

A new HFC architecture using return path multiplexing

James C. Yee

Com21, Inc.

750 Tasman Drive, Milpitas, CA 95035, U.S.A.

Abstract

A Hybrid Fiber Coaxial (HFC) plant is typically configured in a tree topology and covers a large area with tens of thousands of House Holds Passed (HHP) and several return paths into the headend. During initial deployment, it is usually the case that the number of cable modem subscribers is small compared to the HHP, resulting in a small number of modems spread out over the return paths. To operate more efficiently, the return paths should be combined to reduce the port requirement at the headend.

A simple upstream RF combiner can be used to merge separate return paths, but that would also funnel the noise of the separate paths and degrade performance. Instead, what is needed is an upstream aggregation device that will multiplex the paths without aggregating the noise. To do so, such a device needs to operate with the HFC's MAC (media access control) layer and employ multiplexing discipline that incurs limited impact on performance under varying modem distribution scenarios. In addition, this device must be simple and transparent to the rest of the HFC system.

In this paper, we describe through analysis and simulation how such a return path multiplexing device is possible, and how it impacts the HFC network architecture and upstream performance.

Keywords

Hybrid Fiber Coaxial, Cable Modem, Return Path Multiplexing

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35388-3_42](https://doi.org/10.1007/978-0-387-35388-3_42)

1 INTRODUCTION

Depending on the topology of the HFC plant, the distribution of users, and the node size, the number of return paths into the headend of a HFC plant may vary. For example, a HFC plant with 64 000 House Holds Passed (HHP) will result in 32 separate returns assuming a node size of 2000. Further recombination of these returns in each node using Fabre-Perot or Direct Feed-Back lasers may reduce the return port requirement by a factor of 4, yielding a realistic upstream port requirement of 8. A HFC plant with these components is illustrated in Figure 1.

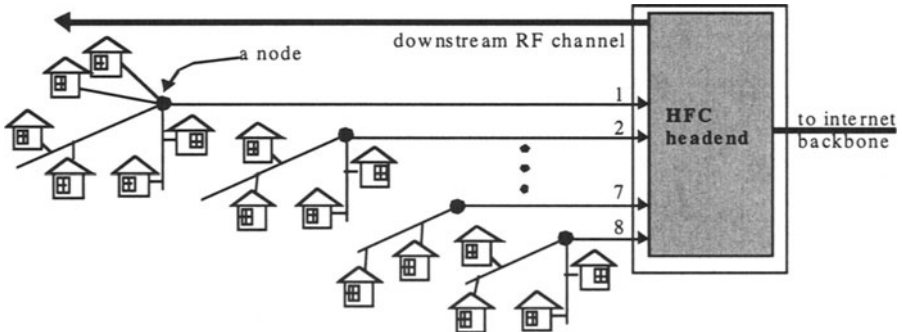


Figure 1: HFC access network with 8 return paths

The number of active subscribers on each trunk is likely to be small in initial deployment, perhaps only 1%. Such typical sparse deployment means each return path only services a few modems. Nevertheless, each return path normally requires a port in the headend. Since the cost of headend equipment increases with the number of ports, this results in an inefficient utilization of headend resources.

To increase efficiency, we need to concentrate the modems into fewer ports. To do so, we can either redesign the HFC plant topology or we can recombine the return trunks. The first solution is too costly and will backfire when more cable modem subscribers join. A simple upstream RF combiner can be used to combine the return paths, but that would also raise the noise floor at the headend ports due to a phenomenon called noise funneling. Noise funneling can be catastrophic to the performance of the HFC access network.

Instead, what is needed is an upstream aggregation device that will recombine the paths without aggregating the noise. To do so, such a device needs to operate with the HFC's MAC (media access control) layer and employ a multiplexing policy that incurs limited impact on performance under varying modem distribution scenarios. The complexity of this device must be low to realize the cost savings. In addition, like a passive RF combiner, this device must be transparent to the rest of the system. In this paper, we call such a device the Return

Multiplexer (RMUX), and illustrated in Figure 2 how it fits into a HFC access network.

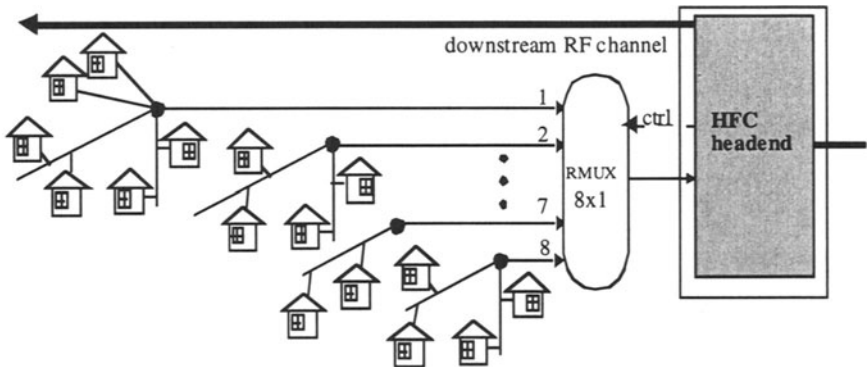


Figure 2: Reduced port requirement using a 8x1 RMUX

As we will describe in the next section, the RMUX introduces a new architecture for HFC systems. There are of course certain trade-offs and limitations in its use, but properly configured, the RMUX can dramatically reduce the headend port requirement with minimal performance degradation. In the remainder of this paper, we examine through simple analysis and detailed simulation how the RMUX performs under various configurations and load conditions.

