Some remarks on Optical Switching

Teunis J. Ott

July 20, 2000

Abstract

Abstract goes here.

1 Introduction

In this note I am collecting some observations on Optical Switching in a WDM (Wavelength Division Multiplexed) Optical network. These observations grew out of a number of conversations with George Clapp, and a meeting with Ann von Lehmen, Stu Wagner, George Clapp, GK Chang, and Stu Personick on 04/28/00.

In the problem area at hand an optical fiber contains a large number of "channels". Each channel is a single wavelength. All channels have exactly the same bitrate. The number of channels (wavelengths) per fiber can be well over 100. The channels carry IP packets. In nodes in the network there can be E–O and O–E conversion (Electronic to Optical and Optical to Electronic). In nodes in the network packets also can move from one channel in one fiber into another channel in another fiber. This can be done by O–E followed by E–O, but it can also be done without conversion (purely optical switching). In case of purely optical switching it is possible to move from one wavelength to another.

(Name here).

All–Optical connections in the node can be by nailed down connections (Optical Cross Connect) or by statistical multiplexing. In the first case all packets from one channel automatically move all into the same dedicated (output) channel.

In case of statistical multiplexing we need to ty a (preferably large) number of channels into a "bundle". The channels (wavelengths) in the bundle can all reside in one fiber, or can be distributed over a number of fibers. All channels in the bundle share a common origin and destination. At the origin there is contention between packets wanting to use the bundle. Let the number of channels in the bundle be N. When a new packet arrives at the origin of the bundle, if at that time fewer than N packets are in progress of being pumped into the bundle, the new packet starts being pumped into the bundle (into one of the channels of the bundle). Since all channels have exactly the same bandwidth we do cut-through routing: there is no storage in the node. The packet comes out of the "input port" exactly as fast as it goes into the "output port". The input-channel and output channel not having the same wavelength is no problem.

In case of statistical multiplexing there is the possibility that when the header of a packet starts arriving, there is no free channel in the output link. In that case the packet is discarded (lost). (In more sophisticated systems there may be the possibility of saving such packets, see ...). We need to estimate the probability of a packet being lost.

In order to estimate this probability we need to build a model. The Erlang–B model will be presented in Section 2. That section also contains results on the largest possible load such a multiplexer is allowed to receive, given the number of channels and the allowed packet drop probability.

In case of O–E conversion, the electronic images can be stored in a buffer. This allows a buffer (an electronic buffer) in front of the bundle of channels. When a channel (wavelength) in the bundle frees up, the next packet (if any) is converted to O again and pumped into the channel. Since both the number of channels in the bundle, and the number of packets that can be buffered, are finite, loss can still occur.

Still, O–E–O conversion allows higher utilization of the bundle of channels. How much higher depends on the delays allowed in the buffer, and on the permitted loss probability. A model must be built. The Erlang–C model will be presented in Section 3. In Section 4 we compare results from Sections 2 and 3, and show that as long as the number of wavelengths in the bundle is at least 50, use of O–E–O conversion and electronic buffering increases allowed bandwidth utilization by a factor 25 % to 75 %, depending on the permitted packet drop probability and the number of wavelengths in the bundle. This improvement decreases as the number of wavelengths in the bundle increases, and as the requirement of a low drop probability is relaxed.

In case the next link on the normal route of a packet is not available, we could attempt to re–route the packet without using O–E conversion.

(Give the name of such schemes. Hot potato routing?)

(This idea was developed together with George Clapp).

This is somewhat analogous with DNHR and similar schemes in circuit switched telephony. A difference is that in optical packet switching there is no time to obtain information on the state of other links in the overflow route. There is danger of doing more damage than good, by causing loss of other packets on the overflow route. ("Hysteresis" etc.).

Still, it is possible to somewhat safely do dynamic re-routing.

One way of doing this is the following:

In the shimheader (other name? ask GK) of every packet, create space for a"kick–me" bit or "kick–me" counter. When a packet is re–routed over a route different from the standard route to its destination (or to its egress–router), its "kick–me" bit is set. Once the kick–me bit is set, a packet can be allowed into a bundle of channels only if sufficiently many many channels in the bundle are free. This "sufficiently many" number of channels is chosen such, that the probability that any blocking will occur in the bundle during the time that the extra packet is being pumped into the channel is negligibly small.

A formula to estimate this probability, as function of the total number of channels in the bundle and the offered load in the bundle, and (last but not least!) the number of currently occupied channels in the bundle is given in Appendix C.

Thus, once the "kick–me" bit of a packet is set, it is routed only through relatively underutilized bundles.

In a variation on this theme, we could have a "kick–me" counter. Every time the packet is re–routed its kick–me counter in increased. If the counter reaches a critical level (say 4), the packet is discarded instead of routed. We could have a two-bit counter: levels 0, 1, 2, 3 exist in the network. If a packet of level 3 must be re-routed, it is instead discarded. Even at levels 1 etc it is allowed only into relatively underutilized bundles.

