Bufferless Switching of TCP Traffic

Teunis J. Ott

August 3, 2001

Abstract

Abstract goes here.

1 Introduction

In purely Optical packet routers, with statistical per packet routing and no O-E-O conversion, switching is necessarily bufferless. In this situation, there is a bundle of a number of wavelengths associated with a "next interface". Whenever a packet must exit the router through that interface and at least one wavelength is free, the packet uses one of the free wavelengths. When no wavelength is free, the packet is discarded.

Since we are considering an all-optical router, the assumption is that input also is purely optical and that all wavelengths have exactly the same bitrate. When a packet arrives, its optical label is read. Then, while the optical label is processed, the packet goes through an optical delay loop or a number of optical delay loops. When the packet starts exiting from the optical delay circuit, the output interface and wavelength have been determined and the packet is "cut-through" to the chosen output interface and wavelength.

This model assumes availability of wavelength conversion.

In this paper we study the situation where the offered traffic to an output interface consists of TCP traffic.

With TCP traffic, when a packet is dropped, the TCP feedback mechanism soon causes the sending socket to reduce its TCP congestion window. This causes a reduction in the arrival rate.

In a thought-experiment, we distinguish three different rates:

- **The Inherent Rate**

  This is the arrival rate (bytes per second) the interface would see if it had an infinite capacity, for example if it had an infinite number of wavelengths.

- **The Offered Load**

  This is the actual arrival rate (bytes per second) the interface sees. If the interface has measurent device to count bytes attempting to use the interface, the measurement device measures "Offered Load".

- **Carried Load** This is the actual number of bytes per second that is carried (serviced) by the interface. Again, a measurement device measuring "carried load" can measure this entity.

As noted above, offered load an carried load can in principle be measured at the interface in question. "Inherent Load" can not, but offered load and carried load *can*. The loss probability (frequency, on a per byte basis), is

$$\frac{offered\,load - carried\,load}{offered\,load}. \tag{1.1}$$

Similarly, loss probability per packet (etc) can be defined.

Intuitively, it ought to be possible to deduce the "inherent load" from offered load and carried load: If the offered load and drop probabilty are high, the inherent load must be quite high. In fact, this is not an easy exercise: it depends on Round Trip Times, TCP Windows, etc, of the TCP connections involved.

In this paper we will study a special case where the "inherent load" is well defined, and we will show that in that case the TCP end-to-end protocol makes the system quite well behaved under overload. An issue highlighted is that under overload (in terms of a very high inherent load) both offered load and drop probability can seem very acceptable, given an unsophisticated operator the impression there is no overload.

It must be noted that in bufferless switching as above, under low load the drop probability is low, while at increasing (offered) loads the drop probability increases. This, bufferless switching has a built-in form of RED (Random Early Detection), see.

2 The Erlang B model, and the Square Root formula for TCP

In this Section we combine the Erlang B formula with the Square Root formula for TCP to formulate the problem of predicting the offered load and the drop probability as a fixed point problem.

To achieve a common unit for packet sizes and bandwidths, we express bandwidths in bytes per second, and packet sizes in bytes.

Let the bufferless interface have N wavelengths with each bandwidth b (bytes per second). The total bandwidth of the outgoing bundle is $N \times b$ bytes per second.

Let there be M TCP connections (data only) using the bundle. Let the round trip times of these M connections be $(RTT_1, RTT_2, \cdots, RTT_M)$, and let the MSSs (Maximal

Segment Sizes, i.e. Packet Sizes) of these TCP connections be $(PS_1, PS_2, \cdots, PS_M)$. We assume that all M TCP connections are sending a very large file ("persistent TCP").

Let the drop probability at the bufferless interface be p. We assume this is the same for all TCP connections. In this Section we also assume none of the TCP connections drops packets at some other bottleneck in the network, and that all TCP connections have a sufficiently large advertised window and "TCP Window". In the next section we will allow "external loss", i.e. loss in other places than the link under consideration.

By the Square Root Formula for TCP, see e.g. [2] , if there is no delayed acknowledgement, TCP connection m has in average a congestion window of $\frac{1.527}{\sqrt{p}}$ MSSs, and an average throughput of $\frac{1.707 \times PS_m}{RTT_m \times \sqrt{p}}$ bytes per second. If there is delayed acknowledgement these numbers have to be divided by two. For the time being, we will use 1.527 and 1.707. We will see that having these coefficients wrong by a factor 2 does not make much difference in the final prediction of the offered load or loss probability: The model used is remarkably robust.