2 SYSTEM DESCRIPTION

As described in the UPSTREAMS HFC MAC/PHY protocol specification (Laubach, 1997), upstream data in many deployed HFC access network is transported in fixed sized TDMA slots carrying ATM cell payloads, and the allocation of the slots is centrally scheduled by the headend. The RMUX, as a cell-level multiplexer, can therefore be controlled by the headend to open and close the corresponding path during the appropriate slots. This can be done without any loss of data cells. This is illustrated in Figure 3, where the RMUX will serve the incoming data cells in the port sequence of {1,2,4,3}. In this way, the RMUX isolates each return path from the RF noise of other paths.

Typical HFC protocol employs a request-grant mechanism for modems to acquire upstream data slots during preallocated phases of the protocol's operation. During the request phase, a fixed number of slots are made available for modems to send requests upstream. Such requests are sent according to a multiple access protocol with a collision resolution mechanism similar to that

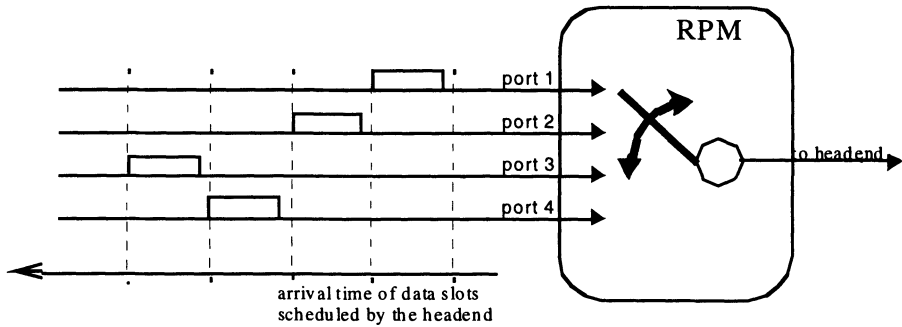


Figure 3: switched forwarding of data in a 4x1 RMUX

of 802.3 Ethernet. However, the headend has no a priori knowledge of when or on which return path the next request will arrive. Consequently, such a cell may arrive at a port not currently listened to by the RMUX. A request cell that is not serviced by the RMUX is lost, and is referred to as a blocked cell.

From the viewpoint of the transmitting cable modem, there is no difference between a blocked request cell and a request cell lost to collision. Therefore, the RMUX can be viewed as an additional contention mechanism during the request phase. To optimize the performance of the RMUX is to minimize the number of blocked cells, and to do so in a fair manner across the input ports of the RMUX.

The manner in which cells are blocked at the RMUX depends on two main factors. One is the service discipline used in servicing the input ports of the RMUX during the request phase. The other is the load distribution across the input ports of the RMUX. We next discuss these two factors.

2.1 RMUX Service Discipline

The RMUX service discipline dictates which input port to service and at which time. The simplest form of service discipline is the round-robin, and there are several variations.

- Round-Robin: by serving each input port in a cyclic manner and serve a constant number of arrivals during each visit, round-robin is a fair policy.
- Grouped Round-Robin: if certain ports can be identified as lightly loaded or have high SNR and can afford to withstand some noise aggregation, groups of input ports can be serviced at the same time, thereby reducing the latency between the sojourn time between visits. Again, a fixed number of slots is served during each visit.

Aside from round-robin, we can also vary the order of service of the input ports.

- Load Dependent Acyclic: the load on each input port can be used to determine how many consecutive slots are serviced each time and in what order the inputs are serviced.

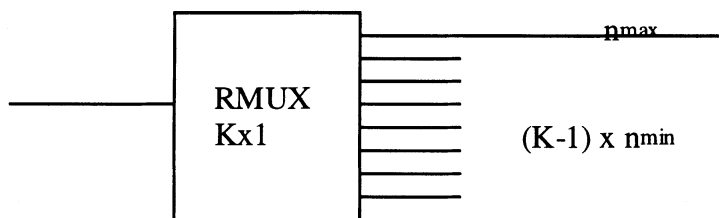
However, such acyclic policies are more complex to implement. We propose some in this paper that should be relatively easy to implement, but no performance results are available for them at this time.