@@@ I (tjo) am not very optimistic about this scheme. I doubt the yield in performance is worth the complexity. But let's have a look. @@@

2 The Erlang–B Model

In this section we describe the Erlang–B model for the purely optical statistical multiplexer.

We assume that packets arrive according to a Poisson Process with intensity λ. The acceptability of this assumption for the analysis at hand will be diascussed at the end of Section 3. The conclusion is that while the assumption is wrong, for the analysis at hand it probably leads to reasonably accurate predictions.

Packets are not (necessarily) all the same size. We assume they have sizes that are stochastically independent and all have the same distribution. Let T be the average amount of time it takes to pump such a packet into a channnel. Let N be the number of "servers" (the number of wavelengths). We define:

$$R = \lambda T, \rho = \frac{R}{N}. \tag{2.1}$$

R is the average workload or traffic intensity, and with a slight abuse of notation we call ρ the average utilization. In a sensibly engineered system, $\rho < 1$, and if no packets were ever lost ρ would be the average utilization of the channels in the link.

In this model, the probability that at some point in time exactly k packets are in the process of being pumped into the link (thus using k of the N channels) equals

$$P\{k \text{ busy}\} = \frac{\frac{R^k}{k!}}{\sum_{j=0}^{N} \frac{R^j}{j!}}. \tag{2.2}$$

Thus, the probability that a packet gets lost is the probability that upon its start all channels are busy. This is given by the famous Erlang–B formula $B(N, R)$: The probability that a packets gets lost is

$$P\{N \text{ busy}\} = B(N, R) = \frac{\frac{R^N}{N!}}{\sum_{j=0}^{N} \frac{R^j}{j!}}. \tag{2.3}$$

We define

$$D(N, R) = \sum_{k=0}^{N} \frac{R^k}{k!} e^{-R}, \tag{2.4}$$

and

$$\Phi(x) = \int_{-\infty}^{x} \frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}} dy, \tag{2.5}$$

i.e. $\Phi(.)$ is the distribution function of the normal distribution. For R large and $N \geq R$ (or least *not* $(N < R$ *and* $R - N >> \sqrt{R})$ we have

$$D(N, R) \sim \Phi\left(\frac{N + \frac{1}{2} - R}{\sqrt{R}}\right), \tag{2.6}$$

(The Central Limit Theorem in the special case of the Poisson Distribution). Hence under the same conditions also

$$B(N, R) = \frac{\frac{R^N}{N!}e^{-R}}{\sum_{j=0}^{N} \frac{R^j}{j!}e^{-R}} \sim \frac{\frac{R^N}{N!}e^{-R}}{\Phi\left(\frac{N+\frac{1}{2}-R}{\sqrt{R}}\right)}. \tag{2.7}$$

If $N > R$ *and* $N - R >> \sqrt{R}$ a practically equally good approximation is

$$B(N, R) \sim \frac{R^N}{N!}e^{-R}. \tag{2.8}$$

In the appendix I have collected some more relevant observations on the Erlang–B function.

The value of (2.3) is at least as good as the assumptions made. The main issue is whether packet arrivals form a Poisson Process. This assumptions is almost never good. However, in the case that the actual stream of packets consists of very many flows, each of which is (almost) negligible in intensity compared with the whole stream, the assumption seems quite acceptable: Over time periods short compared with the time between two packets of the same flow, the process is likely to be “quite Poisson”.

(2.3) allows us to estimate the maximal utilization at which a bundle can be driven before the packet loss rate becomes unacceptable. Suppose we are willing to accept a packet loss probability γ. (For example, $\gamma = 10^{-6}$ or $\gamma = 10^{-4}$). We need to find that $R = R(N, \gamma)$ for which $B(N, R) = \gamma$. That value of R is the highest acceptable offered load, and the corresponding value $\rho = \frac{R}{N}$ is the highest acceptable "utilization".

