# In-call Probing and End-to-End Congestion Control: Theory and Performance

Alan Bain
Statistical Laboratory
Cambridge

Peter Key
Microsoft Research Limited
Cambridge

**Abstract**

This paper looks at adaptive applications that can switch between a small number of different levels, with switching decisions made solely by the originating end-system. Typical of such applications are real time streaming protocols which can use different coding rates. The end-systems probe the network with their own traffic to determine congestion, and decide at what rate to enter according to the fate of their "probe" packets. During the lifetime of a connection, the procedure is repeated to reassess and possibly readjust the rate. We derive analytic models, based on diffusion limits under a natural scaling, to quantify the benefits of in-call probing. We then use simulation to compare the results in a number of scenarios, and show that this simple theory is remarkably accurate in predicting large-scale behaviour. The results also show that a small amount of in-call probing produces significant benefits to the system.

## 1 Introduction

Study of current rate control mechanisms for the Internet has mainly concentrated on applications and protocols that are almost infinitely adaptable, in the sense that the applications can adapt their rate over a wide range, and can tolerate an arbitrarily small rate – TCP can be interpreted in this light. Such protocols are well-suited to so-called elastic traffic, and are appropriate for applications such as file transfer or web browsing, which are relatively insensitive to delay.

In contrast, real-time streaming protocols may require a minimal level of bandwidth to function properly, and may only be able to transmit packets at one of a small number of discrete rates. Specific examples are certain voice over IP (VOIP) or conferencing applications where the coding rate may be set at call initiation. In the Internet, UDP based protocols such as RTP have the potential to allow sources to alter their sending rates using periodic feedback, where again the rate can take values from a limited set.

The traditional approach to giving any sort of guarantee to such streaming protocols is to use signalling with some form of bandwidth reservation, such as that used in the Intserv RSVP proposal [4]. In such an approach, it is invariably the network (operator) or the resource which plays a key role in deciding whether or not the connections should be admitted.

Recent research [11, 17, 2, 7, 5, 12] has taken a different approach, seeking to involve a gateway, edge-device or end-system in the decision process. In this sort of *distributed admission control* [5, 12], the end-system probes the network with a number of packets and decides to enter or not depending on the fate of these probe packets (where the network drops or marks packets to signal congestion). Some of the advantages of this approach, such as scalability benefits, are detailed in [5].

Work to date has concentrated on distributed admission control for non-adaptive applications. The contribution of this paper is to study such distributed admission control for applications that are adaptive in the piecewise-constant sense discussed above – that is, they can send packets into the network at one of a small number of levels or rates. We assume that applications or end-systems probe the network by sending in a small number of probe packets. These packets are marked, for example using the recent ECN proposal [15], at resources (routers) if the resources are nearing congestion, and the probe packets are fed back to the source. The source then decides whether to enter, and what rate to chose, according to the fraction of probing packets that have been marked. In addition, whilst in progress, the application can occasionally reprobe the network and alter its rate.

In related work, Bolot et al. [3] look at rate adaptation for video, using loss estimation to adjust rates; and Daguiklas et al [6] also looked at video over ATM. However, both of these papers assume very frequent in-call adaptation, on a millisecond timescale, whereas we have in mind infrequent adjustment, on a time scale of tens of seconds. Indeed, theory and simulation suggests that little is gained by frequent adaptation.

The outline of the paper is as follows: In Section 2, we describe our general framework, and consider some of the architectural issues connected with implementing feedback and reaction of the type proposed. We introduce a specific motivational example which will be used throughout the paper, in which an application can send at one of two levels, and can switch between them during the call; and the network uses a specific packet-level marking strategy. Section 3 describes the analytic models which allow the benefits of in-call probing to be assessed. Section 4 presents results of simulations of various scenarios, such as for a star network, and compares these results with the analytic predictions, examining the sensitivity of the results to the assumptions.

## 2 Framework and Architecture

In this section we discuss a specific model of adaptation and end-point admission control, and consider some of the associated architectural issues. We are primarily concerned with end-system or user behaviour where packet marking is used as the primary means of conveying information about the network. Packet loss could be used as an alternative, though less informative, channel of information. Later in this section we describe the corresponding packet marking function required in the routers.

## 2.1 Connection level adaptation

We are concerned with relatively long-lived flows in the Internet that can send packets into the network at one of a small number of rates, the number of permissible rates being at least two. Connections may, whilst in progress, vary their rates among those in the set of permissible rates, where such changes are relatively infrequent – perhaps occurring on a timescale of seconds rather than milliseconds. The packet transmission behaviour at a particular rate level may generate bursty or Variable Bit Rate (VBR) traffic. A specific motivational example is a VOIP protocol where the coding rate may be set to one of a number of different rates (the two of G723.1 [10] for example), and where, if silence-suppression is used, the packet transmission is bursty. More generally, our results can be applied to any transcoding adaptation of a source. There are many ways in which the desired adaption could be implemented at the source; for example [3] discuss ways in which video block-coding schemes can react to congestion information.

For simplicity, we concentrate on the simplest case, where there are only two rates (high and low), a single resource, and where the resource load is generated by this class of calls alone. This is an appropriate model if such applications are segregated for example in a separate DiffServ class.

When a call (flow) starts, we assume that the end system sends a small number of probe packets into the network, and receives feedback in the form of those packets being marked (or dropped). This information is then used to determine whether to enter at the high or low rate, where for the moment we assume a call is always accepted. In addition, whilst in progress, the call re-probes the network periodically, and is able to switch between rates. We shall assume that there is a sufficiently great timescale difference between the packet level and the call level that individual acceptance/switching decisions may be considered as independent. Then at the call timescale, the probing and marking behaviour may be taken as having defined implicitly an acceptance probability $a(\rho)$ for a connection, where $\rho$ is the load on the network. The call enters at the high rate with probability $a(\rho)$, and at the low rate with probability $1 - a(\rho)$. Hence $1 - a(\cdot)$ a measure of the congestion in the network. When in-call probing occurs, $a(\rho)$ is also the probability for switching from a low rate to a high rate. All the probing packets are assumed to be part of the data-stream of the connection: in-call probing can use the data packets of the connection itself.

Current ECN proposals [15] only describe the reaction of TCP to marks, and only TCP receivers return marks to the source. We require the streaming source applications to have access to the state of the marked packets: this can either be done at the application level, such as by having the receiver generate return UDP packets, or by using a control channel (corresponding to RTCP for RTP [16]). In contrast to [5], we do not use probing to estimate the precise level of congestion, but rather use it as a guide. Connections are continually arriving and making decisions, hence the system has a self-regulating property which keeps the overall system behaving well.

We assume that in-call probing is a relatively infrequent process, occurring a few times per call. In the context of an IP telephony call lasting say 200 seconds, this corresponds to probing on the order of tens of seconds. We shall see simulation results which show that we get most of the benefit by probing just once or twice during the call.

## 2.2 Packet-level Models and Marking

The function $a(\cdot)$ determines the level to enter at. The function $a(\cdot)$ is determined by what happens at a timescale related to the packet level. We assume standard FIFO scheduling at the routers, which additionally marks packets according to some policy. The framework which we use is completely general, and can allow arbitrary marking policies. For example, we may mark packets when some threshold in a buffer is exceeded, or use enhancements based on active queue management such as RED [9]. In our examples and simulations we use a virtual queue (VQ) marking scheme [11, 12], which is able to provide early warning of problems.