The total throughput of the M TCP connections together therefore is

$$\frac{1.707}{\sqrt{p}} \sum_{m=1}^{M} \frac{PS_m}{RTT_m} \tag{2.1}$$

(bytes per second).

Making one more assumption, we assume that the number of input interfaces, and the total numbers of wavelengths in these interfaces, and the way the total load to the output interface is coming from the input wavelengths, etc., is such that we can use the Erlang B formula to predict the loss probability in the bufferless output interface. Thus,

$$p = B(N, \frac{1.707}{b\sqrt{p}} \sum_{m=1}^{M} \frac{PS_m}{RTT_m}) \tag{2.2}$$

.

We denote

$$R = \frac{1.707}{b} \sum_{m=1}^{M} \frac{PS_m}{RTT_m}. \tag{2.3}$$

$\frac{R}{\sqrt{p}}$ is the offered load expressed in average number of wavelengths. Thus, for a sensible system system we must find that $\frac{R}{\sqrt{p}}$ is less than N, but a sizable fraction of N. N and R are known, and p must be solved from

$$p = B(N, \frac{R}{\sqrt{p}}). \tag{2.4}$$

Here, $B(N, y)$ is the Erlang B function:

$$B(N, y) = \frac{\frac{y^N}{N!}}{\sum_{j=0}^{N} \frac{y^j}{j!}} \tag{2.5}$$

It is convenient to define $x = \frac{R}{N\sqrt{p}}$ and to re-write 2.4 as

$$B(N, Nx) = \left(\frac{R}{Nx}\right)^2. \tag{2.6}$$

x, solved from 2.6, is the predicted offered load, as fraction of $N \times b$.

For $N = 100$ and $\frac{1}{2} \leq R \leq 32$ the solution to 2.4 is given in the Figures 1 and 2.

The solution techniques will be described in ... .

The next two paragraphs must be updated: numbers are wrong.

It is interesting to note that the actual offered load can increase above N. When the Inherent Load reaches 295, the actual offered load as predicted by the model reaches $1.00177 \times N$, and the predicted drop probability reaches $p = 0.0767175$, so that the predicted carried load becomes $0.9249 \times N$.

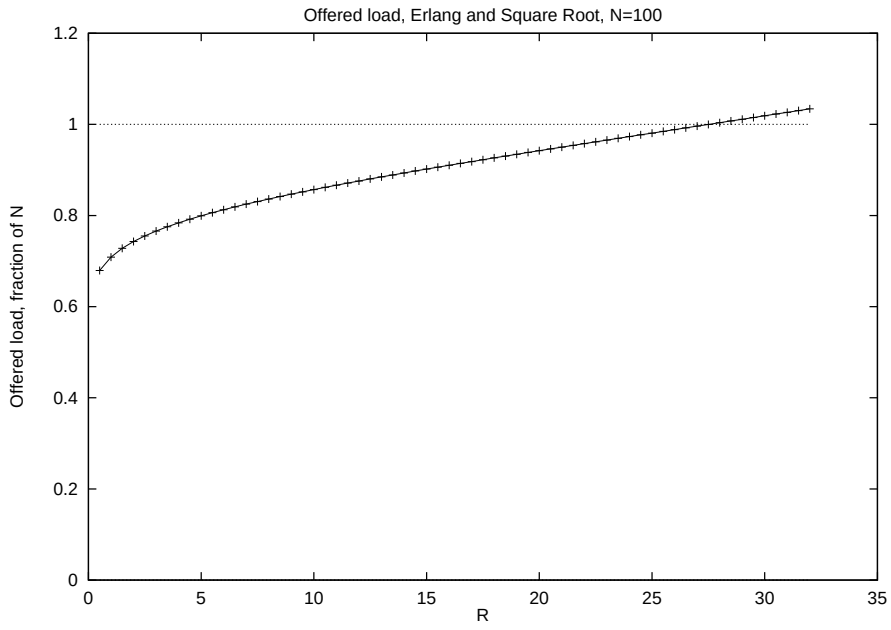

Figure 1: Offered Load (fraction of N), without external loss

When the Inherent Load reaches 320 these numbers become $1.02135 \times N$, $p = 0.0881629$, predicted carried equals $.9313 \times N$.

We see that the predicted offered load increases from $x \sim .71$ at $R = 1$ to $x \sim .96$ at $R = 22$, while over the same range the drop probability (per packet) increases from $p \sim .0002$ to $p \sim .053$.

