Combining Multilayer, Topological and Configuration Information in Performance Analysis

Marc F. Pucci

Abstract --**The thorough analysis of network performance often requires understanding the details of activity occurring in several dimensions. First, there is the information that can be observed by active or passive measurements of network traffic on a link. This in itself is a multilayer problem in that activity at one layer, say loss at the ATM transport layer, can impact another, for example, IP layer performance. A second dimension concerns topological relationships, both physical and logical, that interconnect network elements and higher level service points. In this dimension we find the physical connections, MPLS pathways, routing tables, etc., that bridge a network together. Finally, we add the configuration details that define service characteristics or link behavior. Here we include transmission MTUs, leaky bucket parameters and high-level policies on traffic such as admission control. We need to visualize all of these interrelationships in order to understand subtle details that affect user-perceived performance.**

Using examples drawn from detailed, packet-by-packet analyses taken over the years, we demonstrate how understanding the complete nature of network performance requires reconstructing much of the basic network structure – across multiple layers and in consideration of physical and logical topological relations. At times it is necessary to shift between analysis techniques that average away details, such as histograms, and instead examine long sequences of individual packet delays to uncover the cause of network degradation.

We conclude with a description of work in progress to use data modeling techniques to represent these multilayer and topological considerations so that we can automate some of the processes involved in the determination of the root causes of performance problems.

Index Terms-- **data modeling, network performance analysis, trace analysis.**

I. INTRODUCTION

Great effort has been expended in diagnosing anomalies in networks. The methods used generally rely upon the experiences of the analyst, gleaned from situations or patterns previously encountered. Recognizing measurements that 'feel' wrong can point us in the direction of a problem we may not know exists. More often, knowledge of details that are external to the measurements at hand are crucial to the solution, and is needed to understand all the complexities at work.

The author is with Applied Research at Telcordia Technologies, Inc., 445 South St., Morristown, NJ 07960. (Email: marc@research.telcordia.com)

For many years, Telephone Operating Companies and Network Service Providers have monitored and analyzed their communications links in order to improve the level of understanding of the traffic on their networks. This knowledge has been used to improve the provisioning and operational procedures that keep these networks running at extremely high reliability and efficiency, and has established the base lines that bracket expected, normal behavior.

We have been involved in many of these measurement and analysis cycles, especially when the amount of network data traffic increased significantly in proportion to voice traffic. In particular, new data are needed to evolve, improve and validate changing traffic models. In every case, we uncovered some unexpected condition in the collected data that warranted further study. Often the details needed to resolve a problem fully were unknown at the time of the measurement process, and only discovered after we ran into inconsistencies during analysis. The process of reconstructing this ancillary structure and of building an informal data model to represent this information (in the head of the analyst) now leads us to pursue a more formal representation that can be used to improve the overall process. We hope to capture the complete physical and logical network connectivity and use this as a framework for storing and analyzing measurement data.

II. BACKGROUND

The examples used in this paper are drawn from T1 link measurements of Frame Relay traffic. The instrumentation equipment consisted of a set of independent, intelligent I/O processors, each dedicated to a single bi-directional circuit, to gather and time-stamp incoming packets with 1 ms resolution. A common Single Board Computer running a real-time operating system managed the large memory buffers for the collectors and maintained the secondary and tertiary storage (8mm tape stackers) used to archive the data.

The amount of data recorded from each packet is configurable. It can be set to a small constant, a protocol dependant size for a built-in set of recognized Frame Relay headers, or the entire payload. By guaranteeing the security and integrity of both the collection process and the archival data, we were permitted to collect entire payloads. An intermediate processing step scrubbed sensitive information from the data set.

We cannot overstate the value of collecting entire packets. In some cases we have found the degree of encapsulation to

be extremely deep, and would have hampered our analyses if we could not dig into the payload. This is especially true in a central office environment, where legacy communications protocols must be preserved across technological changes, requiring encapsulation to protect the underlying format. As shown in Figure 1, an actual packet breakdown from a trace dataset, one layer's payload is another layer's header. In addition, the payloads from link management packets often contain interesting information such as routing table updates, channel activations and the like, that are useful in diagnosis. A more sophisticated implementation of packet payload processing is described in [1], a data warehousing system that parses, compresses and stores payload data into a database for subsequent analysis.

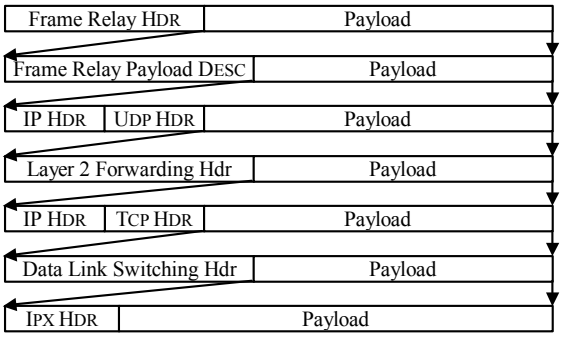

Fig. 1. Encapsulation in the extreme. Limiting the size of collected payloads can result in information being truncated or completely unavailable. This actual IPX packet would never have been recognized if we had captured only the first 64 bytes of the initial payload.