VQ marking can be implemented with a counter which behaves as a virtual queue, running at a reduced service rate of say 90% of the real service rate. Packets arriving at the real queue are notionally also put into the virtual queue, and marked when the buffer in the virtual queue exceeds a threshold. Note that no real scheduling of packets occurs in the virtual queue.

Packet marks are fed back to the application. For example, we may assume that the Congestion Experienced (CE) bit defined by the ECN proposal [15] is set at the IP level in routers, and that the state of the probing packets is reflected to the sender. This may be implemented at the application level, or by using a control channel as mentioned above. Note that we do not assume any separate channel or scheduling for the probing packets.

The acceptance function $a(\cdot)$ is determined by the marking function and the user or end-point policy. A particularly simple user policy is the following: send $n$ packets into the network, enter at the high rate if no packets are marked (or lost), and otherwise enter at the low rate. In-call probing then uses the same policy, for $n$ packets of the data stream.

## 2.3 Example

Now consider a specific example, with a particular marking scheme, which we will analyse and in a later section. Suppose that calls arrive as a Poisson process of rate $\nu$, and when admitted last for an exponentially distributed holding time with mean one (a convenient renormalisation – we can rescale to general mean holding times). At the start of the call, at time $t$, the call probes the network and enters at the high rate, $r_H$ with probability $a(X_t^{(1)}, X_t^{(2)})$ and enters at the low rate, $r_L$ with probability $1-a(X_t^{(1)}, X_t^{(2)})$. Here $X_t^{(1)}$ is the number of high rate calls, and $X_t^{(2)}$ is the number of low rate calls in progress at time $t$. For the moment, we assume that connections are always accepted, and so calls enter at the high level at rate $\nu a(X_t^{(1)}, X_t^{(2)})$. Since we have rescaled the mean holding time to one, high-rate calls depart at rate $X_t^{(1)}$, and low rate calls depart at rate $X_t^{(2)}$.

While the connection is in progress, probing occurs as a random (Poisson) process of rate $\lambda$, and with probability $a(X_t^{(1)}, X_t^{(2)})$ a call can switch from a low rate to a high rate; this occurs at rate $\lambda X_t^{(2)} a(X_t^{(1)}, X_t^{(2)})$ since there are $X_t^{(2)}$ calls in progress at the low rate. The corresponding rate for switching between high rate and low rate is $\lambda X_t^{(1)} \left(1 - a(X_t^{(1)}, X_t^{(2)})\right)$

Suppose that we look at the network on such a large scale that we do not see the random effects of the individual calls. Then, informally, the state of the network may be represented by continuous functions $x_1(t)$ and $x_2(t)$ instead of the discrete stochastic variables $X_t^{(1)}$ and $X_t^{(2)}$. Combining in-call adaptation with the arrival and departure rates, we see that the rates satisfy,

$$\begin{aligned}\frac{\mathrm{d}x_1}{\mathrm{d}t} &= \nu a(x_1(t), x_2(t)) - x_1(t) + \lambda a(x_1(t), x_2(t))x_2(t) - \lambda(1 - a(x_1(t), x_2(t))x_1(t), \\ \frac{\mathrm{d}x_2}{\mathrm{d}t} &= \nu\left\{1 - a(x_1(t), x_2(t))\right\} - x_2(t) - \lambda a(x_1(t), x_2(t))x_2(t) + \lambda\left\{1 - a(x_1(t), x_2(t))\right\} x_1(t).\end{aligned} \tag{1}$$

This is made precise in Section 3, where we show that these equations arise naturally under a specified limiting regime.

For simplicity, assume that the high and low rates correspond to (mean) packet generation rates, so that the load upon a resource with $x_1$ high, and $x_2$ low rate calls is $y = x_1 r_H + x_2 r_L$. Suppose further that the real queue can serve at rate $C$ and $\kappa < 1$ is the reduction rate for the virtual queue. Then the nominal utilisation of the queue is

$$\rho = \frac{(x_1 r_H + x_2 r_L)}{C}.$$

Let $K$ be the threshold for marking in the virtual queue. It is suggested in [12] that one set $\kappa = (K+1)^{-1/K}$. Approximate the packet level queueing behaviour by an M/M/1 queue, then for the simplest strategy where just a single probing packet is used, the acceptance function is given by

$$a(x_1, x_2) = \max\left(0, 1 - \left(\frac{r_H x_1 + r_L x_2}{\kappa C}\right)^K\right), \text{ so } a(\rho) = \max\left(0, 1 - (K+1)\rho^K\right). \tag{2}$$

### 2.4 Extensions

In a network with several resources, indexed by $j$, a user or route $r$ may use a subset of these resources. Provided acceptance functions are independent across resources, we have the generalisation

$$a_r(\mathbf{x}) = \prod_{j \in r} a_j(\mathbf{x})$$

where $\mathbf{x}$ is the vector of the number of calls at the high and low rates for each route. This is legitimate when the acceptance decisions are conditionally independent given the mean load; this happens when there is a separation of timescales between the packet intervals and the probing intervals, as described in [12]

In general, an application may have many allowable levels, rather than just two, and may also choose not to enter. The theory of the next section applies equally well to multiple levels with rejection.

## 3 Analysis and Performance

The previous section defined an adaptive application at the microscopic level of individual calls. We now show that aggregate quantities satisfy certain stochastic differential equations, which leads to a functional weak law of large number and a corresponding functional central limit theorem.

### 3.1 Scaling and a Diffusion Limit

In order to examine the behaviour of the link as a large scale structure, we need to consider the correct large scale limit. This limit is defined over a family of systems indexed by $N$, which will be taken to tend to infinity. In the $N$th system, the arrival intensity (arrival rate divided by mean call length) is taken to be $\nu N$, whilst the mean call length is held fixed; thus the arrival rate grows linearly in $N$. The link capacity, that is the service rate of the queue, will be taken as $CN$ packets per second. It is natural to define $\tilde{X}_t^{(1)} = X_t^{(1)}/N$ and $\tilde{X}_t^{(2)} = X_t^{(2)}/N$. Under this scaling one might expect $a_N(\cdot)$, the acceptance function for the $N$th system, to scale so that $a_N(Nx_1, Nx_2) = a(x_1, x_2)$ for some fixed function $a$ (since as we noted earlier the $a$ function is typically a function of the utilisation $\rho$). For practical networks, we might typically be interested in $N$ of the order of $10^2$ or $10^3$.

Figure 1 shows how, when the system is allowed to reach its equilibrium point (which will be described later), the value of the nominal utilisation $\rho$ and the percentage of the traffic being carried at the high rate, depend upon the arrival rate $\nu$. For low values of $\nu$, it can be seen that almost all traffic is carried at the high rate, and for large $\nu$ almost all at the low rate.