We see that the predicted offered load is fairly independent of R. The explanation is that low R (say few TCP flows, or TCP flows with large RTTs) the drop probability is low, so that the congestion windows are large, and the predicted offered load still is large. When R increases also p increases, the congestion windows decrease, and the predicted offered load increases only moderately. The basic reason therefore is that around $x = .8$ the drop probability as predicted by the Erlang B formula ($p = B(N, Nx)$ is a fairly steeply increasing function of x.

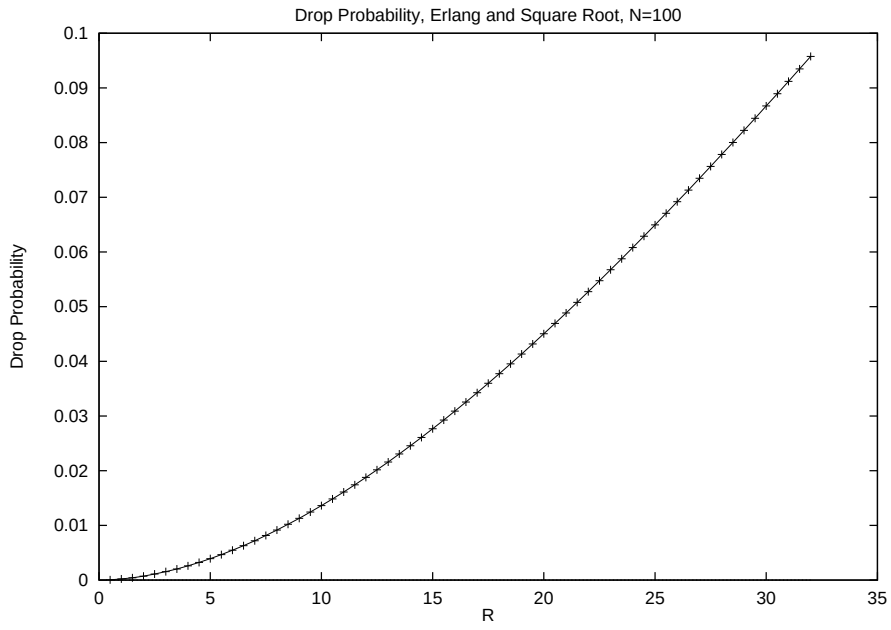

Figure 2: Drop probability, without external loss

3 The case with external loss

In the situation of 2.4 and 2.6 there is no "inherent load": If the link had infinite capacity, we would have $p = 0.0$, and therefore "infinitely large" offered load.

One way of getting a well-defined "inherent load" is to assume that all packets have a drop probability $a > 0$ outside the link being studied. If the drop probability in the link being studied is p, the total drop probability (assuming independence) is $p + a - pa$, and the total (predicted) offered load is $\frac{R}{\sqrt{p+a-pa}}$ (expressed in bandwidths).

This gives an "inherent offered load" (at $p = 0$) of $\frac{R}{\sqrt{a}}$.

We assume that p and a both are small and approximate $p + a - pa \sim p + a$. The predicted offered load as function of N and R now is computed by solving p or $x = \frac{R}{N\sqrt{p+a}}$ (and hence $p = \left(\frac{R}{Nx}\right)^2 - a$) from either

$$B(N, \frac{R}{\sqrt{p+a}}) = p, \tag{3.1}$$

or

$$B(N, Nx) = \left(\frac{R}{Nx}\right)^2 - a. \tag{3.2}$$

For $N = 100$, $a = .01$ and $\frac{1}{2} \leq R \leq 32$ the solution to 3.1 is given in the Figures 3 and 4. In these figures, the x-axis is the "Inherent Load" $\frac{R}{\sqrt{a}}$, i.e. the load the link would see if the link had infinite capacity and therefore drop probability $p = 0$. Since $a = .01$, the Inherent Load equals $10 \times R$.