Complete packet traces add a burden to the collection process, limiting its capacity and requiring greater bandwidth to secondary storage. However, we were often in the position where our equipment could only be deployed for a limited duration in a crowded central office, and therefore we did not have the ability to return and alter the measurement technique.

In many of these sites, data were collected at several points in a network in order to compute transit characteristics such as delay and loss. In particular, there was a great deal of interest in localizing the occurrence of loss to the interior or exterior of the network. Collections were taken over prolonged periods of time, usually one week, sometimes several weeks, and in some cases were repeated after a year's time. Given the line speeds, line utilization and the assistance of local craftsmen to reload tapes, we were able to gather large sets of data for subsequent analysis.

As mentioned above, the examples are taken from a variety of Frame Relay networks monitored over the years. For completeness, we briefly describe the basics of this data format. Frame Relay is a link layer protocol that fits in as layer 2 between the physical and network layers of the OSI model. A frame consists of a leading flag byte, a 2-byte address field, up to 8K bytes of payload, a 2-byte CRC and a trailing flag byte. Frame Relay uses Data Link Connection Identifiers (DLCI) for addressing, which essentially define paths through a series of switches. Multiple DLCIs separate frames into logical flows along the same physical link. The address field also contains 3 bits related to frame behavior through the network. These are the Forward and Backward Explicit Congestion Notifiers (FECN, BECN) that indicate queuing problems in intermediate switches, and Discard Eligible (DE) that marks a frame as having exceeded its allocated burst rate and is subject to dropping. A payload descriptor identifies the particular protocol used to convey information in a frame. This Network Layer Protocol Identifier (NLPID) is a variable length, multi-byte header that is often abbreviated to conserve bandwidth. Different NLPIDs can be used on the same DLCI to carry different packet formats.

III. ILLUSTRATIVE EXAMPLES – PART 1

Studies of the traffic mixes, holding times, etc. from these data sets have been reported elsewhere [2], [3]. We concentrate here on the detection of anomalies in these traffic traces and on the sometimes 'murder mystery' investigation that ensued in order to isolate the cause.

A. Unexpected Loss

The usual factors that contribute to loss in a network are either degraded link transmission characteristics that cause framing errors or CRC failures, or buffer overflows where bursty traffic exceeds the capacity of link interfaces. A symptom of the former will be indicated by MIB-2 peg counts for received-errored-packets, while the latter can be seen as output-discards also recorded in the MIB.

It is important to record and correlate errors across network layers so that losses at higher levels can be reconciled against failures at lower ones. ATM cell loss will indirectly impact IP layer metrics. A dropped cell will cause AAL5 reassembly to fail, thereby causing a packet to be lost at the IP layer. This can be undetected by either of the methods outlined above unless layering and topological relationships are understood. Figure 2 illustrates this case. A data model representing this subnetwork can be found in Section IV.

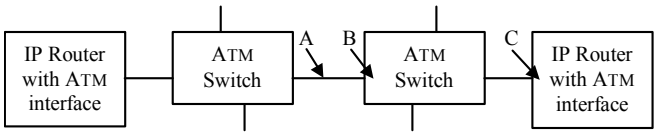

Fig. 2. Uncorrelated loss at the ATM layer. An errored cell at point A would be recorded at point B in the switch's cell statistics. No indication would be seen at point C, in the router's packet input MIB.

Even when such considerations are taken, loss can still creep in from unexpected sources. Figures 3 – 6 are plots derived from a set of measurements taken at the ingress and egress of a subnetwork. The losses measured did not correlate

with any link layer statistics. The configuration in question was operating within reasonable load variations. Neither the affected traffic nor that on other logical channels of the medium exceeded the link's Committed Information Rate (CIR), where excessive traffic would cause a switch to drop frames intentionally. The aggregate traffic did not exceed total link capacity.

Figure 3 shows the total link traffic averaged over 5 minute intervals. Points in the figure marked 'A' and 'B' locate regions of low and high loss, respectively, on this and subsequent charts. While point A occurs at the busiest link activity, it corresponds to relatively low loss. Point B, at low utilization, corresponds to packet loss that exceeds 25%. The levels occurring here are insufficient to generate losses due to queue overflows.

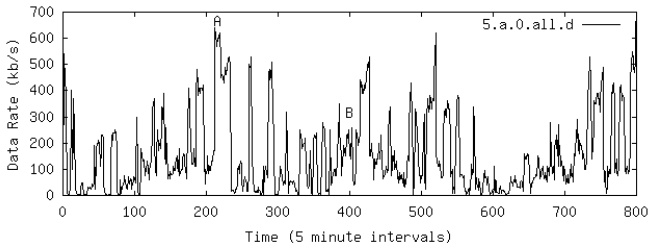

Fig. 3. Total data traffic across all DLCIs. Load never exceeds 50% of T1 capacity (1500 kbps). The points labeled 'A' and 'B' identify areas of low and high loss, respectively, and can be found in the next 3 figures.

Figure 4 shows the activity in terms of packet rates over the same interval for the entire link and for the busiest logical channel. The latter represents most of the transmitted link load, accounting for 94% of the packets and 77% of the bytes transmitted over the link. The difference between the 2 traces is often difficult to distinguish, and is shown separately in Figure 5. Note the scale change in this figure. The activity on the other channels is considerably less than the total.