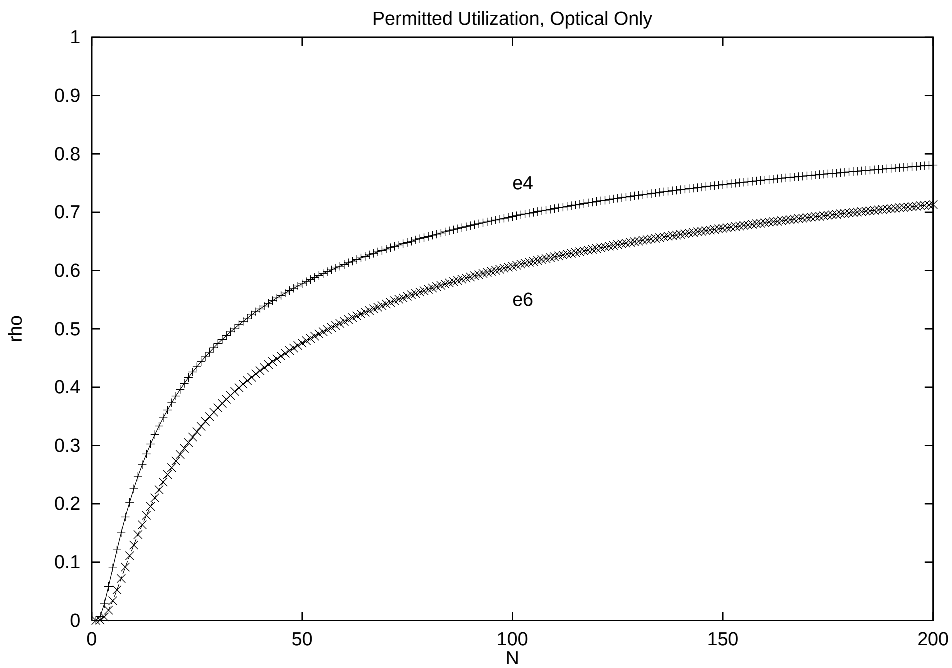

Figure 1: Maximal allowed ρ as function of N, Erlang B (Q = 0)

Finding $R = R(N, \gamma)$ can be done using a standard search meachnism, e.g. bisecting search, or Newton's method. The appendix contains some approximate methods to reasonably quickly get an approximate solution. The approximate methods use a first approximation based on (2.8) and then a refinement based on (2.7). Because the Central Limit Theorem for the Poisson distribution is used the approximation must be quite good for $N \geq R \geq 30$, and probably are acceptable over a much larger range.

Figure 1 contains results of this approximate method. For $\gamma = 10^{-4}$ and $\gamma = 10^{-6}$ it gives the (approximate) maximal allowed $\rho = \frac{R}{N}$ as function of N, for $1 \leq N \leq 200$. The statement that "$Q = 0$" is another way of saying that we have the Erlang–B model:

arriving packets that do not find an idle server are not allowed to wait (waiting space has size zero) and are discarded.

3 The Erlang–C model

In case of optical to electronic conversion we can add a buffer where packets can queue up. Let N be the number of channels (wavelengths), and let Q be the number of packets that can be waiting in the buffer. For convenience we assume that this number Q is to be determined, and that the number of packets that can be buffered then is at most Q, independent of the packet sizes. This assumption makes the analysis quite approximate, but also more manageable.

We again assume that packets arrive according to a Poisson process with intensity λ. The reasonableness of this assumption will be discussed at the end of this section. We again assume that the average packet size is T (expressed in amount of time it takes to play the packet into a channel). In this Section we need to assume that the packet sizes have an exponential distribution. This assumption is false, but probably less worrysome than the assumption that the number of packets that can be buffered is independent of their sizes. We define $R = \lambda T$ and $\rho = \frac{R}{N}$. We need to compute the normalizing constant

$$G = \sum_{k=0}^{N} \frac{R^k}{k!} + \sum_{k=1}^{Q} \frac{R^{N+k}}{N!\, N^k} = \sum_{k=0}^{N} \frac{R^k}{k!} + \frac{R^N}{N!} \frac{\rho(1-\rho^Q)}{1-\rho}. \tag{3.1}$$

In the model, as long as there is at least one packet in the buffer, all channels must be busy receiving a packet. Only when all channels are busy receiving a packet can there be packets buffered.

For $0 \le k \le N$ the probability that no packet is buffered while exactly k channels are receiving a packet equals

$$P\{k \text{ busy, none buffered}\} = \frac{R^k}{k!\ G}. \tag{3.2}$$

For $0 \leq k \leq Q$ the probability that all N channels are busy receiving a packet and in addition exactly k packets are buffered is

$$P\{N \text{ busy}, k \text{ buffered}\} = \frac{R^{N+k}}{N!\ N^k\ G}. \tag{3.3}$$

The probability that a packet gets lost therefore is

$$C(N,R) = P\{\text{loss}\} = \frac{R^{N+Q}}{N!\ N^Q\ G}. \tag{3.4}$$

This is the Erlang–C loss function. Unlike in case of the Erlang–B formula, the derivation of the Erlang–C formula needs the assumption of exponential packet sizes.

Unlike in the Erlang–B model, in the Erlang–C model customers (packets) can wait in the buffer. When a packet arrives and finds $N-1$ or fewer packets in the system in front of it, (each occupying one channel), service starts immediately. If the arriving packet finds $N+Q$ packets in the system in front of it (N being served, Q waiting), the packet is discarded. However, if the packet finds $N+k$ packets in front it, with $0 \leq k \leq Q-1$, the packet joins the waiting room. In that case, its expected waiting time (until service starts) is $\frac{(k+1)T}{N}$. This is the conditional expected waiting time, conditional on k. The unconditional expected waiting time of a packet therefore is

$$\sum_{k=0}^{Q-1} \frac{r^{N+k}}{N!\ N^k\ G} \frac{(k+1)T}{N}. \tag{3.5}$$

Of more interest is the conditional expected waiting time conditional on the waiting time being positive. ("packets that wait, wait in average ...").

