Evaluation of Layered Optical Networking Technology Alternatives

Ronald G. Addie*, David Fatseas* and Moshe Zukerman†
* University of Southern Queensland, Australia, addie@usq.edu.au, david_fatseas@yahoo.com
† Electronic Engineering Department, City University of Hong Kong, Hong Kong SAR, m.zu@cityu.edu.hk

Abstract

Comparison between alternative technologies for switching and routing is difficult for several reasons: (i) the range of options is quite large and appears to demand knowledge over a very wide range – wider than we can reasonably expect to find in any one individual; (ii) in addition, each of the alternative technical solutions can potentially be used *in combination* with one of more of its competitors; (iii) the impact on the traffic being handled is difficult to model, and therefore it is almost inconceivable that we can predict the performance which can be delivered by complex configurations.

In this paper we propose and develop a unified framework for multiple technologies that interwork together. To do this we adopt a simple model based on the principle that (i) technologies can be modelled as *layers*; (ii) traffic can be modelled as a Poisson stream of *flows*; and (iii) the dynamic (and static) choice between layers can be modelled by a strategy for splitting traffic between layers.

Once this unified framework has been adopted, alternative technologies can be incorporated into in this model by choosing a collection of parameters which determine, for each layer, what it will cost to deliver a certain type of traffic and how traffic will be *split* between layers when there is a choice.

I. Introduction

The currently evolved Internet architecture is based on a variety of technologies and serves numerous services and traffic types. Internet design that will minimize cost subject to meeting quality of service (QoS) requirements of the various services (including protection and survivability) requires simultaneous optimization of (1) technology choices, (2) Topology (although we consider the physical topology of nodes and links as given, the virtual topology still needs to be optimized), and traffic engineering (including traffic grooming). It is clear that optimizing the entire Internet from all these three aspects is too difficult. In the following we list some of reasons for difficulty in optimizing the Internet design:

1) the range of technology options is quite large and appears to demand knowledge over a very wide range – wider than we can reasonably expect to find in any one individual;
2) each of the alternative technical solutions can potentially be used *in combination* with one of more of its competitors;
3) the impact on the traffic being handled is difficult to model, and therefore it is almost inconceivable that we can predict the performance which can be delivered by complex configurations.
4) There are over a billion Internet users generating traffic trough a very large numbers of nodes. The scale of the Internet is so large that global optimization is not feasible.
5) the individual demands are very bursty and unpredictable (flow sizes are known to follow heavy tailed distributions).

There are many papers which propose mathematical programming formulations of network designed (e.g. [1]) however these are applicable only to small networks and only consider a subset of the above problems. This limitation is also applicable to *some* heuristics that have been proposed (e.g. [2], [3]), that are not scalable to large problems. Furthermore, when a large number of technologies are considered simultaneously, formulation of the design problem as a mathematical program is a task of massive complexity, and its solution, if it can be obtained, cannot be intuitively understood.

The book [4] adopts a formulation of the network design problem in terms of a problem of optimal routing through a multi-layered network (or a multi-layered graph). This is also the approach we adopt, although there are a number of significant differences in the two approaches. In particular, our objective is somewhat different since it includes *technology choice* (which technologies should be used), and dimensioning of the physical layer as part of the problem, whereas [4] assumes that the physical layer is given, a specific collection of grooming technologies is available, and the task to be solved is how to route traffic through this network. Also, we adopt a statistical model of traffic model in which flows have randomly distributed size with a Pareto distribution, whereas in [4] a more specific traffic model (a scenario rather than a statistical model) is adopted. Also, [4] takes the constraint that the same wavelength can be used in each fiber once into account whereas we do not. Taking this constraint into account increases the complexity of the problem considerably and given the other challenges, for solving the problem of technology choice it seems preferable to defer this issue form the moment.

A. A Unified Layered Routing Model

In this paper we propose and develop a unified framework for multiple technologies that interwork together under realistic assumptions of traffic demand. To do this we adopt a simple model based on the principle that (i) technologies can be modelled as *layers*; (ii) traffic can be modelled as a Poisson stream of *flows*; and (iii) the dynamic (and static) choice between layers can be modelled by a strategy for splitting traffic between the available paths, which make use of different layers, making use of *flow size* dynamically when choosing between paths.