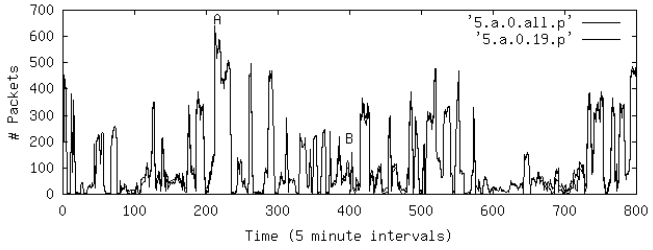

Fig. 4. Total packet load across all DLCIs and across the busiest DLCI. This DLCI accounted for 94% of packets and 77% of all data and so, closely matches the total. An expanded view of the difference is shown below.

Figure 6 shows the total link loss. It appears that the correlation to loss is not a function of the amount of traffic, but of the number of active sub-channels (DLCIs) in use on the physical link. In a short time span, if the number of packets on the link from multiple, independent channels was high, the loss rate increased even if the total number of packets was relatively low.

We attribute the losses to the inability of the link processing hardware to handle simultaneous channels, rather than the more commonly expected loss due to buffer overrun. Apparently, the switching circuitry could not keep up with the number of frames that needed to be examined, possibly because of a limited size cache table for output selection. An interesting metric to include for subsequent analysis would be a measure of the number of consecutive packets or cells that would be routed or switched by a network element in the same manner. At the least, it would be informative to record the CPU utilization of the processors involved.

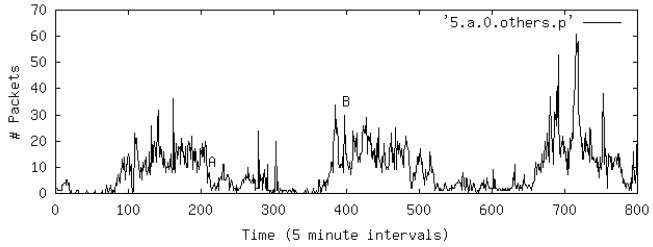

Fig. 5. Combined packet load across all but the busiest DLCI. Note the scale change from above. Traffic on these links represents only about 5% of the overall link load.

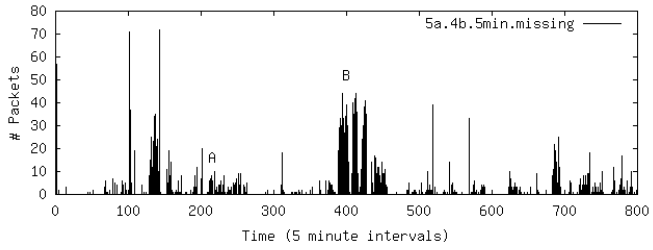

Fig. 6. Packet loss across the entire link. The loss is more correlated with activity on adjacent channels than with the total amount of traffic.

B. Delay Distributions and Configuration Errors

B.1. Service Configuration Errors

Packet delay distributions typically cluster about a single value that represents the base transmission time across a network and also include outliers that extend beyond this base. Often these outliers extend a considerable distance, especially when many network elements are involved. Figure 7 is an example taken from the forward traffic on a T1 link. The reverse traffic delay distribution is shown in Figure 8. Note how the baseline delay differs in the two directions. We have often found these nonsymmetrical relationships in central office environments where differences in the sending and receiving equipment will multiplex and demultiplex signals with dissimilar results.

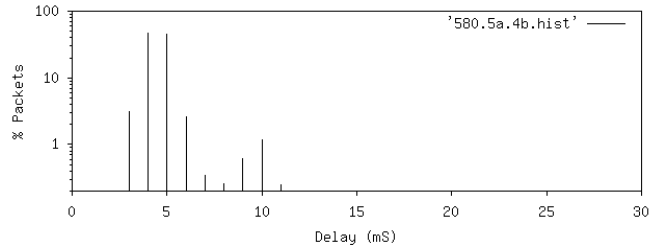

Fig. 7. Link delay distribution showing variance about a central value with limited outliers.

Most of the links we measured fell into the broad, single peak category. Some, as shown in Figure 8, exhibit a multi-node distribution, due to the interspersing of small and large packets that arrive independently and vie for placement on a single outbound link. This effect is more pronounced when the packet transmission time is comparable to the overall delay.

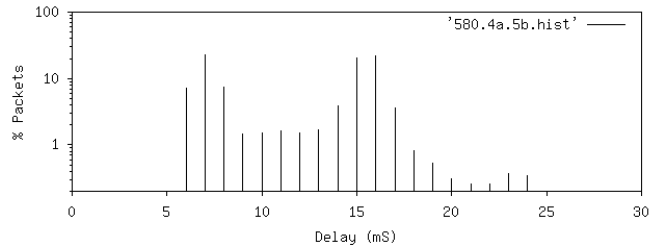

Fig. 8. Bipartite distribution with tail. The 8 ms separation between peaks corresponds to the transmission time of a 1500 byte packet at line speed.