2.2 RMUX Input Load Distribution

Assume the RMUX employs a round-robin policy. With the total number of modems fixed, the worst performance should occur when if all upstream traffic is grouped onto one input port. Reasoning similarly, the best performance will occur under an evenly distributed load across the input ports.

The probability of arrival at the ports can be used to quantify the load distribution across the input ports. However, such characterization is not very useful to the HFC access network architect, since such fine grained information is typically not available. A more practical way to quantify the load distribution is to use the number of active modems on each port. If the HFC system used is capable of providing Quality of Service (QoS) levels delineated by minimum and maximum rates, like that of Com21's system, then the subscribed minimum rates together with the number of modems can together be used to quantify the load on each port. For simplicity, we assume the QoS of all modems, if applicable, to be the same.

Let the number of input ports be K , we define the parameter Index of Symmetry (IOS) as illustrated in Figure 4 to better quantify the symmetry of the input port loading.



N = total # of cable modems

$$= n_{max} + (K-1) \times n_{min}$$

$$\text{IOS (Index of Symmetry)} = n_{min}/n_{max}$$

Figure 4: Illustrate Index of Symmetry

Table 1 Example Index of Symmetry calculations

K	N	n_{min}	n_{max}	IOS
8	128	16	16	1.0
8	128	0	128	0.0
8	128	14	30	0.467
4	64	14	22	0.636

Using this definition:

$$IOS = n_{min}/n_{max} \quad (1)$$

For the type of load distribution shown in Figure 4, we have

$$n_{max} = N/[(K-1)IOS + 1] \quad (2)$$

which can be conveniently used in network planning.

An IOS can be similarly defined for a system with bimodal load. That is, if the ports are partitioned into two group with group 1 having n_{max} cable modems and group 2 having n_{min} cable modems, where $n_{max} \geq n_{min}$, we can define the IOS to be n_{min}/n_{max} . Through simulation, we have found that regardless of how nodes are distributed, the IOS definition in (1) allows us to establish at least approximate regions of operation that gives desirable performance.

2.3 Implications

The above discussion tells us that there are many design and configuration alternatives associated with the RMUX. There are many extensions and variations of service disciplines based on the ones proposed. For example, a load dependent round robin policy can vary the number of slots served depending on the number of modems active on each port. An optimal service discipline that delivers zero blocking under all load distributions may exist, but it will require accurate estimates of traffic pattern of all modems and coordinate closely with the headend scheduler. The complexity of the resulting RMUX will likely be prohibitive. We will demonstrate in the following that a simple RMUX can deliver quite decent performance.

Even when the choice of service discipline and load distribution is not optimized to ensure no blocking, we must keep in mind of the alternatives facing the HFC

access network architect. If a RF combiner is used to aggregate 8 return paths, the signal to noise ratio (SNR) may decrease by as much as 9 dB. Even with the SNR of each return path at a favorable 20 dB or better, the impact on system performance is significant.

3 ANALYSIS

The RMUX can be modeled as a simple polling system where arrivals and service are slotted with slot duration D . Different from a typical polling system, however, the buffer size is zero and the switching time depends on the service time. We denote the n th time the server visits buffer i by $t_i(n)$, and the number of arrivals during this n th visit is $a_i(n)$. The maximum number of slots serviced during that visit is denoted by $s_i(n)$, and the switching time from buffer i to the next buffer after $t_i(n)$ is denoted by $w_i(n)$. The value of $w_i(n)$ is given by

$$w_i(n) = D(s_i(n) - a_i(n)). \quad (3)$$

In Figure 5 is shown such a polling system with K buffers. The load on port i is represented by the arrival process x_i into buffer i . Each arrival process x_i describes the slotted arrival of requests into port i . The process x_i needs to capture the behavior of the aggregated traffic generated by the modems connected to port i , as modulated by the Ethernet like contention resolution algorithm.

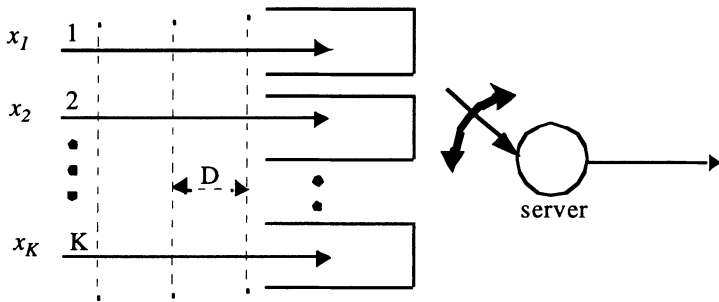


Figure 5: polling system model for the RMUX

There are existing analytical results (Takagi, 1985) which can be extended to describe the behavior for this polling system, but under unrealistic stochastic assumptions. For example, one possible metric for evaluating the performance of the RMUX is the probability that an arrival to buffer i will occur during the interval $[t_i(n), t_i(n) + Ds_i(n)]$, for some n . But such a metric requires a valid traffic model describing the arrival process, which is difficult to obtain.