If the network is operated in these large $\nu$ regions, packets will be lost. This will cause all users of the network to

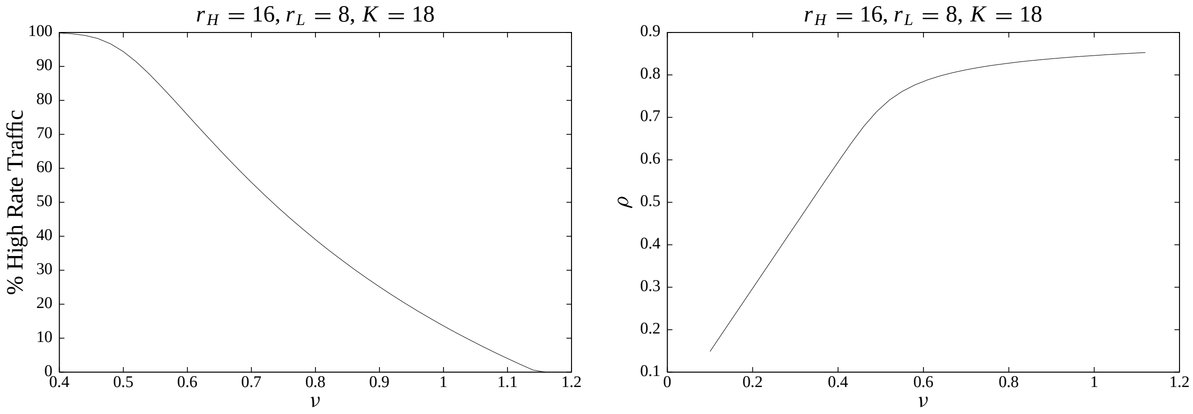

Figure 1: The effect of $\rho$ on the average percentage of high rate calls.

see a degradation in performance. This means that in practice it is necessary to reject some calls, which can be done in a similar distributed fashion by using multi-level probing.

When the network is operating at high $\nu$, in regions where we expect rejection to be significant, we would expect from the Figure 1 that almost all of the calls would be being carried at the low rate. This is effectively the situation which is analysed in [12]. So we shall not model the rejection mechanism explicitly here[1].

### 3.2 Fluid Limit

We shall see that in the limit as $N \to \infty$, one can show that the scaled network traffic $(\tilde{X}_t^{(1)}, \tilde{X}_t^{(2)})$ converges weakly to the fluid limit process, which is the process that solves the system of ordinary differential equations given by (1). Note that

$$\frac{\mathrm{d}}{\mathrm{d}t}(x_1(t) + x_2(t)) = \nu - (x_1(t) + x_2(t)),$$

which yields $x_1(t) + x_2(t) = \nu - Ce^{-t}$. Hence as $t \to \infty$, $x_1(t) + x_2(t) \to \nu$; thus the system dynamics are fully described by those on the manifold $x_1 + x_2 = \nu$.

An example of the trajectories of this system of these fluid equations is shown in Figure 2, for a probing rate of $\lambda = 1$. The equilibrium point is marked by the triangle. The individual curves correspond to different starting values.

### 3.3 Convergence and Stability

The fluid equations have an equilibrium point given by the solution of the equations

$$\overline{x}_1 + \overline{x}_2 = \nu, \qquad \overline{x}_1 = \nu a(\overline{x}_1, \nu - \overline{x}_1).$$

[1] The analytical techniques which are described here can easily be extended to cover explicit rejection.

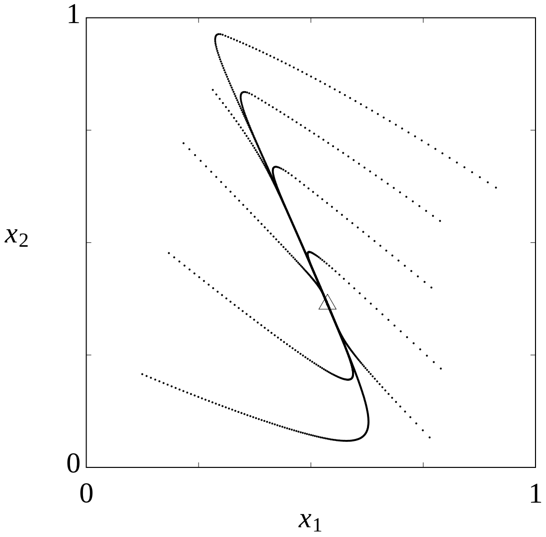

Figure 2: Trajectories of the Fluid Limit.

Note first that the equilibrium point is independent of $\lambda$ (the rate of in-call probing), and second that the fixed point is unique. Uniqueness follows from a simple monotonicity argument, since for any reasonable acceptance strategy, transferring calls from the low rate to the high rate should not increase the acceptance probability for a new call! The previous observation that the dynamics are described by those on the manifold $x_1 + x_2 = \nu$ means it suffices to consider the local stability about this point; if it is locally stable, then it must also be globally stable. Define $u_i = x_i - \overline{x}_i$, for $i = 1{,}2$ then linearising the fluid equations about their fixed point $(\bar{x}_1, \bar{x}_2)$ gives:

$$\begin{pmatrix} \dot{u}_1 \\ \dot{u}_2 \end{pmatrix} = H \begin{pmatrix} u_1 \\ u_2 \end{pmatrix},$$

where $H$ is given by

$$H = \begin{pmatrix} \nu a_x - 1 + \lambda((\overline{x}_1 + \overline{x}_2)a_x - 1 + a) & \nu a_y + \lambda((\overline{x}_1 + \overline{x}_2)a_y + a) \\ -\nu a_x - \lambda((\overline{x}_1 + \overline{x}_2)a_x - 1 + a) & -\nu a_y - 1 - \lambda((\overline{x}_1 + \overline{x}_2)a_y + a) \end{pmatrix},$$

and

$$a = a(\overline{x}_1, \overline{x}_2), \qquad a_x = \left.\frac{\partial a(x,y)}{\partial x}\right|_{\overline{x}_1, \overline{x}_2}, \qquad a_y = \left.\frac{\partial a(x,y)}{\partial y}\right|_{\overline{x}_1, \overline{x}_2}.$$

Direct evaluation of the determinant of $H$ shows that the eigenvalues are $-1$ and $\Psi$, where

$$\Psi = (1 + \lambda)\left(\nu(a_x - a_y) - 1\right).$$

Thus provided that,

$$a_x - a_y < \frac{1}{\nu},$$

both eigenvalues will be negative and so the fixed point will be stable. This condition is satisfied by the packet-level model we consider.

For the packet level model described by (2), the acceptance probability is a function of the nominal intensity $\rho$, implying that $a_x - a_y$ is some fraction of $a_x$; in addition the acceptance probability is non-increasing in $\rho$, or equivalently $a_x \leq 0$ hence the inequality is trivially satisfied.

When the acceptance function is a function of $\rho$, that is $a(x_1, x_2) = a(\rho)$, where $\rho = (x_1 r_H + x_2 r_L)/C$, then we can find Lyapunov functions for the system; this gives an alternative proof of our uniqueness and stability results.