Once this unified framework has been adopted, alternative technologies can be incorporated into in this model by choosing a collection of parameters which determine, for each layer, what it will cost to deliver a certain type of traffic and how traffic will be *split* between layers when there is a choice.

We assume that the network under study is made of layers, each layer is a network in its own right, and the links of layer $n+1$ correspond to the traffic streams of layer n. We then use shortest-path routing as the design principle for each layer, so the design of each layer is *almost* independent from the design of each other.

However, since the cost of links in layer $n+1$ is determined by the cost of the whole path that these links follow, in layer n, the traffic in layer $n+1$ will change with the costs in layer n, and this will cause the traffic in layer n to change and hence the costs in layer n will change again.

An iterative scheme for designing all the layers appears to converge, in our experiments, quite quickly, although the experiments are at an early stage.

B. Cost of Communication and Flows

We consider three Internet cost and energy consumption causes:

1) Individual Packet Processing: In today's Internet packets are processed (including repeated buffering, and routing-table look-up) individually at each router [5] which is energy inefficient especially for large flows, and fails to respect our carbon emission budget as we approach the "zettabyte era";
2) connection set-up, including recording and updating hash tables in flow-based routing, or the setup cost for an SDH or WDM paths in cases where those are used;
3) transmission.

In line with these three sources of cost, which are clearly incurred by packets to a different degree according to the size of flow and what types of setup costs are incurred, we distinguish three size-based flow types:

1) Mice: these are the smallest flows and there are many of them. It is not efficient to treat them as flows. Their tunnels may be longer than shortest path (similar to busses that drop off and take up passengers along their routes).
2) Elephants: these are the largest flows. Their numbers are relatively small, so they justify complex flow setup and clear-down, possibly including the setup of an LSP or WDM routed wavelength. Individual packet processing (cause 1) is avoided. Their aggregation is achieved by using a pool of wavelength channels per trunk.
3) Kangaroos: they are routed based on the current IP architecture. The existence of Kangaroos is practical and consistent with our aim for evolution. It may be beneficial to use shortest-hop-path routing for them because their packets are treated individually at every router.

These categories may seem somewhat arbitrary, however because the range flow sizes is very wide – Internet flow sizes have a heavy tailed distribution [6] – it seems likely that a finer distinction between different classes of traffic may be unnecessary. At the risk of oversimplification, elephants are the flows for which, even if a flow setup cost is incurred it will be insignificant; the mice are the flows for which any flow setup cost will be more significant than any per-packet or per-byte costs, and the kangoroos are the flows where these different costs are approximately in balance.

II. The Traffic Model

We will assume that all traffic is formed as a collection of flows. We assume that the flow sizes are random, each uncorrelated with the other, with a distribution of the *truncated Pareto* type, as described in Subsection II-A. The arrival times of the flows are assumed to form a Poisson process.

A. The Truncated Pareto Distribution

We assume that traffic is either continuous at a fixed rate, or is made up of a Poisson stream of flows, of rate λ, which have a truncated Pareto distribution with shape parameter γ, with minimum size δ and maximum size D. Assuming that flow sizes are measured in bits, the probability that a randomly selected flow with the truncated Pareto distribution with these parameters is shorter than t is

$$\begin{cases} 0, & t < \delta, \\ \frac{\left(\frac{D}{t}\right)^{-\gamma} - \left(\frac{D}{\delta}\right)^{-\gamma}}{1-\left(\frac{D}{\delta}\right)^{-\gamma}}, & \delta \le t < D, \\ 1, & t \ge D. \end{cases} \tag{1}$$

The mean bit rate of such a Poisson stream of flows is

$$\frac{\lambda\gamma\left(\delta^{1-\gamma}-D^{1-\gamma}\right)}{(\gamma-1)\left(\delta^{-\gamma}-D^{-\gamma}\right)}. \tag{2}$$

III. Fixed Point Analysis of Network Traffic, Performance, and Link Capacities

A very simple but effective way to analyse the way in which traffic will be carried (and the performance it will experience) in networks is to conduct a *fixed point iteration*. The behaviour of traffic is affected by the capacity of the links, switches, and routers that it passes through, and also by the other traffic that uses the same resources at the same time. Given all the conditions which apply, relatively simple models can be used to predict the performance and behaviour of one traffic stream. A fixed-point algorithm can then be used to predict the behaviour and performance of the complete collection of traffic streams which share a network by successively replacing the traffic streams by streams which exhibit their behaviour in the presence of the other traffic.