An interesting choice of Q would be $Q = N$. In this situation the worst case conditional expected waiting time would be T: namely, if a packet arrives and finds N packets "in service" and $N - 1$ more waiting in the buffer. In this case the worst-case conditional expected waiting time is equal to the expected service time. If we choose this value and at the same time make sure that γ is small, then the average waiting time of packets that must wait is considerable smaller than T.

The assumption of exponentiallity is critical in the derivation of the conditional expected waiting time.

Given the number of channels N and a highest allowed loss probability γ we can again compute the highest allowed load R.

The procedure now is first to determine the highest allowed waiting delay, and the highest allowed drop probability γ. The highest allowed waiting delay determine Q, by the formula "highest delay $= \frac{QT}{N}$ ". Then we find a traffic intensity R that with the chosen Q generates drop probability γ.

Techniques similar to those in Appendix A can be used. Details are given in Appendix B. Also this approximate method relies on the Central Limit Theorem for the Poisson Distribution. It should give almost exact results for $N \geq R \geq 30$

Figure 2 gives the maximal allowed ρ for the Erlang–C model with "$Q = N$", i.e. as many waiting positions as there are servers (wavelengths). It gives this maximal allowed load for $\gamma = 10^{-6}$ and for $\gamma = 10^{-4}$.

It is easy to generate more such Figures for other choices of Q, e.g. $Q = mN$. In that case the worst possibly waiting time (before service starts) is roughly m times a typical service time.

The assumption that the arrival process is Poisson is of course untenable. This is true, among other reasons, because packets are part of larger structures (messages, files).

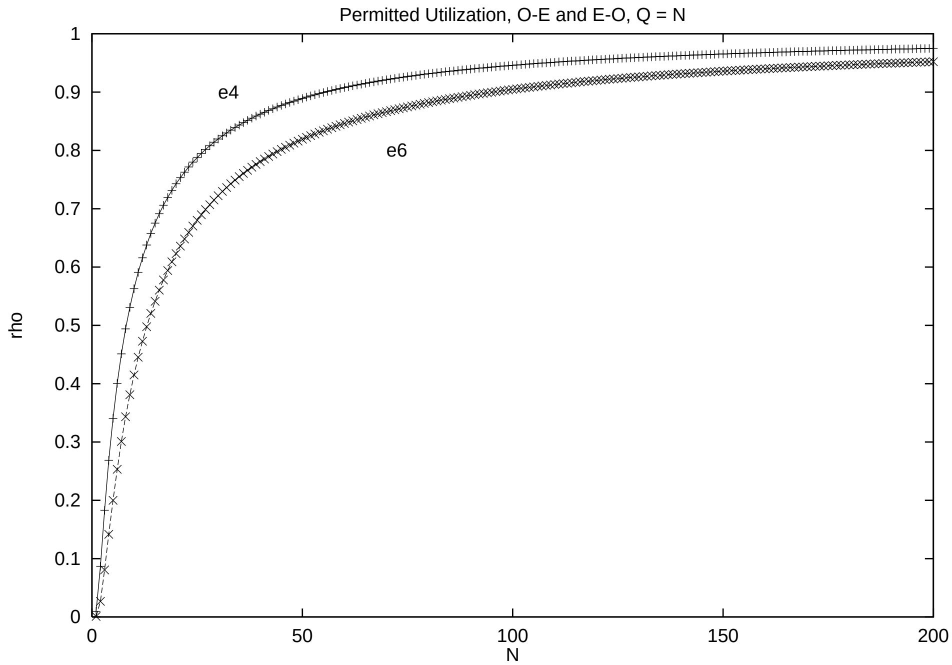

Figure 2: Maximal allowed ρ as function of N, Erlang–C ($Q = N$)

A more important reason the Poisson assumption is untenable is the TCP end–to–end feedback, which is strongly "self-clocked" and makes the traffic much smoother than Poisson. The two factors work in opposite directions, with the TCP feedback probably the stronger factor.

As long as there is no buffering ($Q = 0$, the case of Erlang–B) the Poisson assumption probably does not introduce a major error in the result. Even if there is buffering ($Q > 0$, the case of Erlang–C), as long as $\frac{QT}{N}$ is small, the analysis above probably yields reasonably good predictions.

When $\frac{QT}{N}$ becomes a non-trivial fraction of the round–trip time of a typical packet, the analysis above becomes suspect. In particular, when the average waiting time of a packet becomes a non–trivial fraction of its round–trip–time, the actual system will probably perform much better than the analysis predicts.