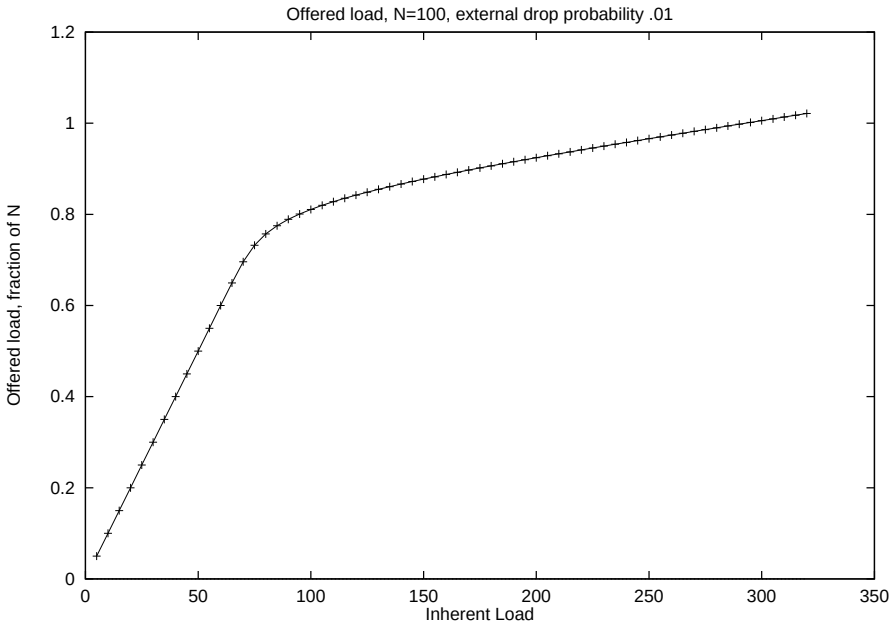

Figure 3: Offered Load (fraction of N), without external loss

We see that as long as the Inherent Load is less than about $.75 \times N$ (less than about .75 of the sum of the bandwidths of the N wavelengths) the actual offered load the bundle

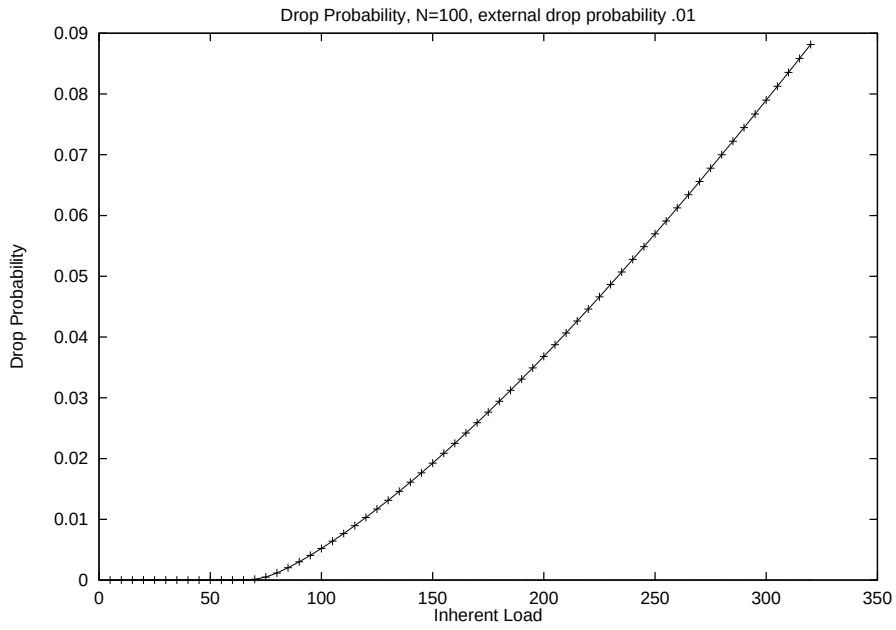

Figure 4: Drop probability, without external loss

sees practically equals the Inherent Load. When the Inherent Load gets larger than $.8 \times N$ the loss probability p for the bundle becomes significant, the congestion windows of the TCP sources decrease, and the actual offered load still increases, but increases slower than the Inherent Load.

It is interesting to note that the actual offered load can increase above N. When the Inherent Load reaches 295, the actual offered load as predicted by the model reaches $1.00177 \times N$, and the predicted drop probability reaches $p = 0.0767175$, so that the predicted carried load becomes $0.9249 \times N$.

When the Inherent Load reaches 320 these numbers become $1.02135 \times N$, $p = 0.0881629$, predicted carried equals $.9313 \times N$.

The solution techniques for 2.4 presented later in this paper will be found to require a condition which for large values of N is even stronger than 3.4, namely (roughly)

$$0 < R < \left(\frac{2}{\pi}\right)^{\frac{1}{4}} N^{\frac{3}{4}}. \tag{3.3}$$

(This needs some more investigation).