For now, we only present some first order approximate analysis for the proposed service disciplines to illustrate the general behavior of the service disciplines. The more detailed evaluation of the system is left to simulation.

3.1 Round-Robin Service

We model a RMUX with round-robin service using a polling system with a cyclic server, where after buffer i is visited, the server moves on to buffer $(i+1) \bmod K$. We also require that $s_i(n) = 1$, for all i, n .

This is a system with symmetric load, so the probability that an arriving customer will not be blocked is $1/K$. Fortunately, this does not translate automatically into a scaling of throughput by $1/K$. Let the normalized occupation of upstream slots, or load, of a system without the RMUX be λ , where $\lambda = 1$ implies that every upstream slot is occupied. Assuming traffic is divided evenly among the K ports, then the load on each port i , denoted by L_i , is λ/K . However, this does not mean only λ/K of the slots on each port will be occupied. With less contention on a path with fewer modems, the ratio of slots occupied is scaled up by a factor α , $1 \leq \alpha \leq K/\lambda$, as illustrated in the shaded region of Figure 6 (a). So scaled, the portion of slots occupied per port is $P = \alpha\lambda/K$.

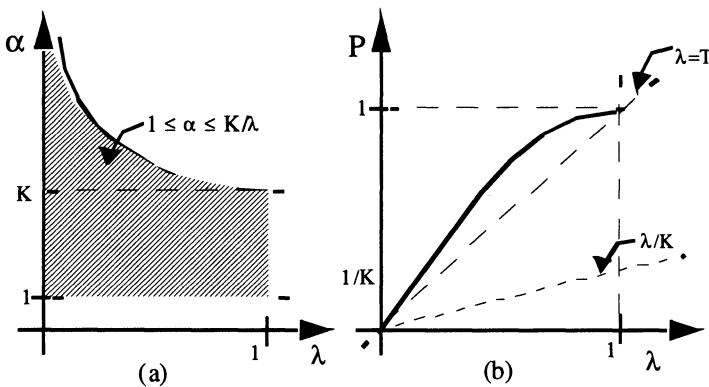


Figure 6: illustrate effects of traffic divided into K ports

The exact value of α depends on many details of the binary exponential back-off algorithm used in the contention resolution algorithm, but we can interpret the boundary cases intuitively. In a highly loaded system with large λ , we have modems that failed to acquire requests slots previously utilizing the newly available request cell slots in the now lightly loaded return path, which results in $\alpha \cong K/\lambda$, and $P = \alpha\lambda/K \cong 1$. Only in a very lightly loaded system do we have $\alpha = 1$, where no addition request cells are generated. In this case, with $\lambda = \varepsilon \cong 0$, we can write $P = \alpha\lambda/K \cong \varepsilon = \lambda$. The resulting P is plotted in Figure 6 (b), where we see

qualitatively how the portion of slots occupied is increased from λ/K as a function of λ .

The round-robin RMUX scales the throughput of each port by allowing only $1/K$ th of the slots through, which gives us a throughput of $\alpha\lambda/K^2$. Combining the K paths, the aggregate throughput is then $T := \alpha\lambda/K$. Using the value of α reasoned above, we can plot the approximate throughput T as a function of λ in Figure 7 (a).

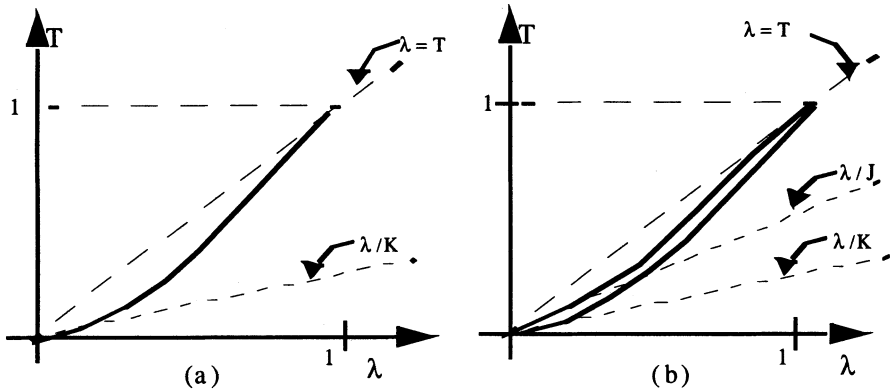


Figure 7: illustrate approximate analysis of RMUX throughput

3.2 Grouped Round-Robin Service

We model a RMUX with a grouped round-robin service using a polling system with a cyclic server, a reduced number of buffers, and aggregated arrival processes. Let J be the number of aggregated arrival processes, where $J < K$. As before, we have $s_i(n) = 1$, for all i, n .