On the other hand, we see that even with $Q = N$, for larger values of N, it already is possible to achieve utilizations of over 90%. When Q increases to the point where the Poisson assumption becomes harmfull, the acceptable utilization still remains below 100%! The error in predicted acceptable utilization therefore is limited.

4 Optical Only versus O–E and E–O

Figures 3 and 4 give the maximal allowed ρ as function of N, comparing the "all optical case" (Erlang–B, no waiting allowed, $Q = 0$) with the case where O–E and E–O conversion is allowed, and thus waiting is allowed. We choose "$Q = N$", i.e. as many waiting positions as there are wavelengths.

Figure 3 does this comparison for a maximal allowed drop probability $\gamma = 10^{-4}$. Figure 4 does it for $\gamma = 10^{-6}$.

In the situation of Figure 3 ($\gamma = 10^{-4}$), for $N = 50$ the maximal allowed ρ values are .58 and .89 for respectively $Q = 0$ (Optical Switching Only) and $Q = N = 50$ (O–E and E–O, with waiting). For $N = 200$ these values are .78 and .98. Thus, we see that for the same drop probability $\gamma = 10^{-4}$ use of O–E and E–O conversion allows the throughput to increase by a factor 1.53 at $N = 50$, and by a factor 1.26 at $N = 200$.

In the situation of Figure 4 the allowed loads for $N = 50$ are .48 and .82, and for $N = 200$ they are are .71 and .95. Thus, we see that for the drop probability $\gamma = 10^{-6}$ use of O–E and E–O conversion allows the throughput to increase by a factor 1.71 at $N = 50$, and by a factor 1.34 at $N = 200$.

The allowed load for the situation with O–E and E–O can be somewhat increased by increasing Q, say to $2N$ or to $3N$. However, the possible increase is limited: the allowed

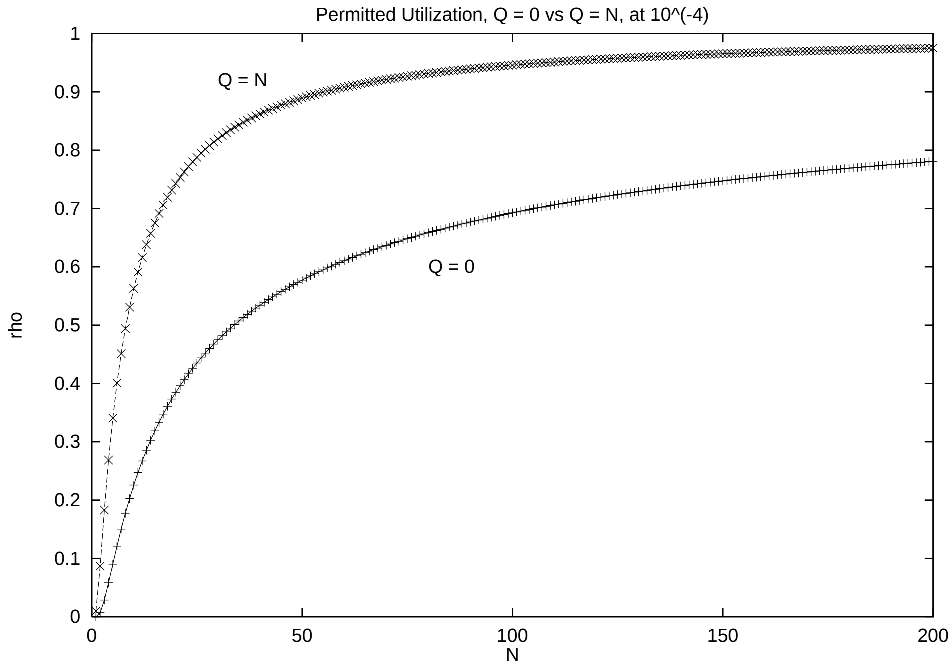

Figure 3: Maximal allowed ρ as function of N, Optical Only versus mixed

load already is fairly close to 100%.

The main disadvantage of allowing O–E and E–O conversion is the cost of the equipment needed to do the conversion. Other disadvantages are the increased operational complexity and and increased delay due to waiting.

The main advantage of allowing O–E and E–O conversion is an increase in the allowed load. i.e. an increase in the possible utilization of the fiber.

Whether the advantage of O–E and E–O converson outweighs the disadvantage thus has become a reasonably straigthforward economic analysis. This analysis needs to be done.