Equation 2.1 uses the "square root formula" for TCP. This is valid only for p moderately small. For $p > .1$ the square root formula becomes a questionable approximation. This means that we must restrict our attention to R values (roughly) in the range

$$0 < R \leq \frac{N}{3}. \tag{3.4}$$

In the next sections we will show how to find, for N large, an approximate solution to this equation. The p thus found must be small in order to justify use of the the Square Root Formula for TCP.

4 Approximating the fixed point equation (2.6)

Let Φ denote the cumulative normal distribution:

$$\Phi(x) = \int_{-\infty}^{x} \frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}} dy. \tag{4.1}$$

By the normal approximation to the Poisson distribution we can approximate the Erlang B formula by

$$B(N, y) = \frac{\frac{y^N}{N!}}{\sum_{j=0}^{N} \frac{y^j}{j!}} = \frac{\frac{y^N}{N!} e^{-y}}{\sum_{j=0}^{N} \frac{y^j}{j!} e^{-y}} \sim \frac{\frac{y^N}{N!} e^{-y}}{\Phi(\frac{N+\frac{1}{2}-y}{\sqrt{y}})}. \tag{4.2}$$

This approximation needs y to be large and (either $N \geq y$ or at least $y - N$ not large compared with $\sqrt{y}$). By a second approximation, Stirling's formula:

$$N! \sim \sqrt{2\pi} N^{N+\frac{1}{2}} e^{-N} e^{\frac{1}{12N}}, \tag{4.3}$$

we get a second approximation for the Erlang B formula:

$$B(N,y) = \frac{\left(\frac{y}{N} e^{1-\frac{y}{N}}\right)^N}{\sqrt{2\pi N}\Phi(\frac{N+\frac{1}{2}-y}{\sqrt{y}})} e^{-\frac{1}{12N}}. \tag{4.4}$$

In 4.4 we use the "refined" version of Stirling's formula, which includes the factor $e^{\frac{1}{12N}}$. Later we will see that this refinement is overkill.

By denoting

$$x = \frac{R}{N\sqrt{p}},\ p = \left(\frac{R}{Nx}\right)^2, \tag{4.5}$$

we reformulate the fixed point equation 2.4 (approximately) as

$$x^2 \left(xe^{1-x}\right)^N = \frac{R^2\sqrt{2\pi}}{N\sqrt{N}} \Phi(\frac{N(1-x)+\frac{1}{2}}{\sqrt{Nx}}) e^{\frac{1}{12N}}, \tag{4.6}$$

or

$$xe^{\frac{N}{N+2}(1-x)} = \left(\frac{R^2\sqrt{2\pi}}{N\sqrt{N}} \Phi(\frac{N(1-x)+\frac{1}{2}}{\sqrt{Nx}}) e^{\frac{1}{12N}}\right)^{\frac{1}{N+2}}. \tag{4.7}$$

x, and thereby $p = \left(\frac{R}{Nx}\right)^2$, must be solved from 4.7. This will be done in the next section. The iterative scheme in the next section is based on the observation that in all likelyhood $0 < x < 1$ and therefore $\frac{1}{2} < \Phi(\frac{N(1-x)+\frac{1}{2}}{\sqrt{Nx}}) < 1$, and thus

$$\left(\frac{R^2\sqrt{\pi}}{N\sqrt{2N}} e^{\frac{1}{12N}}\right)^{\frac{1}{N+2}} < xe^{\frac{N}{N+2}(1-x)} < \left(\frac{R^2\sqrt{2\pi}}{N\sqrt{N}} e^{\frac{1}{12N}}\right)^{\frac{1}{N+2}}, \tag{4.8}$$

etc. This gives new, usually improved, lower and upper bounds for x, etc.

5 Solving the fixed point equation (4.7)

Consider the function

$$f_r(x) = xe^{r(1-x)}, \tag{5.1}$$

where $0 < r \leq 1$. Clearly, on $0 \leq x \leq \frac{1}{r}$, f_r is a non-negative, increasing, concave function. For $0 \leq A \leq \frac{e^{-(1-r)}}{r}$ the equation $f_r(x) = A$ therefore can be solved numerically by the iteration (Newton scheme)

$$\begin{aligned} x_0 &= 0, \\ x_{k+1} &= x_k + \frac{A - x_k e^{r(1-x_k)}}{(1 - rx_k)e^{r(1-x_k)}}. \end{aligned} \tag{5.2}$$

This scheme 5.2 is garanteed to quickly converge to a solution. The sequence (x_k) is increasing. This we call the inner loop of the algorithm to solve 4.7. When solving 4.7 we have of course $r = \frac{N}{N+2}$.