The difference between nodes is about 8 ms, approximately the time required for the transmission of a 1500 byte packet on a T1 link. Small packets arriving at the outbound link were occasionally delayed by the transmission of a much larger packet. A third node is evident around 23 ms, near the end of the tail. These nodes and their implied jitter reinforce our concern that network provisioning must take traffic characterization into account to avoid cases where mixing streams with differing packet lengths can cause unacceptable side effects. In particular, small voice packets and large file transfer packets tend not to mingle well without degradation.

Since the access rates of DSL and cable modems are in the T1 range, the use of these facilities to offer simultaneous voice and data services needs to be provisioned carefully, especially with regard to the Maximum Transmission Unit size (MTU). The mixing of large data packets with small voice packets can increase the amount of jitter present on an end-to-end connection, pushing it beyond the voice hardware tolerance level for acceptable voice reconstruction. This is especially troublesome since jitter can have a more dramatic impact on voice quality than steady-state delay. Reference [4] is an example of a system that defines a service grammar to insure that configuration requirements are satisfied before a complex, end-to-end service can be turned-up.

B.2. Line Card Configuration Errors

The delay distribution shown in Figure 9 is unusual in the shape of its envelope. The y-axis is not a log scale and does not overemphasize smaller values. The shape of the tail of the distribution enticed us into examining the delays on a packet-by-packet basis from the trace files.

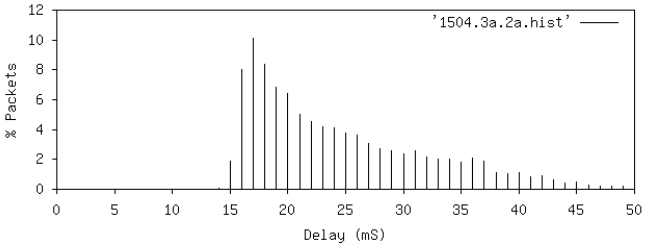

Fig. 9. Packet delay distribution with unusual envelope. Note that this is a linear scale and that the tail delays are not overemphasized.

Figure 10 shows the individual packet delay times of a 5-minute sequence of packets. Note the constant rate at which the delay increases. The rate is approximately 1.4 ms per second. The majority of packets were 1500 bytes long and represented a number of large file transfers. The sudden drops generally appear every 30 seconds.

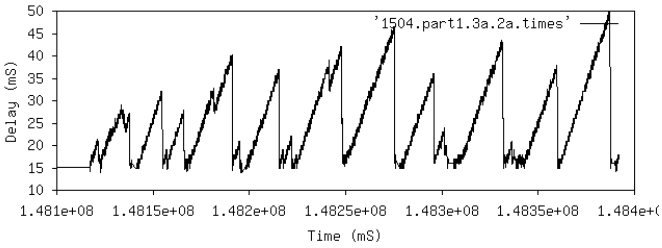

Fig. 10. Delays for a 5-minute sequence of packets. Each consecutive packet was delayed by a constant time from its predecessor until a periodic reset restored the baseline about every 30 seconds.

The steady increase in delay represents about 2 byte times per 1500 bytes at a full T1 speed of 1.544 Mbps. The HDLC transport mechanism used for Frame Relay brackets each frame between a pair of flag bytes for synchronization purposes. One of the configuration options for HDLC allows packets that are back-to-back to share a common flag byte such that the trailing flag byte of one packet becomes the initial flag byte of the next. Refer to Figure 11. If incoming packets arrive at a network element in this back-to-back fashion but leave without this sharing, additional bytes and gaps must be inserted in the data stream. During periods of high utilization, this will result in the delay pattern seen.

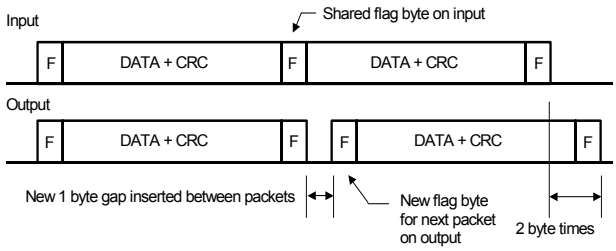

Fig. 11. Input and output relationship between adjacent packets. Packets can arrive sharing a common flag byte 'F' (top) but leave with individual flag bytes (bottom). In addition to the extra flag byte, a one-byte inter-packet gap is needed to separate the flags.

The ramp-up came fairly close to the configured link buffer size, but never reached it, so this problem never revealed itself as a measurable interface loss statistic. At approximately 30-second intervals there was enough of a delay in the source traffic to allow the buffer to drain. We suspect this was due to some periodic process running on the source that caused the packet generation process to be preempted.

Detecting such configuration problems is a complicated procedure. Nonetheless, the periodic capture of limited trace data is useful in uncovering subtle errors that may not appear often.

C. Packet Sizes and Encapsulation Layer Problems

This example illustrates how knowledge of higher-level functions in the network, such as encapsulation, is needed to detect the root cause of excessive fragmentation in a transport system. A simple profile of packet sizes will often show a common set of impulses around particular values [6], [7], [8]. These tend to correspond to the smallest TCP packet that can be transmitted (40 bytes), as well as the largest packet (~1500 bytes) and others around 512 bytes as shown in Figure 12. An unexpected impulse can be caused by particular application traffic such as voice or game data, but can occasionally indicate other problems.