When a fiber already exists and the number of wavelengths can still be increased, the choice is likely to be between increasing the number of wavelengths N, and allowing O–E–

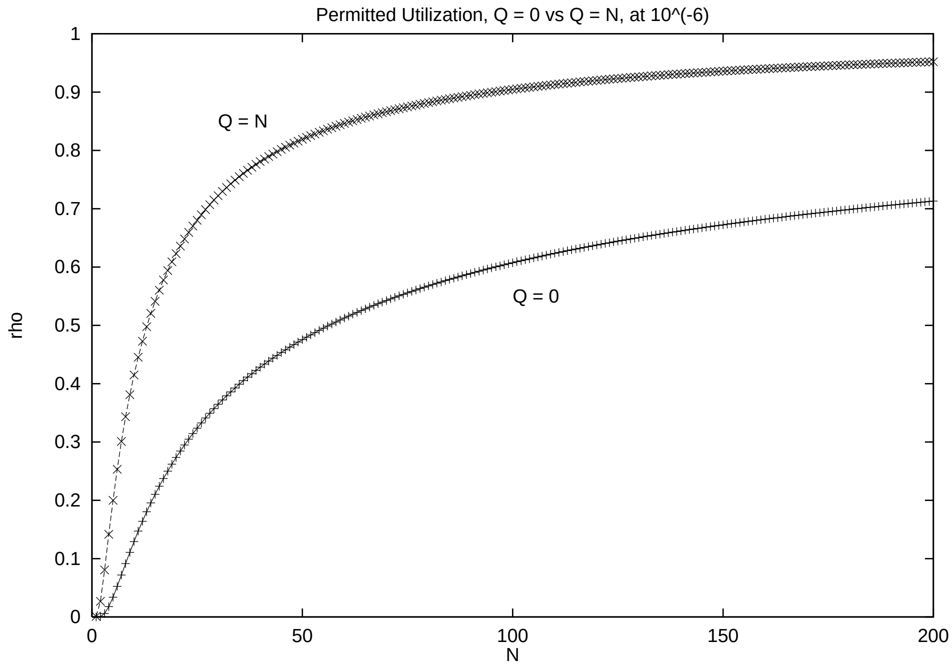

Figure 4: Maximal allowed ρ as function of N, Optical Only versus mixed

O conversion. Once the number of wavelengths N has reached its upper limit, the choice is one between increasing the number of strands of fiber and allowing O–E–O conversion.

5 The TCP Paradigm with general increases and decreases

Here a section.

6 Outside the TCP Paradigm

etc.

A Remarks on the Erlang–B formula

In this appendix we collect some information on the Erlang–B formula:

$$P\{N \text{ busy}\} = B(N,R) = \frac{\frac{R^N}{N!}e^{-R}}{\sum_{j=0}^{N}\frac{R^j}{j!}e^{-R}}. \tag{A.1}$$

A first approximation, based on the Central Limit Theorem for the Poisson Distribution, is

$$B(N,R) \sim \frac{\frac{R^N}{N!}e^{-R}}{\Phi\left(\frac{N+\frac{1}{2}-R}{\sqrt{R}}\right)}. \tag{A.2}$$

This approximation is good as long as R is "large" *and* $N \geq R$. Actually, it is good over a wider range: if R is "large"it is a good approximation even if $N < R$, as long as $\frac{R-N}{\sqrt{R}}$ is not too large. If N is large we can can use Stirling's approximation for $N!$, and get:

$$B(N,R) \sim \frac{\left(\frac{R}{N}\exp\{1-\frac{R}{N}\}\right)^N}{\sqrt{2\pi N}\,\Phi\left(\frac{N+\frac{1}{2}-R}{\sqrt{R}}\right)} = \frac{(\rho\exp\{1-\rho\})^N}{\sqrt{2\pi N}\,\Phi\left(\frac{N+\frac{1}{2}-R}{\sqrt{R}}\right)}. \tag{A.3}$$

We expect that $N > R$. If this holds and also $N - R >> \sqrt{R}$ (and also "N not small") we have that

$$\sum_{j=0}^{N}\frac{R^j}{j!} \sim \exp\{R\}, \tag{A.4}$$

or

$$\Phi\left(\frac{N+\frac{1}{2}-R}{\sqrt{R}}\right) \sim 1, \tag{A.5}$$

and thus

$$B(N,R) \sim \frac{\left(\frac{R}{N}\exp\{1-\frac{R}{N}\}\right)^N}{\sqrt{2\pi N}} = \frac{(\rho\exp\{1-\rho\})^N}{\sqrt{2\pi N}}. \tag{A.6}$$

(A.6) is convenient for numerical work. Before we look at numerical issues, consider the function

$$f(x) = x\exp\{1-x\}. \tag{A.7}$$

Clearly, $f'(x) = (1-x)\exp\{1-x\}, f''(x) = -(2-x)\exp\{1-x\}$, and in general

$$f^{(k)}(x) = (-1)^{k-1}(k-x)\exp\{1-x\}. \tag{A.8}$$

Thus, f is non–negative, increasing and concave on $[0,1]$, with $f(1) = 1, f'(1) = 0, f''(1) = -1$. We see that for $\rho < 1$ fixed, $B(N,R)$ goes to zero exponentially fast if $N \uparrow \infty$. Even if $\rho = 1$ $B(N,R)$ still goes to zero if $N \uparrow \infty$, but now it goes to zero quite slowly. Of course, when $N \uparrow \infty$ while R is constant (in stead of ρ), $B(N,R)$ goes to zero much faster, because now ρ decreases while N increases.