### 3.4 Stability and Delays

There is a RTT delay between sending probe packets and receiving congestion indication signals. We have seen that when there is no such delay, the fixed point is stable for all values of $\lambda$. In the case when a delay is present we shall see that there is a maximum value of $\lambda$ for which the fixed point is stable. Let $D$ be the round trip delay, and assume the acceptance function is a function of $\rho$, with $\rho(t)$ defined naturally. Then the fluid limit equations are:

$$\begin{aligned}
\frac{\mathrm{d}x_1}{\mathrm{d}t} &= \nu a(\rho(t-D)) - x_1(t) + \lambda a(\rho(t-D))x_2(t) - \lambda(1 - a(\rho(t-D))x_1(t), \\
\frac{\mathrm{d}x_2}{\mathrm{d}t} &= \nu(1 - a(\rho(t-D)) - x_2(t) - \lambda a(\rho(t-D))x_2(t) + \lambda(1 - a(\rho(t-D)))x_1(t).
\end{aligned}$$

As before as $t \to \infty$, $x_1(t) + x_2(t) \to \nu$. Without loss of generality, put $r_H = 1$, $r_L = \frac{1}{2}$ and define the function $p(y) = 1 - a(y/C)$, then it suffices to consider the equation

$$\frac{\mathrm{d}x_1(t)}{\mathrm{d}t} = (1+\lambda)\left[\nu\left\{1 - p\left(\frac{x_1(t-D)}{2} + \frac{\nu}{2}\right)\right\} - x_1(t)\right].$$

Linearising about the fixed point, setting $\psi(t) = x_1(t) - \overline{x}_1$ with $\bar{y} = \bar{x}_1 + \bar{x}_2/2$ gives

$$\frac{\mathrm{d}\psi(t)}{\mathrm{d}t} = (1+\lambda)\left[-\frac{\nu}{2}p'(\bar{y})\psi(t-D) - \psi(t)\right].$$

Taking Laplace transforms gives the characteristic equation in $s$, whence setting $\gamma = sD$ we obtain

$$-D(1+\lambda)e^{\gamma} - \gamma e^{\gamma} - D(1+\lambda)\frac{\nu}{2}p'(\bar{y}) = 0.$$

This fixed point is stable if the roots of the characteristic equation both have negative real parts. By the theorem of Hayes [1, Theorem 13.8] this holds if and only if

$$-D(1+\lambda) < D(1+\lambda)\frac{\nu}{2}p'(\bar{y}) < \sqrt{\phi^2 + (D(1+\lambda))^2},$$

where $\phi$ is the root of $\phi = -D(1+\lambda)\tan\phi$ such that $0 < \phi < \pi$. (Note that in contrast to the no delay solution, increasing $\lambda$ has a negative impact on convergence). Since $D > 0$ and $\lambda > 0$ then $\phi > \pi/2$. Hence for virtual queue marking, using the notation of Section 2.3 and equation (2) for $a = 1 - p$ we have the sufficient condition

$$D(1+\lambda)\frac{\nu}{2c}(K+1)K\left(\frac{\bar{y}}{C}\right)^{K-1} < \sqrt{\left(\frac{\pi}{2}\right)^2 + (D(1+\lambda))^2}.$$

We are interested in the case $\rho < 1$, implying $\nu < 2C$, giving the sufficient condition

$$D(1+\lambda)(K+1) < \pi/2.$$

This gives us an upper bound on the reprobing rate $\lambda$ in order that the equilibrium point remain stable. For example with $K = 18$, we require $D(1+\lambda) < 0.08$ (a more exact calculation gives the right hand-side as 0.2). Recall that we have rescaled units so that the holding time is 1, and effectively $D$ is measured in holding time units, which in practice means there should be no stability problems. We are typically interested in flows lasting tens of seconds: for a mean holding time of 200s, and a round-trip time of 200ms we can allow $\lambda$ to increase to 80 and still have a stable system.

### 3.5 Variance and the Diffusion Limit

The foregoing has shown that there is no significant dynamic behaviour described by the fluid limit. The network traffic simply converges to the unique fixed point. This fluid limit gives insufficient information to assess the performance of the network, so now we look at a finer level of detail.

Let $\mathbf{U}(t)$ be the vector of differences of the traffic vector from the equilibrium, so $\mathbf{U}_t = (\tilde{X}_t^{(1)} - \overline{x}_1, \tilde{X}_t^{(2)} - \overline{x}_2)$. Then the following central limit theorem holds, the proof of which may be found in the Appendix.

**Theorem 1.** *In the limit as $N \to \infty$, the deviations from the equilibrium fluid limit satisfy*

$$\lim_{N\to\infty} \sqrt{N}\mathbf{U}(t) \overset{\mathcal{D}}{=} R(t),$$

*where $R(t)$ is the unique solution of the stochastic differential equation*

$$\mathrm{d}\mathbf{R}_t = H\mathbf{R}_t\mathrm{d}t + F\mathrm{d}\mathbf{B}_t, \tag{3}$$

*where $H$ and $F$ are matrices, and $\mathbf{B}_t$ is a six dimensional Brownian motion. This equation describes an Ornstein-Uhlenbeck process. The matrices are given by*

$$H = \begin{pmatrix} \nu a_x - 1 + \lambda(a_x(\overline{x}_1 + \overline{x}_2) - 1 + a) & \nu a_y + \lambda(a_y(\overline{x}_1 + \overline{x}_2) + a) \\ -\nu a_x - \lambda(a_x(\overline{x}_1 + \overline{x}_2) - 1 + a) & -\nu a_y - 1 - \lambda(a_y(\overline{x}_1 + \overline{x}_2) + a) \end{pmatrix},$$

*and*

$$F = \begin{pmatrix} \sqrt{a\nu} & -\sqrt{\overline{x}_1} & \sqrt{a\lambda\overline{x}_2} & -\sqrt{(1-a)\lambda\overline{x}_1} & 0 & 0 \\ 0 & 0 & -\sqrt{a\lambda\overline{x}_2} & \sqrt{(1-a)\lambda\overline{x}_1} & \sqrt{\nu(1-a)} & -\sqrt{\overline{x}_2} \end{pmatrix}$$

*and*

$$a = a(\overline{x}_1, \overline{x}_2), \qquad a_x = \left.\frac{\partial a(x,y)}{\partial x}\right|_{\overline{x}_1,\overline{x}_2}, \qquad a_y = \left.\frac{\partial a(x,y)}{\partial y}\right|_{\overline{x}_1,\overline{x}_2}.$$

This theorem has the following corollary.

**Corollary 2.** *At fixed time $t$, the scaled difference vector $\sqrt{N}U_t$ is distributed according to a multivariate normal distribution with mean zero and covariance matrix $\Sigma$ given by*

$$\Sigma = \int_{-\infty}^{0} e^{-uH}FF^T(e^{-uH})^T\mathrm{d}u,$$

*where $F$ and $H$ are as defined in the previous theorem.*

If the matrix $H$ is diagonalisable then the following decomposition simplifies the evaluation of the above covariance, since we may write

$$H = \Gamma\Phi\Gamma^{-1},$$

where $\Phi$ is a diagonal matrix, with the eigenvalues of $H$, namely $\phi_1$ and $\phi_2$ along the diagonal. Following simple manipulation

$$\Sigma = \left(\Gamma\left[\int_{-\infty}^{0} e^{-u\Phi}\Gamma^{-1}FF^T\left(\Gamma^{-1}\right)^T e^{-u\Phi}\mathrm{d}u\right]\Gamma^T\right).$$

The integral inside the square brackets may be expressed directly terms of the eigenvalues of $H$, so

$$\left[\int_{-\infty}^{0} e^{-u\Phi}\Gamma^{-1}FF^{T}\left(\Gamma^{-1}\right)^{T}e^{-u\Phi}\mathrm{d}u\right]_{r,s} = \frac{\left(\Gamma^{-1}FF^{T}\left(\Gamma^{-1}\right)^{T}\right)_{r,s}}{\phi_r+\phi_s}.$$

Hence the steps needed in a computation of the covariance matrix are:

(i) Find the equilibrium point. This may be done by numerical techniques, or numerical integration of the fluid limit equation. In many cases this equilibrium point is independent of $\lambda$, so this step does not need to be repeated when considering the effect of varying $\lambda$.