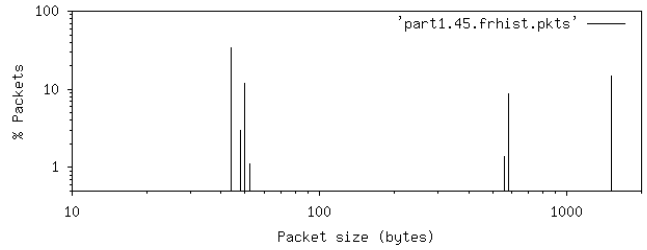

Fig. 12. Packet size distribution with expected peaks around 40, 500 and 1500 bytes

Contrast the profiles in Figures 12 and 13. The additional impulses occur around 13, 80 and 300 bytes. The 13-byte packets contained Link Management Information (LMI) used in the maintenance of Frame Relay connections. The 300-byte packets contained periodic router announcements. The 80-byte packets were not as easily explained.

The network we were monitoring used Layer 2 Forwarding [5] to encapsulate the traffic. The addition of the L2F header increased the overall transmission packet size. For the largest packets, the added header would now exceed the MTU of the network link.

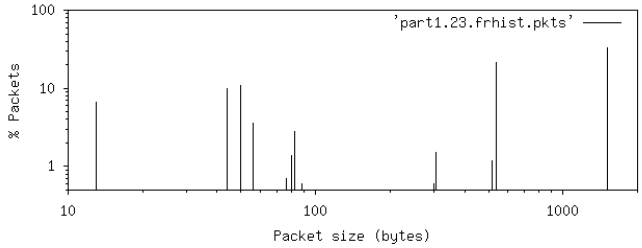

Fig. 13. Packet size distribution with unusual peaks around 13, 80 and 300 bytes.

Each large packet would then be fragmented into 2 packets, one carrying the bulk of the traffic, the other, the bytes that were bumped from the first to accommodate the header. This caused both an increase in overall network utilization as well as CPU cycles for fragmentation and defragmentation. As shown in example A, line card CPU cycles can be a critical resource. Since the payload size for Frame Relay can be as large as 8K bytes, it is possible to redefine the MTU to allow the encapsulated 1500-byte packets to travel without fragmentation.

D. Ineffective Load Balancing

This example illustrates how neighboring traffic, thought to be independent, can be strongly correlated. The analysis required knowledge of the adjacent topology of the network and the alternate routes that existed between the source and destination.

Layer 2 Forwarding (L2F) was used to offload data traffic from traditional voice traffic switches and to convey the former directly to a Data Service Provider (DSP). L2F essentially wraps a dial-up PPP connection between a Network Access Server (NAS) and a DSP. An individual L2F multiplexor identifier session is established for each dial-up user. A single L2F tunnel is used to carry all traffic from the same modem pool to the same DSP. A pair of links was used to carry this traffic, as it was provisioned to exceed a single T1's capacity.

Our initial examination of a week's data indicated the daily variations in utilization that we have come to expect, and is shown in Figure 14. This is, of course, dramatically different from the conventional utilization of voice links, which is the basis of much traffic engineering research and was one of the reasons for monitoring these networks. In particular, the peak hours occur much later at night, and the weekend traffic is

still prominent. However, regardless of the particular envelope shape, the traffic utilization seemed much too jagged.

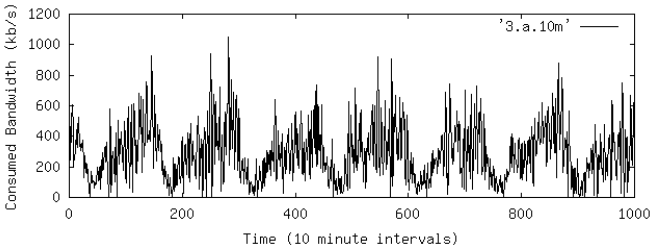

Fig. 14. Link bandwidth over a one-week period. The typical daily variations in load are evident. The jagged shape of the curve is unusual.

A magnification of a one-day period is shown in Figure 15 and illustrates this concern. The traffic pattern varied too much about the envelope to be reasonable. The higher frequency components represent 20- and 40-minute cycles that continuously ride the longer daily pattern. Note that the link is driven towards 0% utilization during periods of moderate activity.

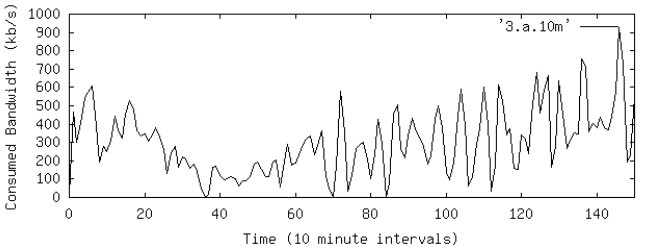

Fig. 15. Link bandwidth over a 1-day period showing unusual 40-minute cycles.

Figure 16 is a plot from the other link in the measurement set and also reveals the same load pattern. Reverse traffic on these links indicated similar variations, though not as dramatic in the magnitude of the swing. After careful examination of the accuracy of the traffic recording system, we verified that there were no 20- or 40-minute time constants in the collector.

