise the present model to non-exponential service times, using the M/G/1 building block, and to Markov modulated Poisson (MMPP) arrivals, which can better represent the bursty, correlated nature of modern telecommunications traffic; or even further, to a MAP (Markovian Arrival Process). Whilst a Poisson point process is frequently realistic for node arrival streams since many users are often connected, none of which dominate, the arrival stream from a single user is often not well approximated by a renewal process since inter-arrival times tend to be correlated. Moreover, the streams are often bursty. To handle these problems we will consider modulated *batched* arrival processes in which, at each arrival instant, several packets arrive, not just one, with specified probability distribution. Combined with modulation of rates, queues with such arrivals have been solved at equilibrium, for example in [2].

Finally, we will also consider the effect of network transmission protocols such as TCP which may have a dominant effect on our models' parameters and structure. If the time varying effect on arrival rate turns out to be highly significant, we may consider fluid models rather than discrete-state queues to represent the nodes in the same optimisation framework [4].

Acknowledgements

This research was supported by EPSRC grant GR/S27849/01.

References

[1] J. Borje, H.A. Lund, A. Wirkestrand, "Real-time Routers for Wireless Networks", *Ericsson Review Journal*, Vol 76, 4, 1999, pp. 190–197.

[2] R. Chakka, P.G. Harrison, "A Markov Modulated Multi-server Queue with Negative Customers - The MM CPP/GE/c/L G-Queue", *Acta Informatica*, Vol. 37, 2001, pp.881-919.

[3] Harrison, P.G, N.M. Patel, *Performance Modeling of Communication Networks and Computer Architectures*, Addison-Wesley Publishing, 1992.

[4] D. Mitra, "Stochastic Theory of a Fluid Model of Producers and Consumers Coupled by a Buffer", *Advances in Applied Probability*, Vol. 20, 1988, pp. 646-676.

[5] NAG Library, `http://www.nag.co.uk`, The Numerical Algorithms Group, Oxford.

[6] L-F Pau, "Mobile Network Evolution Requirements on Optimisation of Routing Function", Seminar, Department of Computing, Imperial College, London, 2004.

[7] L-F Pau, B. Wafae, and A. Bousquet, "Fault Detection Capability Implications for Flow Control in Data Communication Networks", *Proc. 1979 IEEE Int. Conf. AUTOTESTCON'79*, Minneapolis, 19-21 Sept. 1979, pp. 42–56.

[8] K.C. So, K.E. Chin, "Performance Bounds on Multiserver Exponential Tandem Queues with Finite Buffers", *European Journal of Operations Research*, Vol 63, 3, 1992, pp. 463–477.

[9] C.S. Sunga, M.H. Parkb, "Performance Analysis of a Queuing System with Facilities Requiring Post-processing", *Computers and Operations Research*, Vol 19, 8, 1992, pp. 767–781.