(ii) Calculate the values $a$, $a_x$ and $a_y$ at this equilibrium point, and hence evaluate the matrices $H$ and $F$.

(iii) Diagonalise the matrix $H$ and find the eigenvectors, giving the matrix $\Gamma$.

(iv) From the above result, evaluate the covariance matrix by simple matrix multiplication.

### 3.6 Example

We let the acceptance probability be given by (2) and consider the total traffic variance, which may be expressed in terms of the covariance matrix $\Sigma$ as

$$v(\lambda) = r_H^2\Sigma_{11} + r_H r_L\left(\Sigma_{12}+\Sigma_{21}\right) + r_L^2\Sigma_{22}.$$

We evaluate this numerically and plot it against $\lambda$. Figure 3 shows the effect of the threshold value $K$ upon the shape of the acceptance surface, and hence upon $v(\lambda)$. Recall that the acceptance functions give the probability of accepting at the high rate, which decreases as the load increases. The variance decreases sharply with $\lambda$: a value of $\lambda = 1$, corresponding to just one in-call probe per call on average, reduces the variance of the carried traffic by about 30%; while for $\lambda = 7$ the reduction is about 50%. Little is gained by having much larger values of $\lambda$.

## 4 Results and Performance

### 4.1 Simulation Methodology

The analysis has used some simplifications in the interests of tractability. Significant assumptions are that the calls have exponentially distributed lengths, and that the round-trip delay for packets has a negligible effect on the limiting process. To look at the effect of these assumptions, the *ns* network simulator was used to model various scenarios,

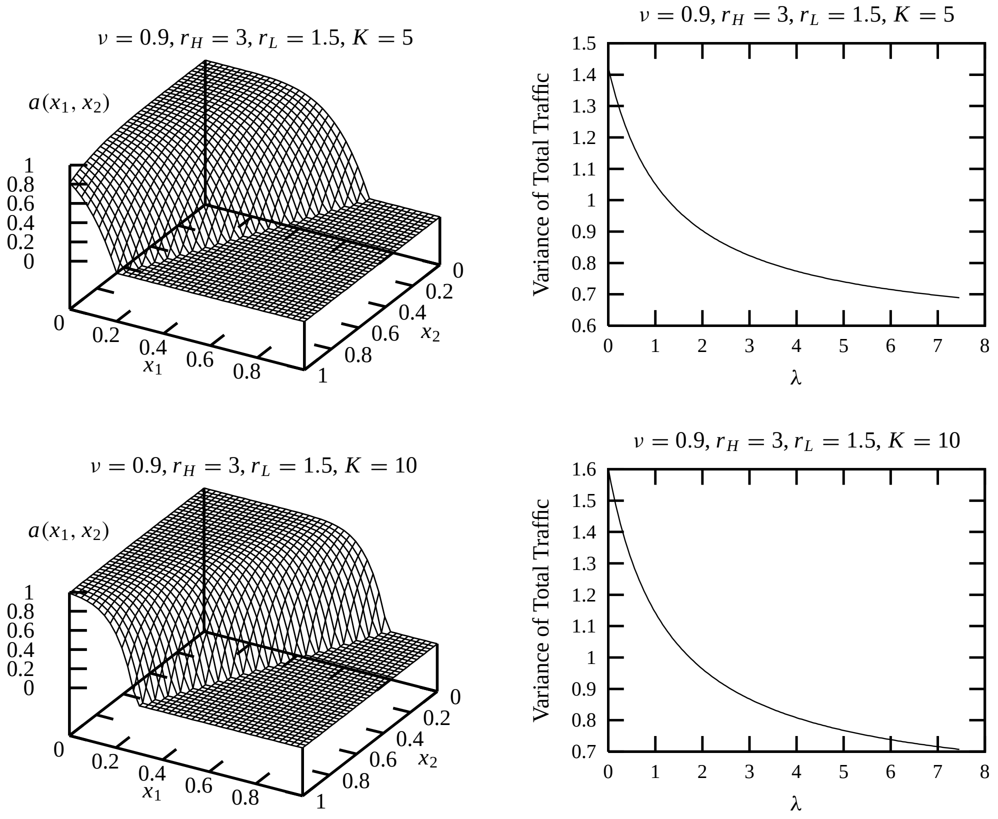

Figure 3: Acceptance probabilities and Total Traffic Variance as a function of $\lambda$.

such as a single-link and a star-network. Enhancements to *ns* were created to allow in-call probing and adaptation based on ECN marks.

The ‘calls’ in the simulation resemble typical voice calls carried over an IP network. The high data rate has been chosen as 64kbit/s (the value used in the ITU G711 PCM encoding of voice), and the low rate chosen as 32kbit/s.

Calls arrive as a Poisson process of rate $\nu$ per holding time; each call lasts for an exponential period of time, with mean 200s. Each call also reprobes at rate $\lambda/200$ (as a Poisson process, with the scaling such that a value of $\lambda = 1$ corresponds on average to one “in-call” probe per call). Each call generates a stream of 500 byte UDP packets.

In the simulations the acceptance strategy is implemented through the following mechanism. When the destination receives a packet with the CE bit set, it immediately send a small *mark indication* UDP datagram back to the source. Each source uses a single packet to probe the network and enters at the high rate if a mark indication packet is not received back by the source; otherwise it enters at the low rate. It is possible that the probe packet is lost, in which case no CE mark will be received. As a consequence, the implied acceptance probability $a(\rho)$ will differ slightly from that of (2). Note that the call traffic itself is being used as the probe, so there are no issues of whether probe packets can cause network overload.

Congestion detection used the virtual queue marking algorithm in routers, as described in section 2, which has

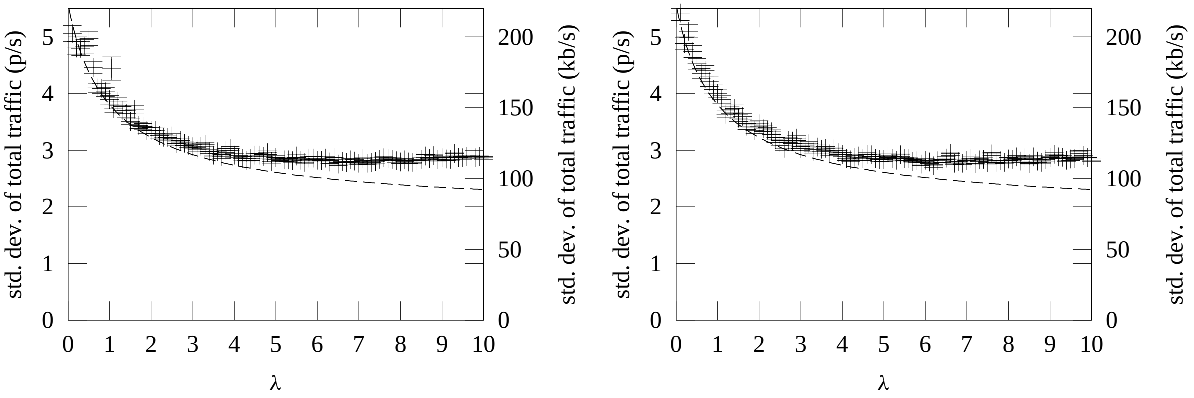

Figure 4: Traffic Standard Deviation against λ for links with a delay of 10ms (left) and 100ms (right). The theoretical prediction is shown by the dashed line.

one free parameter $K$, and marks packets (sets the CE bit) if the virtual buffer exceeds $K$, where $K = 18$. The router itself has a buffer capacity for 20 packets.