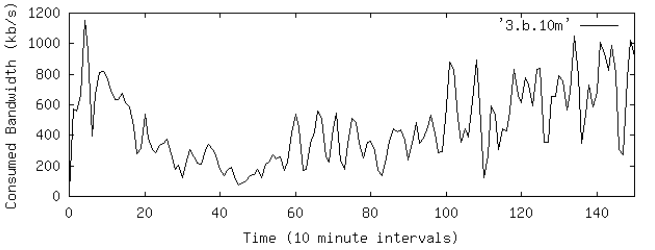

Fig. 16. Link bandwidth over a parallel link for the same period.

Because our data collectors were configured to record the entire packet payload, we were able to examine and decode the Layer 2 Forwarding encapsulation layer surrounding the data. One goal of our analysis was to extract the duration of these sessions to enhance our traffic models of user behavior. In so doing, we identified periods of inactivity in the sessions. Far too many of these gaps lasted exactly 20 minutes; it seemed unlikely that this behavior was unrelated to the load variation problem.

The final clue to the solution involved superimposing the individual load lines, as shown in Figure 17. The traffic buildups on these links were clearly out of phase. Increases on one were somewhat balanced by decreases on the other.

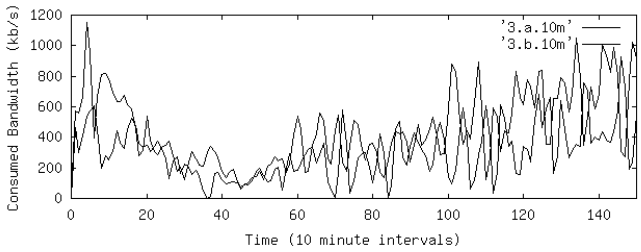

Fig. 17. Superimposed link bandwidths for both links showing the out-of-phase relationship between consumed bandwidth caused by an ineffective load balancing mechanism.

The multiple links were being used to load balance the total traffic between the NAS and the DSP. To corroborate this we examined the individual sessions within the L2F tunnels to see if the gaps corresponded across the links. In many cases we were able to track a single session from one link to the other and back again. However, not all gaps were reconciled.

By summing the traffic on the two links, we expected the overall traffic to become relatively smooth. As shown in Figure 18, the size of some of the swings has diminished, but not to the point where the traffic would resemble a normal, slowly varying load pattern. We suspected the existence of another link carrying load balancing traffic and consultation with the client confirmed this to be the case. Unfortunately, we could not re-instrument the experiment and measure this new link. It does explain the missing gaps from the session analysis. Fortunately, we were able to extract a sufficient number of long sessions that remained on the 2 monitored links for our user modeling work.

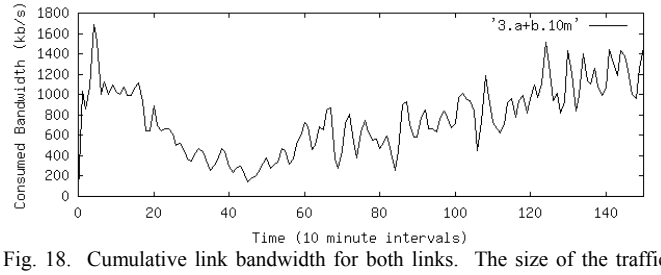

Fig. 18. Cumulative link bandwidth for both links. The size of the traffic variations has decreased (contrast with Fig. 16) but has not disappeared. We later discovered a 3rd link carrying traffic that we had not measured.

Curiously, the load balancing algorithms were different in the forward and reverse directions. The forward traffic would rebalance many existing sessions every 20 minutes. The reverse traffic would assign a new session to a less loaded link, but would not relocate it once established. Unfortunately, it also continued this process beyond the point where the links would have been balanced. This accounted for the less dramatic variations on those links. In either case, both load-balancing configurations were inefficient and unstable, barely providing an increase in capacity over that which a single link could have achieved. This, coupled with the potential for packet reordering on the forward link when sessions were moved between links, made this configuration inappropriate for its intended use.

IV. DATA MODELING

Before describing the last and most complex example, we present a brief description of the data modeling procedure. The investigation of the last example, to be found in the next section, will benefit from the techniques shown here. The modeling is styled after that in ITU-T Recommendation M.3100 [9].

Recall the example shown in Figure 2, which illustrates a conventional IP over ATM configuration. Figure 19 models this configuration. Each box encloses the realm of a Managed Element (ME) and includes the logical and physical connection points (circles) that terminate there. The horizontal lines indicate these connections and will exist at multiple layers in the communications hierarchy. A more complete representation could include a connection for each layer of encapsulation shown in Figure 1. The dash-dot line at the IP layer serves to indicate that while a direct IP connection appears to exist between the edge routers, there is in fact, no such connection. Instead, packets are conveyed by the ATM layer, which must be taken into account in any performance investigation. The dashed vertical lines are used to establish the dependency relationship between these layers within each ME.

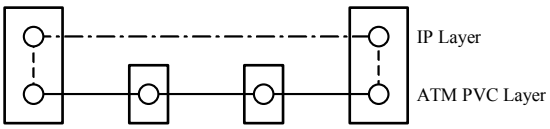

Fig. 19. Data model representation of the ATM/IP configuration shown in Figure 2. The large boxes represent the IP edge routers with ATM interfaces. The smaller boxes are individual ATM switches that are part of a larger ATM infrastructure that is not shown.