Instead of finding $R = R(N,\gamma)$ such, that $B(N,R) = \gamma$ we can compute $x = x(N,\gamma)$ such, that

$$f(x) = \left(\gamma\sqrt{2\pi N}\right)^{\frac{1}{N}}, \tag{A.9}$$

and then let $\rho = x, R = Nx$. Since $f(.)$ is positive, increasing, and concave, we can use a Newton method to solve (A.9) iteratively, with very fast convergence. We see that if

$$\gamma\sqrt{2\pi N} > 1 \tag{A.10}$$

then the (estimated) loss probability is less than γ for all $R \leq N$ (the loss probability estimated by (A.6) is always less than γ). If

$$0 \leq \gamma\sqrt{2\pi N} \leq 1 \tag{A.11}$$

there is a unique solution $x(N,\gamma)$ to

$$(f(x))^N = \gamma\sqrt{2\pi N},\ 0 \leq x \leq 1. \tag{A.12}$$

It is not clear which of (A.9), (A.12) is the more useful for numerical purposes. If the resulting x value is close to 1, (A.12) gives a nice approximation. For x close to one, $f(x) = 1 - \frac{(1-x)^2}{2!} - \frac{2(1-x)^3}{3!} - O((1-x)^4)$, $(f(x))^N \sim \exp\{-\left(\frac{(1-x)^2}{2!} + O((1-x)^3)\right)N\}$, hence

$$(1-x) \sim \frac{1}{\sqrt{N}}\sqrt{\log\frac{1}{\gamma^2} - \log 2\pi N}. \tag{A.13}$$

The last approximation is good only as long as for the resulting x indeed $(1-x)$ is small. If $\log 2\pi N << 2\log\frac{1}{\gamma}$ we have, with the same proviso, the approximation

$$(1-x) \sim \sqrt{\frac{2\log\frac{1}{\gamma}}{N}}. \tag{A.14}$$

For N larger, $(1-x)$ goes to zero slightly faster.

When (A.13) (or (A.14)) gives a value of x not close enough to one, we can numerically solve (A.9) or (A.12). For example, compute $A = \left(\gamma\sqrt{2\pi N}\right)^{\frac{1}{N}}$, choose $x_0 = 0$, and iterate

$$x_{k+1} = x_k + \frac{A - x_k \exp\{1 - x_k\}}{(1 - x_k)\exp\{1 - x_k\}}. \tag{A.15}$$

Since use of (A.13) gave a value of x not too close to one, the iteration does not lead to a problem with division by a very small number.

Thus we know how to solve ρ as function of N and γ from (A.6). However, we want to solve ρ as function of N and γ from (A.3). In order to this we solve ρ from (A.6) with N and γ_1. Once ρ has been obtained we compute the "actual" drop probability γ_2 from (A.3), and then replace γ_1 by $\frac{\gamma_1 \gamma}{\gamma_2}$, and repeat the procedure. In the first iteration we choose $\gamma_1 = \gamma$. After very few iterations we find that γ_2 is extremely close to the target γ, so that we essentially have solved ρ from (A.3).

B Remarks on the Erlang–C formula

As before, we define

$$D(N, R) = \sum_{k=0}^{N} \frac{R^k}{k!} e^{-R}, \tag{B.1}$$

and

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

so that for R large and (for example) $N \geq R$

$$D(N, R) \sim \Phi\left(\frac{N + \frac{1}{2} - R}{\sqrt{R}}\right). \tag{B.3}$$

Using the Central Limit Theorem for the Poisson Distribibition, and Stirlings' approximation for $N!$ we get

$$C(N,R) = \frac{R^{N+Q}}{N!N^Q\left(\sum_{k=0}^{N}\frac{R^k}{k!}+\frac{R^N}{N!}\frac{\rho(1-\rho^Q)}{1-\rho}\right)} \sim$$

$$\frac{\rho^Q\left(\rho e^{1-\rho}\right)^N}{\sqrt{2\pi N}\,\Phi\left(\frac{N+\frac{1}{2}-R}{\sqrt{R}}\right)+\rho\frac{1-\rho}{1-\rho^Q}\left(\rho e^{1-\rho}\right)^N}, \tag{B.4}$$

where $\frac{1-\rho^Q}{1-\rho} = Q$ if $\rho = 1$. If $\rho \neq 1$ (B.4) can be re-written as

$$p_{drop} \sim \frac{(1-\rho)\rho^Q\left(\rho e^{1-\rho}\right)^N}{(1-\rho)\sqrt{2\pi N}\,\Phi\left(\frac{N+\frac{1}{2}-R}{\sqrt{R}}\right)+\rho(1-\rho^Q)\left(\rho e^{1-\rho}\right)^N}. \tag{B.5}$$

For a given Q and γ (γ is the target value for p_{drop} we now can solve ρ from (B.4) or (B.5).