A. The Erlang Fixed-Point Algorithm

One simple and commonly used approach for approximation of blocking probabilities in overflow loss network models is the so-called *Erlang fixed-point approximation* (EFPA) method. It is based on decoupling the given system into independent server groups (subsystems), each of which is modeled by an M/M/m/m queueing system, for which the blocking probability (or overflow probability) can be computed by the Erlang B formula. EFPA was first proposed in [7] in 1964 for the analysis of circuit-switched networks and has remained a cornerstone of telecommunications networks and systems analysis to this day. See for example, [8]–[13] and references therein for applications of EFPA.

B. Fixed Point Algorithms for Link Capacities

In this paper our objective is not to estimate blocking probabilities but instead to estimate required capacities. Therefore, instead of iterating until convergence of loss probabilities has occurred, we adjust capacities at each iteration, and continue iterating until capacities have converged. Whereas in the EFPA the Erlang formula is used to estimate loss, in this iteration it is used to estimate the capacity required to achieve a certain target for loss on each link.

IV. Analysis of Layered Networks

The basic fixed-point iteration described in the previous section can be extended to layered networks so long as we have a scheme for splitting traffic between layers. Splitting the type of traffic described in Section II is straightforward so long as the only criterion used for deciding which way to direct traffic is flow size.

A. Layered architecture concepts

The concept of layered network architecture needs to be defined somewhat more rigorously than usual before we can employ it to define how traffic will be split between layers. The most important principle of layered network architecture is that *each layer of a network employs services provided by layers below to provide services to the layers above*.

Sitting on top of this layer-cake of services are the demands from the users, which we refer to in this paper as *traffic streams*. Traffic from these user-generated traffic streams is carried by assigning it to *paths* through the layer at the top. When there is a layer below this top layer, it is employed for one and only one reason: to implement links at the top layer.

There are three distinct ways in which one layer (layer n) can implement a link between node A and node B at the higher layer $(n+1)$:

(i) layer n already has a link between A and B and it simply passes on this service, at no extra charge, to the layer above; when a link from A to B already exists, and is not overutilized, this will always the best way to provide the link from A to B at layer $n+1$.

(ii) layer n does not have a link from A to B but it has a path from A to B which it can permanently package as a link, for layer $n+1$.

(iii) layer n does not have a link from A to B but it can dynamically set up, as required, a path through its network from A to B.

There is usually a significant extra cost in making use of links provided by the mechanisms (ii) and (iii). In Case (ii), this cost is due to the fact that modularity requirements in Layer n will mean that the capacity of the path set up through this layer might be considerably more than is actually needed. In Case the cost is due to the fact that every time a Layer $n+1$ is set up, a significant cost is incurred. Although the setup cost is always incurred, in Case (ii), the fact that it is incurred only once, ever, means that we can neglect it. Modularity always restricts the use of paths through Layer n, however in Case (iii) we can assume that the path is fully utilized while the link is in use, so we can neglect this cost in Case (iii). Since all the traffic carried by layer n is derived from one of these three mechanisms, the traffic streams in layer n correspond, in effect, to the links in layer $n+1$.

B. Splitting

Splitting is fairly easy to analyse. Suppose the overall rate of flows is λ with a range of flow sizes from δ to ∞, and suppose that traffic is split on the basis of whether a flow is bigger or smaller than D. If the flow rate of flows less than size D is λ_1, then the traffic of the smaller flows will now have the standard model, but with rate λ_1, and flows in the range δ to D. The other traffic stream will have a rate of $\lambda - \lambda_1$ and a range of flow sizes from D to ∞.

C. Merging

The parameters of a merged stream that we need to choose are: (i) the flow arrival rate; (ii) the shortest flow; and (iii) the longest flow. (We assume that all traffic exhibits a common power law, with coefficient γ)

In order to determine the parameters of a merged traffic stream we adopt the principle that the flow rate of the merged stream should be the sum of the flow rates of the component streams, the maximum flow size should be the maximum of the maximum flow sizes of the component streams, and the mean bit-rate of the merged stream should be the sum of the bit-rates of the component streams.