We can extend this model by including more of the network on either side of the edge routers as shown in Figure 20. (We drop the boxes for the ME and instead note that the ME's termination points are contained in a vertical slice.) Here we include collected data from associated measurements taken at the different layers and shown as small rectangles attached to their monitoring source points.

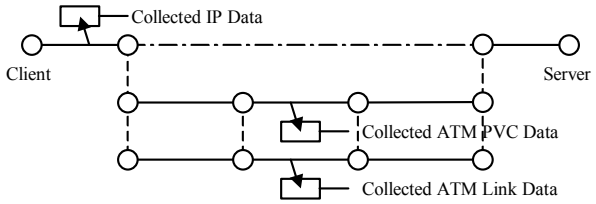

Fig. 20. Adding a client and server IP node to the model in the previous figure. Note how the ATM layer is disconnected as far as the client/server relationship is concerned, but is used as a framework to store ATM statistics. The lowest layer now represents the physical medium that carries the ATM PVC, which in turn supports the IP connectivity.

Also in Figure 20 we further extend the model to include the physical link connection that carries the PVC. Since multiple ATM PVCs may exist on the same physical connection, these should also be shown at the middle PVC layer, with the related connections tied to the same link below. This will serve to relate the seemingly independent flows at the PVC layer to the common fabric used to carry the aggregate traffic, and will be used in the subsequent example. This lower level will have its own aggregate statistics and configuration details. In fact, this is how we can represent the possibility that each ATM link leg may have a different link capacity and utilization.

In turn, the physical link may not be truly physical, but a logical component of an underlying infrastructure. For example, it may exist as a SONET connection riding over a DWDM lambda on a fiber that is only one of many in a sheath of fibers, perhaps sharing a common conduit. The model can serve to represent as much of the physical world as is practical and can be useful in fault isolation as well as performance investigations.

V. ILLUSTRATIVE EXAMPLES – PART 2

E. Relating User Performance across Parallel Universes

In this example we use a combination of configuration, topological and multilayer information to debug a problem of poor user-layer performance in an otherwise properly sized network. We will use the data modeling technique described above and slowly build up the model to define the points where the resolving measurements are taken and related. Our goal is to orchestrate the sets of measurements from operational networks such as in [10] coupled with triggers of performance anomalies, such as those described in [11] to enable the automatic detection and causal analysis of performance problems.

The initial indicator of this class of performance problems is generally a complaint that expected application completion times are not being met. Simple measurements from the application, such as FTP transfer times that differ from previous experience, can indicate such a problem. Figure 21 shows this high level measurement and its position in the data model.

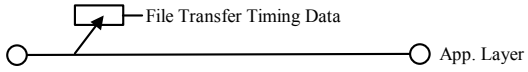

Fig. 21. End-to-end application model. Basic application timing data indicate a high-level problem.

The next step is to isolate the possibility of contention from neighboring applications that may be competing for available resources. Examining the network layer indicated that the only data traffic present was from the application in question. The user traffic appeared bursty in nature; not necessarily unusual, but perhaps atypical for a file transfer. These regular periods of inactivity lasted large fractions of a second and are shown in Figure 22.

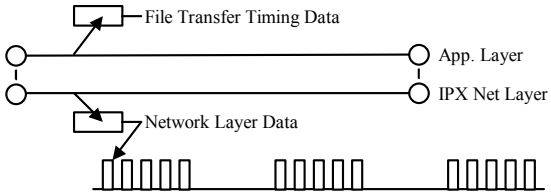

Fig. 22. Extending the model to include the network layer. Captured data indicates no competing traffic but bursty behavior.

Bandwidth limitations at lower layers can restrict the amount of bandwidth available to a user, thereby affecting application behavior. In the configuration shown in Figure 23, either the DLCI can have limits established on the allowable steady state and burst rates, or the entire T1 can become saturated. Neither of these conditions held.

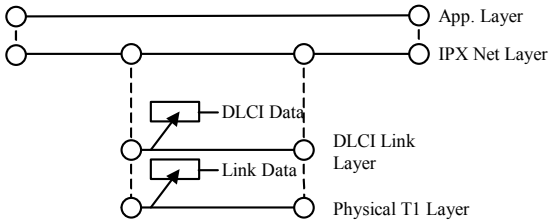

Fig. 23. Collected data at the DLCI link layer and the physical T1 show data rates are not being throttled by allocated bandwidth limitations.

Protocol analysis indicated that most of the transmitted packets were, in fact, retransmissions, though all packets were being acknowledged. We had no indication of loss anywhere in the measured data. While this explained part of the user's low effective data rate, the gaps in the stream as well as the cause of the retransmissions were still problematic.

The pattern of the retransmissions was also unusual and would vary from packet burst to packet burst. An initial burst would contain a normal sequence of packets without any retransmissions. After the gap, we found each packet retransmitted twice. After the next gap, each packet would be resent 3 times. This continued until sequences of 8 duplicates were transmitted. In one session, 432 packets were used to send the 73 packets needed for the transfer.