For simplicity, we examine the case where $J = K/2$, where adjacent ports are grouped into pairs. With the number of input ports reduced by $1/2$, the probability that an arriving request cell will not be blocked is now $2/K$. Following a similar line of reasoning as above, we have $1 \leq \alpha \leq K/2\lambda$, and the throughput T is bounded below by: λ/J . Since the increased aggregated load on each port will push α towards $K/2\lambda$, we argue that the throughput T will be approximately as shown by the top bold line in Figure 7 (b).

We can therefore expect that grouping of ports, when allowed under noise and load conditions, will result in better performance.

3.3 Load Dependent Acyclic Service

We model a RMUX with a load dependent acyclic server using a deterministic fluid approximation polling system model with a rate dependent service discipline.

Instead of fixing $s_i(n)$ based on a fixed loading of the port, the service discipline now adjusts $s_i(n)$ dynamically as a function of time and the load on each port.

Extending existing scheduling results (Clear-A-Fraction policy from (Perkins)), we propose the following policy where $s_i(n)$ is based on estimated per port load.

The policy we propose is called the Highest-Load-First (HLF) policy. Let t_m be the time at which the server visits the m th buffer. Let $e_i(m)$ be the estimated number of arrivals into buffer i during $[t_m, t_m + s_i(\max)]$, where $s_i(\max)$ is a predetermined maximum number of slots each port can be allocated at any time. Such a value can be easily selected based on maximum latency of a HFC system, as well as based on minimum allocations to each port. With HLF, after the server has finished serving a buffer, the next buffer to be visited is buffer j , where $j = \max\{i | e_i(n)\}$. The number of slots serviced at port i is then set to $s_i(n) = \min\{e_i(n), s_i(\max)\}$.

Earlier work (Perkins) has shown that such a policy is complete, that is, a set of $s_i(\max)$, $i = 1, \dots, K$, can be selected to ensure that every buffer will be visited in a given time period. A more generalized version of the HLF is the Fractional-Load (FL) policy, where the next buffer i is selected based on whether $e_i(n) > \varepsilon \sum_j e_j(n)$, where $0 < \varepsilon < 1$, is satisfied. Assuming the estimated load equals the actual load, these scheduling policies have been shown to be stable under different initial conditions. The key potential advantage of these policies is that they are relatively easy to implement and have parameters that can be tuned to optimize performance.

4 SIMULATION MODEL

The analytical models above do not describe the behavior of the system quantitatively, nor do they capture the performance of the system under realistic traffic. By realistic traffic, we mean the traffic generated by the typical cable modem user. That traffic is TCP/IP based, and is mainly generated by World Wide Web (WWW) and FTP client-server application. Such traffic introduce yet another layer of flow control and also is influenced by human behavior not easily captured by analysis. For these reasons, we rely mostly on simulation.

The simulator we use is a customized version of the *ns* simulator (Ns, 1998), a timed discrete event simulator. The simulator contains modules for simulating the UPSTREAMS HFC MAC protocol (Laubach, 1997), full TCP protocol behavior, and WWW browser client-server interactions as described in (Nichols, 1997). Our WWW client-server model, including file size distributions, is based on previous trace data (Cunha, 1995).

Using this version of *ns*, we simulated the system using the following parameters:

- Number of users/modems: Several user population sizes (ranging from 64 to 2000) are used.

- HFC network: For simplicity, we assume a single upstream receive port at the headend and a single downstream channel. Depending on the RMUX configuration, four or eight upstream return paths feed into the single RMUX.
- Quality of service: the HFC system simulated allows maximum and minimum rates to be specified for both up and downstream traffic.
- Servers: For simplicity, we assume there is no limit to the number of concurrent WWW sessions using the HTTP protocol at the servers.

In Figure 8, we illustrate the major components in our simulation model. Important to note is that this is a detailed model that takes into account the behavior of the TCP protocol, but also the interactions between the ATM protocol (e.g., Segmentation and Reassembly - SAR agent), the UPSTREAMS protocol (Laubach, 1997), TCP, and higher layer applications (e.g., WWW/FTP server and clients).

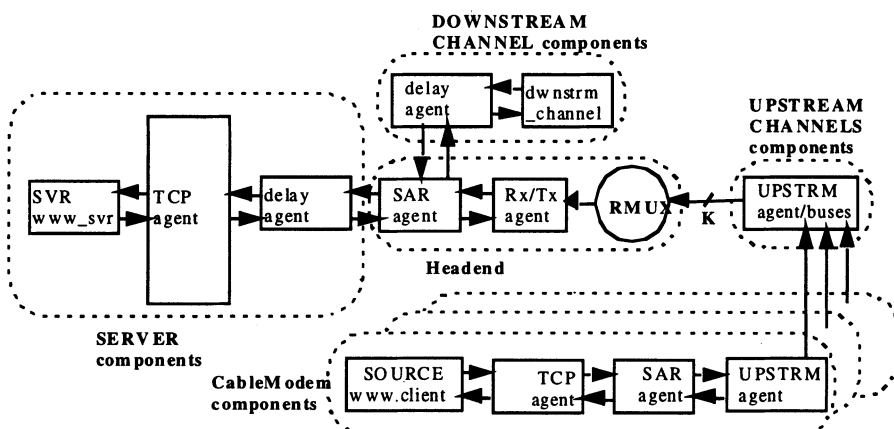


Figure 8: components in the simulation model

During each WWW session between a client and the headend server, a sequence of short and long packets containing TCP SYN and SYN+ACK, URL, and document pages is exchanged. As described in more detail in (Nichols, 1997), each such WWW browser exchange is currently configured with the following parameters:

- TCP window size of $6 \times 1460 = 8760$ bytes
- Packet size of 64 bytes.
- URL size = 3 cells
- Page size distributed according to pareto distribution up to 1.5MB
- Each page contain 4 in-line documents (e.g., images) whose size is distributed according to the pareto distribution up to 0.25MB
- Initial delay of 0.

The resulting average page size is around 10KB, while the in-line document size is around 5KB.

The duration of a typical simulation run is between 150 and 600 seconds in simulated time. This translates into real time depending on the number of events generated, which is proportional to the number of cable modems, load, and collision frequency. Using an Ultra Sparc workstation, a moderately loaded simulation run with 512 cable modems and 300 simulated seconds takes around 3 hours to complete, while a simulation run with 16 cable modems takes about 15 minutes. To eliminate the effects of transient behaviors, the initial 30 seconds of all simulations runs are excluded from the final tallied results.

5 SIMULATION RESULTS

Using the simulated WWW traffic above, we simulated the performance of a HFC system with a RMUX.

Performance metrics

The performance metric used are:

- Latency: median and 90 percentile end-to-end delay for each modem.
- Throughput: average throughput for each modem.

Statistics about cell level, packet level, and page level performance are recorded, though only cell and packet level statistics are presented here.

5.1 Effects of Load Distribution

In Figure 9, we have plotted the upstream latency comparisons between varying number of cable modems. The number of modems was varied from 64 to 2,000, and clearly independent of the number of modems, the index of symmetry (IOS) as defined in equation (1) was the dominant factor in the latency performance.

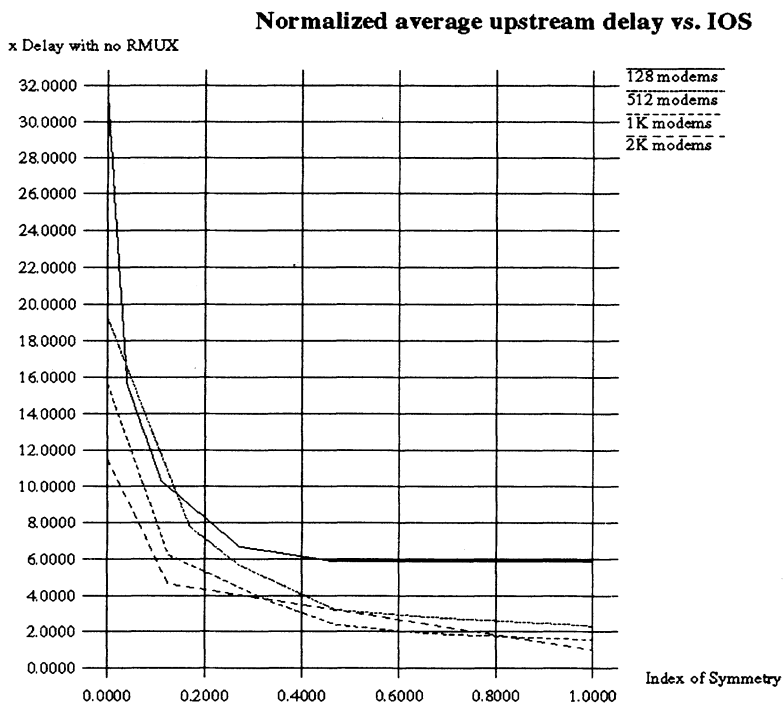


Figure 8: upstream latency as a function of Index of Symmetry

From Figure 9, we see that increased IOS brings about dramatic improvement in the delay behavior. The curves also flatten out to around 2 for an IOS value of approximately greater than 0.4. Using equation (3), this threshold of 0.4 implies that one should place no more than about $1/4$ ($N/3.8$, to be exact) of the cable modems on one input port in a 8×1 configured RMUX, or no more than $N/2.2$ cable modems on one input port of a 4×1 RMUX.

What this implies for throughput is a more complicated question to answer. This is because how increased upstream latency manifests itself in TCP throughput degradation depends on many factors. In (Cohen, 1998), the effects of upstream delay and error on TCP throughput was studied in detail. Although the study did

not take into account the behavior of the HFC MAC, it does illustrate the sensitivity of TCP throughput to the HFC upstream integrity.