It follows that we have precisely one free parameter, the minimum flow size, with which to ensure that the mean bit-rate is correctly matched. Since the mean of a Poisson stream of truncated Pareto flows is given by (2), if the mean rate of such a traffic stream is m, the δ required to match this is the solution of

$$m = \frac{\lambda\gamma\left(\delta^{1-\gamma} - D^{1-\gamma}\right)}{(\gamma-1)\left(\delta^{-\gamma} - D^{-\gamma}\right)}.$$

Rearranging this gives:

$$\delta = \frac{m(\gamma-1)\left(1-(D/\delta)^{-\gamma}\right)}{\lambda\gamma(1-(D/\delta)^{1-\gamma})},$$

which can be used repeatedly to solve for δ quite quickly.

V. Concluding Remarks

We have outlined a layered routing concept which can be applied to modern networks which make use of a diverse range of technologies. An algorithm for network design of a layered network was described and this algorithm has been implemented (see [14]–[16]) and some preliminary experiments carried out. The experiments confirm that the network design algorithm converges quickly for very large networks.

References

[1] R. Ramaswami and K. N. Sivarajan, "Design of logical topologies for wavelength-routed optical networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 840–851, Jun. 1996.

[2] K. Zhu and B. Mukherjee, "Traffic grooming in an optical WDM mesh network," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 1, pp. 122–133, Jan. 2002.

[3] M. Sridharan, M. V. Salapaka, and A. K. Somani, "A practical approach to operating survivable WDM networks," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 1, pp. 34–46, Jan. 2002.

[4] K. Zhu, H. Zhu, and B. Mukherjee, *Traffic Grooming in Optical WDM Mesh Networks*, Springer, 2005.

[5] L. G. Roberts, "A radical new router," *IEEE Spectrum*, vol. 46, no. 7, pp. 34–39, July 2009.

[6] M. E. Crovella, M. S. Taqqu, and A. Bestavros, *Heavy-tailed probability distributions in the World Wide Web*, pp. 3–25, A practical guide to heavy tails: statistical techniques and applications. Birkhauser, 1998.

[7] R. B. Cooper and S. Katz, "Analysis of alternate routing networks with account taken of nonrandomness of overflow traffic," Memo., Bell Telephone Lab., 1964.

[8] G. R. Ash and B. D. Huang, "An analytical model for adaptive routing networks," *IEEE Transaction on Communiations*, vol. 41, no. 11, pp. 1748–1759, November 1993.

[9] S. P. Chung, A. Kashper, and K. W. Ross, "Computing approximate blocking probabilities for large loss networks with state-dependent routing," *IEEE/ACM Transactions on Networking*, vol. 1, no. 1, pp. 105–115, 1993.

[10] F. P. Kelly, "Loss networks," *The Annals of Applied Probability*, vol. 1, no. 3, pp. 319–378, August 1991.

[11] D. Mitra, J. A. Morrison, and K. G. Ramakrishnan, "ATM network design and optimization: a multirate loss network framework," *IEEE/ACM Transactions on Networking*, vol. 4, no. 4, pp. 531–543, August 1996.

[12] Z. Rosberg, H. L. Vu, M. Zukerman, and J. White, "Performance analyses of optical burst switched networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 7, pp. 1187–1197, September 2003.

[13] E. W. M. Wong, K. M. Chan, and T. S. Yum, "Analysis of rerouting in circuited-switched networks," *IEEE/ACM Transactions on Networking*, vol. 8, no. 3, pp. 419–427, June 2000.

[14] A. Zareer and R. G. Addie, "Netml version 2.0," 2004, http://cs.sci.usq.edu.au/netml2.

[15] R. G. Addie, S. Braithwaite, and A. Zareer, "Netml: A language and website for collaborative work on networks and their algorithms," in *Proceedings, the Australian Telecommunication Networks and Applications Conference 2006*, December 2006.

[16] R. G. Addie, "Netml v4.0 – an online environment for access and sharing of network data and software for network analysis and design," Tech. Rep. SC-MC-1001, University of Southern Queensland, 2010.