We reexamined the DLCI layer, this time adding in the neighboring traffic of flows carried on the same physical T1. These are shown on the parallelogram extending out from the main traffic, marked 'Disturbed Traffic' in Figure 24. Note how each additional DLCI is connected to the lower Physical T1 layer indicating the coupling between the flows. While only 3 additional DLCIs are illustrated, there were approximately 20 in use on the physical link.

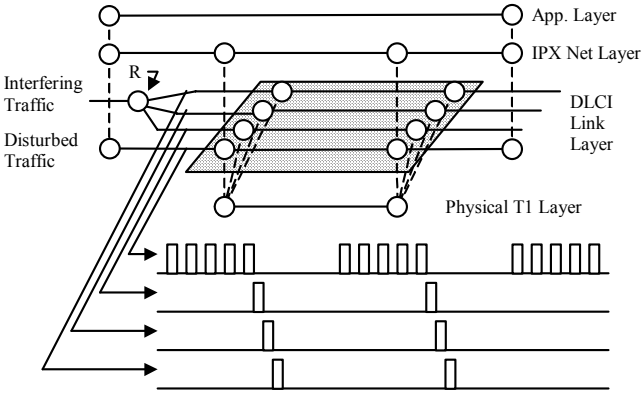

Fig. 24. DLCIs on a common T1 shown on a common plane. The figure emphasizes the neighboring behavior that must be considered.

This neighboring activity was extremely low in bandwidth. However, it exhibited the same periodic characteristic of the main traffic, neatly appearing in the gaps as shown in Figure 24. The traffic did not completely fill the gaps, but was transmitted essentially back to back across successive DLCIs.

By examining the source and destination address spaces in use by the separate flows, we reconstructed the remainder of the layered topology shown at the left of Figure 24, near the marking 'Interfering Traffic', including the router 'R' responsible for regenerating resource availability broadcast packets. We now had sufficient information to complete the puzzle.

The broadcast traffic originated from a single source downstream of router 'R', and was replicated out to multiple subnets for transmission to clients across individual and supposedly independent DLCIs. Unfortunately, they were all carried by the same T1 before eventually diverging to their final destination. The router was extremely efficient in replicating each broadcast packet for the 20 subnets, placing each in the same output queue for the single outbound T1. This could naturally be done faster than incoming packets could possibly arrive and reach the same queue. Therefore

incoming packets would not be fairly interspersed in the output queue, but would be delayed behind a long sequence of packets that had instantaneously 'arrived'. All of this traffic was contained in a set of subnets that should have been entirely independent of our original application, which had the misfortune to be connected across the same physical T1 where these universes collided.

To be fair, the instantaneous burst of packets was not entirely at fault for the disruption to the file transfer. The 20 replicated broadcast packets did not completely fill the file transfer gaps. However, the delay was sufficient to undermine the state machine behind the IPX file transfer, causing it to generate the peculiar duplicate packet sequences. We suspect the delays were just on the wrong side of the IPX timeout window, and the protocol was not robust enough to accommodate it.

In summary, these difficulties were eventually traced to traffic on parallel, independent paths that were destructively interfering with each other, exacerbated by the rapid amplification of packets at the router's internal processing speed.

VI. CONCLUSION

The examples shown are meant to reinforce the concept that complete performance investigations require the inclusion of information gathered from a variety of sources. Representing and automating some of the techniques employed herein leads us to the modeling of network relationships and performance data using a data model such as that described in the M.3100. An appropriate model can represent the complex relationships described here and can be used to support a programming environment to explore these issues.

We feel that this is the necessary next step in performance analysis, and that coordinated collection, storage, modeling and analysis is needed to handle the more complex problems that will plague the Internet. This is an area of ongoing research.

REFERENCES

[1] C. Chen, M. Cochinwala, M. Mesiti, C. Petrone, M. Pucci, S. Samtani, P. Santa, Internet traffic warehouse, Proc. ACM SIGMOD, Dallas TX, May 2000

[2] J. Jerkins, J. Monroe, M. Pucci and J. Wang, Carrying internet traffic over frame relay: frame and call level traffic analyses, ICC 1999, Vancouver.

[3] J. Jerkins, J. Monroe and J. Wang, A measurement analysis of internet traffic over frame relay, Performance Evaluation Review, Vol. 26, No. 6, August 1999.

[4] S. Narain,, A. Shareef and M. Rangadurai, Diagnosing configuration errors in virtual private networks. Proceedings of IEEE International Communications Conference, Helsinki, Finland, 2001.

[5] A. Valencia, M. Littlewood and T. Kolar, RFC 2341, Layer two forwarding L2F, May 1998.

[6] http://www.nlanr.net/NA/Learn/packetsizes.html

[7] http://www.caida.org/analysis/AIX/plen_hist/

[8] B. Viken, Passive monitoring of internet traffic at Supercomputing '98, Proceedings of EUNICE '99, Barcelona, Spain, Sept. 1999.

[9] ITU-T Recommendation M.3100 Generic Network Information Model, 1995.

[10] A. J. McGregor, H-W Braun, Automated event detection for active measurement systems, Passive and Active Measurements Workshop, Amsterdam, 2001.

[11] F. Georgatos et al., Providing active measurements as a regular service for ISP's, Passive and Active Measurements Workshop, Amsterdam, 2001.