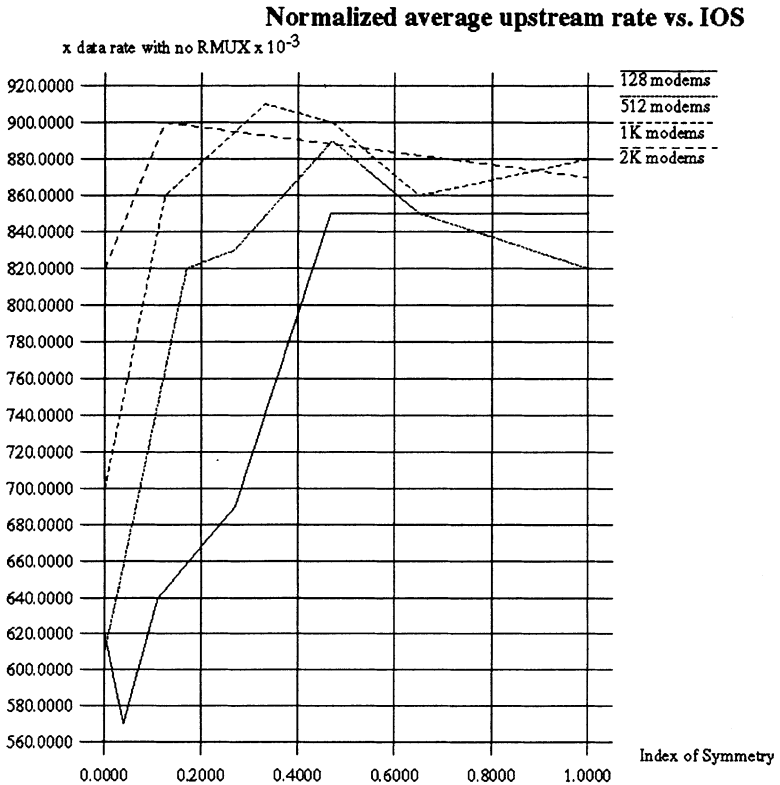


Figure 10: upstream throughput as a function of Index of Symmetry

In Figure 10, we see that there is a knee in the curves at an IOS value of 0.2 - 0.4, and the throughput of highly loaded systems are much closer to the throughput of a system without the RMUX. This was predicted by the simple analysis illustrated in Figure 7. It is not clear why the throughput appears to decrease as the

IOS approaches 1, but it seems that the decrease is accompanied with a decrease in the throughput variance.

5.2 Effects of Service Discipline

We present here a comparison of Round-Robin vs. Grouped Round Robin.

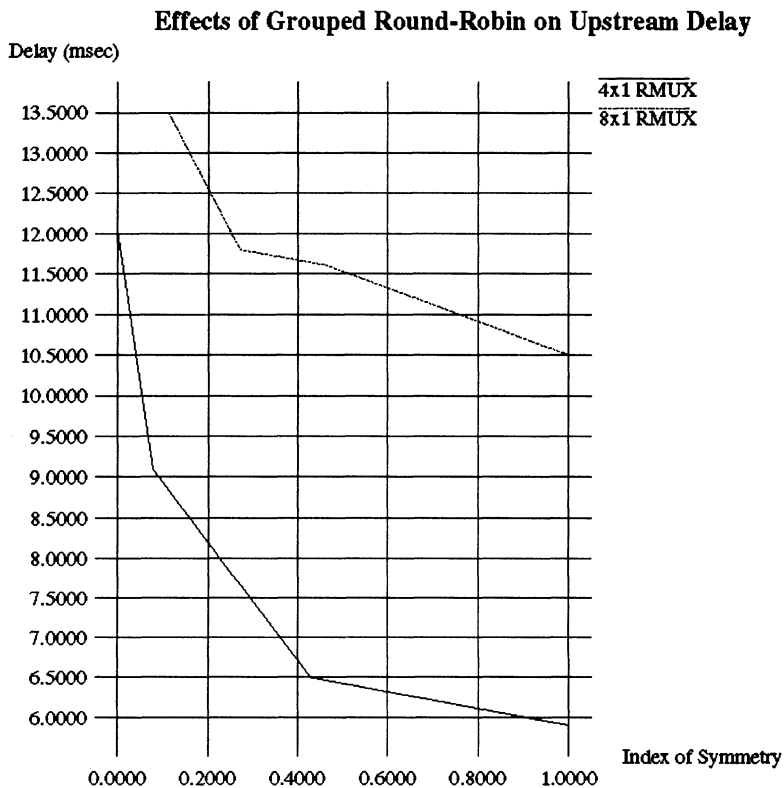


Figure 11: latency improvements from grouped round-robin

In Figure 11, we plotted two sets of simulated latency results from WWW client-server session for a fixed number of modems. We see that by grouping lightly loaded ports into pairs, the resulting latency of TCP transfers is significantly improved. In the $K=4$ case, the upstream delay varied from 6 to 9 msec. For $K=8$, we see that the latency increased to the range of 10 to 13.5 msec, a substantial increase.