### 4.2 Single Link Results

In this section we use $N = 100$, and $\nu = 0.9$ in scaled units, so the real arrival rate is 90 per holding time. Link capacity was set to 4.3Mb/s. Figure 4 compares simulation with theoretical results, as the probing rate $\lambda$ (which is plotted normalised to a unit call holding time) is altered. The theoretical results were computed using an acceptance function which reflects the acceptance strategy used in the simulation. The error bars show the 95% confidence intervals for the computed variance of the traffic on the link during the simulations, where each simulation represents 100000 seconds. Notice first the excellent agreement with theory, and secondly the insensitivity to round trip times. Increasing $\lambda$ from zero to four reduces the variability ($\pm 2\sigma$) by about 10% of the link capacity. For $\lambda$ significantly greater than 10, further simulations showed that the variance of the total traffic increased with $\lambda$, indicating that the effect of the round trip time was no longer negligible at such high probing rates.

### 4.3 Sensitivity

In the theoretical analysis, the call holding times were assumed to be exponential with mean one. Figure 5 shows the effect on the standard deviation of the traffic as a function of $\lambda$, when the call holding times are heavy tailed, distributed according to a Pareto distribution with mean 1 and shape parameter $\beta = 1.2$, or 1.5. Although the initial decay in traffic variance as $\lambda$ increases is slightly less rapid than in the case of exponential call holding times, for $\lambda = 4$, the variability has decreased by approximately 50%, indicating that the "in-call" probing strategy is just as useful as when the holding times were distributed according to an exponential distribution.

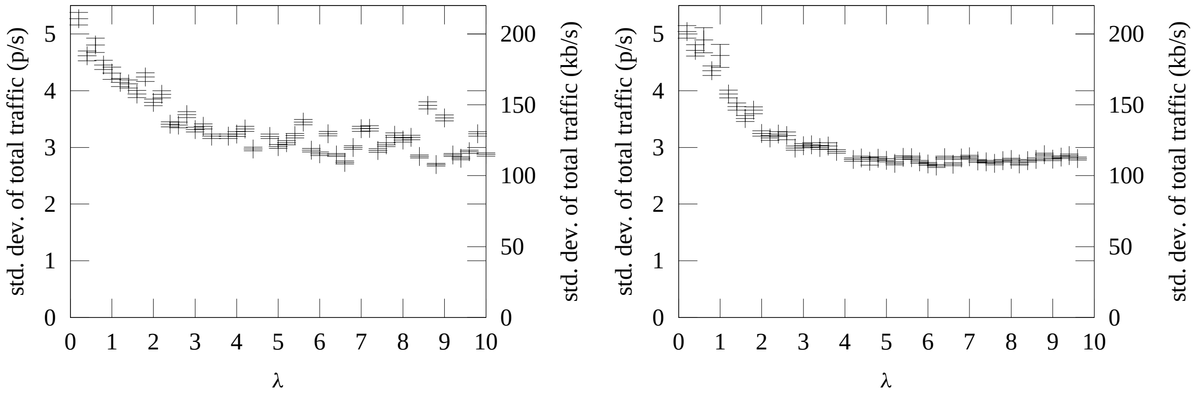

Figure 5: Pareto Call Holding times, shape parameters 1.2 (left) and 1.5 (right).

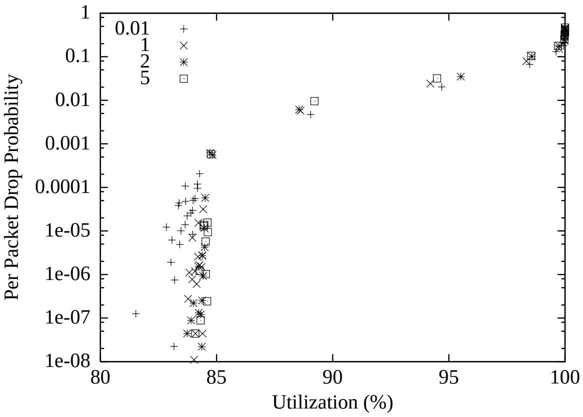

Figure 6: Per Packet Drop Probability against Percentage Utilisation of Link, for $\lambda = 0.01$, 1, 2 and 5.

### 4.4 Packet Loss versus Utilisation

When the issue of rejection was introduced it was mentioned that effectively the objective of a good distributed acceptance policy was to maintain the value of $\nu$ sufficiently low that the packet loss rate did not become unacceptable. To see the effect of this in practice a series of runs with different values of $\nu$ were performed, thus achieving different link utilisation. Figure 6 shows the trade-off between utilisation and packet loss for different values of $\lambda$. Below about 85% loading, there is minimal packet loss, and utilisation improves as $\lambda$ increases. Above this loading, the VQ strategy marks almost all packets so calls only enter at the low rate, increasing $\lambda$ has an almost negligible benefit (as there is no room to manoeuvre), and packet drop increases. At 90% load about 1% of packets are lost.

### 4.5 Star Network

The theory and practice explained so far also apply in the context of networks with product form acceptance functions as was explained in Section 2.4. In the same way as for the single link, one may write down equations for the traffic along each route (source node to destination node) and these may be summed over the routes using a link to obtain

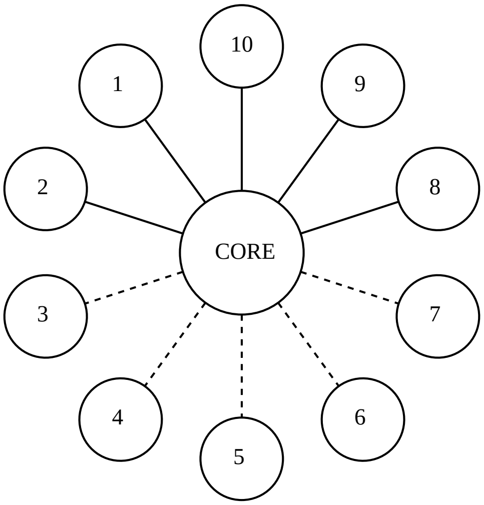

Figure 7: The star network topology. Solid lines denote links with a delay of 10ms, and dashed lines those with a 100ms delay (all links are duplex).

expressions for the traffic along each link. A similar analysis may be used to predict the central limit covariance matrix.

We consider here a simple star topology example, illustrated in Figure 7. Calls originate from the end nodes (which are numbered from one to ten), and each call connects with a randomly chosen distinct edge node. Thus there are 45 possible routes. This may be motivated by an Internet architecture with a transparent core switch connected to routers by a variety of links. These routers are represented in the model by the edge nodes since they aggregate calls arriving from a number of individual sources.