Following the example of Appendix A we instead choose (in a way to be explained next) a γ_1 and solve (by a method to be explained) ρ from

$$\frac{\rho^Q(\rho e^{1-\rho})^N}{\sqrt{2\pi N}} = \gamma_1. \tag{B.6}$$

For the resulting value of ρ we use (B.4) or (B.5) to compute γ_2, the "actual" p_{drop} for that value of ρ. Next, we solve (B.6) again, but now with γ_1 replaced by $\frac{\gamma\gamma_1}{\gamma_2}$. This procedure is repeated until γ_2 is very close to the true target γ. In the first iteration we use of course $\gamma_1 = \gamma$. We found that we never needed more than 4 iterations to get γ_2 within 2 decimals of γ.

It still remains to show how to solve ρ from (B.6).

Define

$$r = \frac{N}{N+Q}, \; g(x) = xe^{r(1-x)}. \tag{B.7}$$

Then (B.6) reduces to solving x from

$$g(x) = \left(\gamma_1 \sqrt{2\pi N}\right)^{\frac{1}{N+Q}} = A. \tag{B.8}$$

Since $0 < r < 1$ and g is increasing and concave for $x < \frac{1}{r}$, with $g(0) = 0$, $g(1) = 1$, we know that as long as $0 \leq A \leq 1$ (B.8) can be solved by the scheme

$$x_0 = 0, \; x_{k+1} = x_k + \frac{A - g(x_k)}{g'(x_k)}. \tag{B.9}$$

We see that at the end the algorithms for the Erlang–B and Erlang–C inverse computations are essentially the same.

For small values of N ($N < 30$? does this need investigation?) It may be worth comparing the approximate results with those from exact computations.

C Trunk Reservation and the kick–me bit

Suppose we have a bundle of N channels, with a Poisson Process (intensity λ) of arriving packets, and mean packet size T (service time on a channel). Let $R = \lambda T$. Let the service times of the packets be independent, identically distributed random variables that have the exponential distribution with expected value T.

Suppose at some point exactly $k < N$ channels are busy, and a new packet arrives. We can let the new packet into the system (so that the new number of occupied channels becomes $k+1$). What is the probability that a packet blocking will occur during the time the new packet occupies a channel?

This probability of course depends on the time the packet will occupy the channel. If we do not know that time, and assume it has an exponential distribution with expected value T, the probability that the packet causes a future loss equals

$$\frac{B(N,R)}{B(k,R)} \tag{C.1}$$

In case of circuit switched telephony we used this rule. That made sense: the call was routed before we knew the holding time. In case of packet routing we know the size of the packet at hand.

It probably is possible to do similar analysis as function of the packet size. That looks like a "do–able" problem.

It is possible to give simple approximations to (C.1). Define

$$D(N,R) = \sum_{k=0}^{N} \frac{R^k}{k!} e^{-R}, \tag{C.2}$$

so that for $N \geq R$ and $R \uparrow \infty$

$$D(N,R) \sim \Phi\left(\frac{N-R}{\sqrt{R}}\right) = \int_{-\infty}^{\frac{N-R}{\sqrt{R}}} \frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}}\, dy. \tag{C.3}$$

@@@ (Actually, (C.3) still holds if $N < R$ as long as $R - N$ is not large compared with $\sqrt{R}$. Maybe even a bit more general than that, I do not recall the details. tjo) @@@

If $R \leq k \leq N$ and $R \uparrow \infty$ we therefore have @@@ (please verify, I am quoting from memory, tjo) @@@

$$\frac{B(N,R)}{B(k,R)} \sim \left(\frac{R}{N}\right)^{N-k} \frac{\Phi\left(\frac{k-R}{\sqrt{R}}\right)}{\Phi\left(\frac{N-R}{\sqrt{R}}\right)} \sqrt{\frac{k}{N}} \left(\frac{k}{N}\right)^{k} e^{N-k}. \tag{C.4}$$

If $N - k << k \leq N$ this reduces to

$$\frac{B(N,R)}{B(k,R)} \sim \left(\frac{R}{N}\right)^{N-k} \frac{\Phi\left(\frac{k-R}{\sqrt{R}}\right)}{\Phi\left(\frac{N-R}{\sqrt{R}}\right)}, \tag{C.5}$$

and if $N - k << \sqrt{R} < R \leq k \leq N$ this further reduces to

$$\frac{B(N,R)}{B(k,R)} \sim \left(\frac{R}{N}\right)^{N-k} = \rho^{N-k}. \tag{C.6}$$

It is quite possible that in many situations (C.6) is sufficient for decision making.

@@@ But again: I am not sure it is worth the complication. tjo. @@@

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.
Submitted to Perf. Eval. Rev. This paper has circulated on the Web since August 1996. see ftp://ftp.bellcore.com/pub/tjo/TCPwindow.ps for a copy.

[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.