This implies that the HFC system architect should identify lightly loaded ports and configure the RMUX to serve those ports concurrently. Doing so will greatly lower the average upstream latency of the entire system. This result also tells us that selecting the smallest K possible when configuring a RMUX will improve the system performance.

6 CONCLUSIONS

Focusing on the scheduling disciplines of a RMUX and how its performance depends on the load distribution of the upstream traffic, we have illustrated through simple analytical and simulation results that Return Path Multiplexing is a useful way of reducing headend port requirements (by a factor of K) in a HFC plant without significant performance degradation. Properly configured, simulation results show that greater than 80% throughput can be achieved. The RMUX's multiplex service discipline only operates during the request phase on top of the contention resolution mechanism of the media access protocol (MAC), and does not affect the operation of the MAC during the data transfer phase.

The simulation results presented in this paper used realistic simulated WWW traffic sources, and validated some of the simple analytical results. However, it is clear that a more detailed simulation study of the RMUX behavior is needed. Specifically, effects of return path multiplexing on TCP level downstream throughput and latency needs to be studied.

We see that a simple cell level switch like the RMUX dramatically changes the architecture and provisioning policy of a cable modem system. Rather than having to accept a given physical layer topology and plant RF characteristic, the RMUX allows the cable operator to change a given topology without detrimental performance consequences typical of current passive RF combination methods. Whereas capacity planning and resource allocation of the HFC plant used to be issues constrained by the given RF condition of the return paths, now a device like the RMUX allows the cable operator to dynamically manage or isolate noisy return paths.

However, we have also seen that it has an operating region that the HFC architect must be made aware of. Although the operating region identified in these results is large and can be easily engineered, an RMUX with an overly asymmetric load, as illustrated by the simulation results, will see performance degradation. In

order to properly configure the RMUX, a HFC system that allows the operator to provision the bandwidth allocation of each modem is required. This highlights the need for the ability to provision and enforce QoS in a HFC system.

There are many aspect of the RMUX's operation which we have not touched on. As with anything in the HFC plant, its proper performance depends primarily on the RF condition of the plant. Indeed, with favorable SNR in the HFC plant, it may be acceptable to service all the input ports of the RMUX concurrently during the request phase. This will allow the RMUX to operate without blocking, and simply let the contention resolution of the HFC protocol operate normally. Note that since the RMUX does not funnel noise during the data transport phase of the protocol's operation, the resulting performance will be substantially better than using a simple passive RF combiner. The precise economic impact of a device like the RMUX is an important topic worthy of study, and investigations into other methods of achieving better RMUX performance are also ongoing.

Disclaimer

The results presented in this paper do not necessarily represent the performance or design of Com21's commercial product that enables return path multiplexing.

Acknowledgments

The idea of return path multiplexing originated not with the author. Instead, it came from colleagues at Com21 driven by HFC system deployment experience. A tremendous amount of effort by the whole Com21 team went into the research and development of the commercial product based on the idea. They are the ones who made this paper possible.

7 REFERENCES

- Cohen R. and Ramanathan S. (1998) TCP for High Performance in Hybrid Fiber Coaxial Broad-Band Access Networks, *IEEE/ACM Transactions on Networking*, Vol 6, No. 1.
- Cunha C.R. et al, Characteristics of WWW Client-based traces, Boston University Computer Science Technical Report, BU-CS-95-010, July 18, 1995.
- Laubach, M. The UPSTREAMS Protocol for HFC Networks, Version 1.09 970623, SCTE-DSS-97-xx, (ed. M. Laubach, Com21, Inc.), June 23, 1997.
- Ns (1998), University of California at Berkeley/Lawrence Berkeley National Laboratory/VINT Network Simulator, <http://www-mash.cs.berkeley.edu/ns/>
- Nichols K.M. and Laubach M., Tiers of Service for Data Access in a HFC Architecture, *Proceedings of SCTE Convergence Conference*, January, 1997.
- Perkins J.R. and Kumar P. (1989) Stable, Distributed, Real-Time Scheduling of Flexible Manufacturing/Assembly/Disassembly Systems, *IEEE Transactions on Automatic Control*, Vol 34, No. 2.

Takagi H. (1986) Analysis of Polling Systems, The MIT Press, Cambridge, Massachusetts.

8 BIOGRAPHY

James C. Yee is a Performance Engineer at Com21, Inc., a company in Milpitas, CA, USA that makes cable modem head-end equipment, cable modems, network management software, and noise-containment technologies. He received his B.S. from Columbia University in 1987, M.S. and Ph.D. in Electrical Engineering and Computer Sciences from U.C. Berkeley in 1988 and 1994, respectively. He is also an adjunct faculty at the Computer Engineering Department of Santa Clara University.

His research interests include network performance modeling and analysis, real-time scheduling, ATM, and IP services. Previously, he had worked as a networking consultant, at Pacific Telesis Group, and at Philips Research.