The "zeroth" time the inner loop is used we use $A = A_0 = \left(\frac{3R^2\sqrt{\pi}}{2N\sqrt{2N}} e^{\frac{1}{12N}}\right)^{\frac{1}{N+2}}$, in the "belief" that for the correct x $\Phi(\frac{N(1-x)+\frac{1}{2}}{\sqrt{Nx}})$ will not to be too far from $\frac{3}{4}$. Using the inner loop, A_0 leads to x_0.

When, using A_k, the inner loop converges for the k-th time, the result is denoted by x_k. A_{k+1} then is computed from

$$A_{k+1} = \left(\frac{R^2\sqrt{2\pi}}{N\sqrt{N}} \Phi(\frac{N(1-x_k)+\frac{1}{2}}{\sqrt{Nx_k}}) e^{\frac{1}{12N}}\right)^{\frac{1}{N+2}}. \tag{5.3}$$

There is no mathematical guarantee (yet!) that the resulting sequence $((A_0, x_0), (A_1, x_1), \cdots, (A_k, x_k), \cdots)$ indeed converges, but if it converges it clearly converges to a solution (A, x). It is likely that the monotonicity argument at the end of

Section 4 can be used to prove convergence (on $0 < x < 1$, $f_r(x)$ is strictly increasing in x, and $\Phi(\frac{N(1-x)+\frac{1}{2}}{\sqrt{Nx}})$ is strictly decreasing in x).

Once x has been computed, the predicted loss probability is given by $p = \left(\frac{R}{Nx}\right)^2$, and the offered load (in number of wavelengths) is given by $\frac{R}{\sqrt{p}} = xN$.

$\Phi(y)$ is approximated as in (Abramowitz and Stegum p 932, item 26.2.17).

We see that A_{k+1} depends on x_k only through $\Phi(\frac{N(1-x)+\frac{1}{2}}{\sqrt{Nx}})$, or, more accurately, through $\left(\Phi(\frac{N(1-x)+\frac{1}{2}}{\sqrt{Nx}})\right)^{\frac{1}{N+2}}$. Since we expect that always $0 < x_k < 1$ and hence $\frac{1}{2} < \Phi(\frac{N(1-x)+\frac{1}{2}}{\sqrt{Nx}}) < 1$, it is likely that $|x_{k+1} - x_k|$ will be (much?) smaller than $|x_k - x_{k-1}|$, and that the scheme above converges.

6 The fixed point equation (3.2)

With $x = \frac{R}{N\sqrt{p+a}}$, $p = \left(\frac{R}{Nx}\right)^2 - a$, we rewrite 3.2 as

$$xe^{\frac{N}{N+2}(1-x)} = \left(\sqrt{2\pi N}\Phi(\frac{N(1-x)+\frac{1}{2}}{\sqrt{Nx}})e^{\frac{1}{12N}}\left(\left(\frac{R}{N}\right)^2 - ax^2\right)\right)^{\frac{1}{N+2}}. \tag{6.1}$$

(6.1) can be solved in the same way as (4.7) in Section 5, with the difference that now

$$A_{k+1} = \left(\sqrt{2\pi N}\Phi(\frac{N(1-x_k)+\frac{1}{2}}{\sqrt{Nx_k}})e^{\frac{1}{12N}}\left(\left(\frac{R}{N}\right)^2 - a{x_k}^2\right)\right)^{\frac{1}{N+2}}, \tag{6.2}$$

and $A_0 = \left(\frac{3R^2\sqrt{\pi}}{2N\sqrt{2N}}e^{\frac{1}{12N}}\right)^{\frac{1}{N+2}}$. This choice is based on the assumption that $0 \leq a << \left(\frac{R}{N}\right)^2$.

A The Ornstein – Uhlenbeck Approximation

Now appendices.

References

[1] Mathis, M., Semke, J. Mahdavi, J. and Ott, T.J. (1997) The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm. *Computer Communications Review 27* (3), pp 67 - 82 (July 1997).

[2] Ott, T.J., Kemperman, J.H.B., and Mathis, M. (1996) The Stationary Behavior of Idealized TCP Congestion Behavior.
ftp://ftp.bellcore.com/pub/tjo/TCPwindow.ps .

[3] Ott, T.J., Lakshman, T.V. and Wong, L.H. (1999) SRED: Stabilized RED. *Proceedings of IEEE INFOCOM'99*, pp 1346 - 1355, March 1999.