The result of simulating this network and measuring the variance of the aggregate traffic along one of the 10ms and one of the 100ms delay links is shown in Figure 8. These simulations were very time consuming; the theoretical model may readily be solved, so that computation of the traffic variance requires only the diagonalisation of a $90 \times 90$ matrix. The simulation results show a similar reduction in variance to those for the single link, however the majority of the effect has been achieved by $\lambda = 1$, which is a quarter of the value observed for a single link.

## 5 Concluding Remarks

We have illustrated the benefits of in-call probing for applications that can adjust their rate. The approach is light-weight – applications look at whether a small number of probing packets are marked, and use this information to decide whether or not to alter their rate. No extra signalling channel is required, and the connection set-up is very fast. We looked at the simplest possible scheme, in which all calls are accepted, and an application sends just one probe packet into the network, if this is marked it chooses the low-rate, otherwise the high rate. with the same

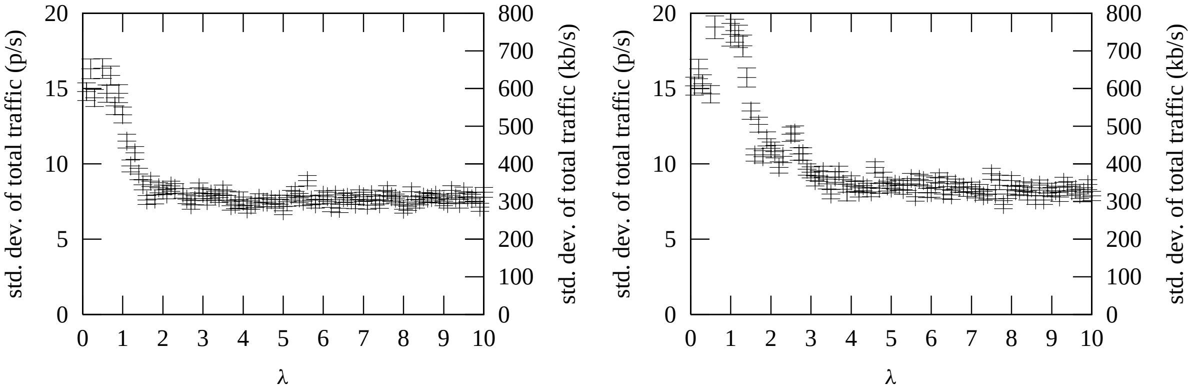

Figure 8: Measured traffic standard deviation as a function of λ for a 10ms delay link (left) and a 100ms delay link (right) in the star network.

policy used at the start of the connection and during the connection. We constructed a diffusion limit to quantify the benefits of probing, which agrees well with simulation, is tractable and can be applied to general networks. Results show that whilst the mean traffic flow on the network is unaffected by the probing, just a small amount of in-call probing significantly improves the behaviour of the system: allowing on average just one in-call change decreases the variance of the system significantly, and allowing between 5 and 10 in-call changes gives almost all of the possible gains, reducing the traffic variance by 50% in some cases. This benefits the system, and means that users also do better in the long run.

We have concentrated on the case where the application has just two admissible transmission rates; the theory applies equally well to more levels. We have addressed call rejection through the use of a reduced arrival rate. In practice, call rejection is naturally implemented by having at least three reactions to probing packets. For example send $n \geq 2$ probe packets into the network, if $m$ is the number that are marked, send at the high rate if $m < m_1$, at the low rate if $m_1 \leq m < m_2$ and reject otherwise. Of course, different applications might chose different values of the probing parameters $(n, m_i)$ reflecting their differing requirements.

To implement such in-call probing, marks have to be fed back to applications, and we have discussed ways in which the ECN proposal could be suitably adapted. Alternatively, loss could be used as the feedback signal.

We have not dwelt on how adaptive traffic should be integrated with other traffic. There are a number of possibilities: such traffic could be segregated into a separate DiffServ class. If end-system behaviour is enforced in some way then soft-guarantees could be given on packet loss and rejection. If such traffic is not segregated, then guarantees are necessarily weakened. The theory can be adapted to the integrated case by suitably altering the acceptance strategy with altered probing behaviour, such as sending more probe packets to allow for marking caused by high unresponsive load.

## References

[1] R. Bellman and K. L. Cooke, *Differential-difference equations*, Academic Press, New York, 1963.

[2] G. Bianchi, A. Capone, and C. Petrioli, *Throughput analysis of end-to-end measurement based admission control in IP*, INFOCOM 2000, IEEE, 2000, `http://www.comnet.technion.ac.il/infocom2000`.

[3] J-C. Bolot and T. Turletti, *Experience with control mechanisms for packet video in the Internet*, Computer Communication Review **28** (1998), no. 1, 4–15.

[4] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, *Resource ReSerVation Protocol (RSVP) version 1 functional specification*, RFC 2205, Internet Engineering Task Force, September 1997.

[5] L. Breslau, E. Knightly, S. Shenker, I. Stoica, and H. Zhang, *Endpoint admission control: Architectural issues and performance*, Sigcomm 2000, August 2000.

[6] A. Daguiklas and M. Ghanbari, *Rate-based flow control of video services in ATM networks*, Globecom ’96 (London), vol. 1, IEEE, November 1996.

[7] V. Elek, G. Karlsson, and R Rönngren, *Admission control based on end-to-end measurements*, INFOCOM 2000, IEEE, 2000, `http://www.comnet.technion.ac.il/infocom2000`.

[8] Stewart Ethier and Tom Kurtz, *Markov processes – characterization and convergence*, Wiley, 1986.

[9] S. Floyd and V. Jacobson, *Random Early Detection gateways for congestion avoidance*, IEEE/ACM Transactions on Networking **1** (1993), no. 4, 397–413.

[10] *Recommendation G.723.1: Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.4 kbit/s*, 1996, ITU.

[11] R. J. Gibbens and F. P. Kelly, *Distributed connection acceptance control for a connectionless network*, Teletraffic Engineering in a Competitive World, Proceedings ITC16 (P.B. Key and D.G. Smith, eds.), Elsevier, June 1999, pp. 941–952.

[12] F. P. Kelly, P. B. Key, and S. Zachary, *Distributed admission control*, IEEE Journal on Selected Areas in Communications **18** (2000), no. 12.

[13] Thomas G. Kurtz, *Strong approximation theorems for density dependent markov chains*, Stochastic Process. Appl. **6** (1978), 223–240.

[14] A. Mandelbaum, W. A. Massey, and M. I. Reiman, *Strong approximations for markovian service networks*, Queueing Systems – Theory and Applications **30** (1998), 149–201.

[15] K. Ramakrishnan and S. Floyd, *A proposal to add explicit congestion notification (ECN) to IP*, RFC 2481, IETF, January 1999.

[16] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, *RFC 1889: RTP: A transport protocol for real-time applications*, 1996.

[17] Z. R. Turányi and L. Westberg, *Load control: lightweight provisioning of internet resources*, `http://www.ericsson.co.hu/ethzrt/`, 1999.

## Appendix

### Proof of Central Limit Theorem and Weak Limit Theorem

This result will be proved in the case that the function $a(x_1, x_2)$ is a continuous Lipshitz function. A sufficient condition for this is that $|\nabla a|$ be bounded. This result can be extended to locally Lipshitz $a(x_1, x_2)$, but the simpler Lipshitz case is usually sufficient for a good approximation to the induced acceptance probability functions from real acceptance strategies.

*Proof.* Let $X_t^{(1)}$ and $X_t^{(2)}$ be stochastic processes describing the number of calls in progress at the low and high data rates at time $t$ respectively, then these satisfy the following Poisson counter driver stochastic differential equations:

$$\begin{aligned}\mathrm{d}X_t^{(1)} = \mathrm{d}A\left(\nu_N \int_0^t a_N(Y_\tau^{(1)}, Y_\tau^{(2)})\mathrm{d}\tau\right) - dB\left(\int_0^t Y_\tau^{(1)}\mathrm{d}\tau\right)& \\ + \mathrm{d}C\left(\lambda \int_0^t a_N(Y_\tau^{(1)}, Y_\tau^{(2)})Y_\tau^{(2)}\mathrm{d}\tau\right) - \mathrm{d}D\left(\lambda \int_0^t \left(1 - a_N(Y_\tau^{(1)}, Y_\tau^{(2)})\right) Y_\tau^{(1)}\mathrm{d}\tau\right),& \qquad (4)\end{aligned}$$

and

$$\begin{aligned}\mathrm{d}X_t^{(2)} = \mathrm{d}\tilde{A}\left(\nu_N \int_0^t \left(1 - a_N(Y_\tau^{(1)}, Y_\tau^{(2)})\right)\mathrm{d}\tau\right) - \mathrm{d}\tilde{B}\left(\int_0^t Y_\tau^{(2)}\mathrm{d}\tau\right)& \\ - \mathrm{d}C\left(\lambda \int_0^t a_N(Y_\tau^{(1)}, Y_\tau^{(2)})Y_\tau^{(2)}\mathrm{d}\tau\right) + \mathrm{d}D\left(\lambda \int_0^t \left(1 - a_N(Y_\tau^{(1)}, Y_\tau^{(2)})\right) Y_\tau^{(1)}\mathrm{d}\tau\right),&\end{aligned}$$

where $A$, $B$, $\tilde{A}$, $\tilde{B}$, $C$, and $D$ are mutually independent unity rate Poisson processes. $X_t^{(1)} = Y_t^{(1)}/N$, and $X_t^{(2)} = Y_t^{(2)}/N$. We also assume that $a_N$ has the scaling behaviour $a_N(X_t^{(1)}, X_t^{(2)}) = a(\tilde{X}_t^{(1)}, \tilde{X}_t^{(2)})$. The following is then a scaled version of the SDEs:

$$\begin{aligned}\mathrm{d}\tilde{X}_t^{(1)} = \frac{1}{N}\mathrm{d}A\left(N\nu\int_0^t a(X_\tau^{(1)}, X_\tau^{(2)})\mathrm{d}\tau\right) - \frac{1}{N}\mathrm{d}B\left(N\int_0^t X_\tau^{(1)}\mathrm{d}\tau\right) \\ + \frac{1}{N}\mathrm{d}C\left(N\lambda\int_0^t a(X_\tau^{(1)}, X_\tau^{(2)})X_\tau^{(2)}\mathrm{d}\tau\right) - \frac{1}{N}\mathrm{d}D\left(\lambda N\int_0^t \left(1 - a(X_\tau^{(1)}, X_\tau^{(2)})\right)X_\tau^{(2)}\mathrm{d}\tau\right),\end{aligned}$$

and

$$\begin{aligned}\mathrm{d}\tilde{X}_t^{(2)} = \frac{1}{N}\mathrm{d}\tilde{A}\left(\nu\int_0^t \left(1 - a(X_\tau^{(1)}, X_\tau^{(2)})\right)\mathrm{d}\tau\right) - \frac{1}{N}\mathrm{d}\tilde{B}\left(N\int_0^t X_\tau^{(2)}\mathrm{d}\tau\right) \\ - \frac{1}{N}\mathrm{d}C\left(\lambda N\int_0^t a(X_\tau^{(1)}, X_\tau^{(2)})X_\tau^{(2)}\mathrm{d}\tau\right) + \frac{1}{N}\mathrm{d}D\left(\lambda N\int_0^t \left(1 - a(X_\tau^{(1)}, X_\tau^{(2)})\right)X_\tau^{(1)}\mathrm{d}\tau\right).\end{aligned}$$

Let $\mathbf{Z}^N$ be the vector $(\tilde{X}_t^{(1)}, \tilde{X}_t^{(2)})$, this can then be expressed in vector form as

$$\mathbf{Z}^N(t) = \mathbf{Z}^N(0) + \sum_{i\in I} A_i\left(N\int_0^t \alpha_s\left(\frac{1}{N}\mathbf{Z}^N(s), i\right)\mathrm{d}s\right)\mathbf{v}_i.$$

where the vectors $\mathbf{v}_i$ and the rate functions $\alpha_i$ for $i = 1, \ldots, 6$ are defined as follows:

$$\begin{array}{llll}\mathbf{v}_1 = \begin{pmatrix}1\\0\end{pmatrix} & \alpha(\mathbf{x}, 1) = \nu a(x, y) & \mathbf{v}_2 = \begin{pmatrix}-1\\0\end{pmatrix} & \alpha(\mathbf{x}, 2) = x \\ \mathbf{v}_3 = \begin{pmatrix}1\\-1\end{pmatrix} & \alpha(\mathbf{x}, 3) = \lambda a(x, y)y & \mathbf{v}_4 = \begin{pmatrix}-1\\1\end{pmatrix} & \alpha(\mathbf{x}, 4) = \lambda(1 - a(x, y))x \\ \mathbf{v}_5 = \begin{pmatrix}0\\1\end{pmatrix} & \alpha(\mathbf{x}, 5) = \nu\,(1 - a(x, y)) & \mathbf{v}_6 = \begin{pmatrix}0\\-1\end{pmatrix} & \alpha(\mathbf{x}, 6) = y\end{array}$$

Under our assumption that $a$ is Lipshitz, it follows that all of the other relevant rate functions ($\alpha_1$ through $\alpha_6$ are also Lipshitz). They are also all continuous. It is now possible through the introduction of suitable initial conditions to apply Kurtz's theorems (see [8, 13, 14]) to establish the weak convergence to the fluid limit

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \sum_{i=1}^{6}\alpha(\mathbf{x}(t), i)\mathbf{v}_i,$$

and also the CLT about the fluid limit, namely that

$$\lim_{N\to\infty}\left[\mathbf{X}_t^{(N)} - \mathbf{x}(t)\right] \overset{\mathcal{D}}{=} \mathbf{R}_t,$$

where $\mathbf{R}_t$ solves the stochastic differential equation

$$\mathrm{d}\mathbf{R}_t = \sum_{i=1}^{6} \left(\nabla\alpha(\mathbf{x}(t), i) \cdot \mathbf{R}(t)\right) \mathbf{v}_i \, \mathrm{d}t + \sum_{i=1}^{6} \sqrt{\alpha(\mathbf{x}(t), i)} \, \mathrm{d}B_i(t)\mathbf{v}_i \, ,$$

where $B_i(t)$ is the $i$th component of a six dimensional standard Brownian Motion process. The most useful application of this is when the fluid limit is the time-independent equilibrium distribution, that is $\mathbf{x}(t) = (\overline{x}_1, \overline{x}_2)$, when the result simplifies to that which was stated. □