Appeared in Proceedings of WORDS '94

Deterministic Characterization and Network Utilizations for Several Distributed Real-time Applications

Edward W. Knightly
UC Berkeley

Robert F. Mines
Sandia National Labs

Hui Zhang
Lawrence Berkeley Lab

Abstract

In this paper, we investigate several distributed real-time applications. The applications are real-time in that they require per-connection end-to-end performance bounds. These bounds, in terms of throughput, delay, and delay-jitter, are provided by the network via two mechanisms: admission control to limit access to the network and priority scheduling to enforce the guarantees. Within this frame-work, we perform a deterministic analysis on the applications and investigate, via the admission control criteria, the resulting utilization of the network. A deterministic analysis must analyze the worst-case properties of the sources in order to provide absolute bounds on throughput and delay. Three distributed real-time applications are analyzed in this paper: a video conferencing tool, a tool for combustion modeling using distributed computing, and an MPEG video archival system. Each has minimum performance requirements that must be provided by the network. By investigating these applications, we provide insights to the traffic characteristics and achievable network utilizations for practical real-time loads.

1 Introduction

Emerging technologies for high speed networks are delivering the means to provide performance guarantees to a wide variety of applications. Such applications can have vastly different traffic characteristics and quality-of-service (QoS) requirements that must be supported by the network. Integrated services networks thus have the challenge of providing per-connection end-to-end real-time service while still achieving a high network utilization. The network provides the real-time service via two components: admission control and scheduling. The admission control mechanism admits new connections only if sufficient resources are available to satisfy the new connection's QoS while not degrading that of existing connections. The scheduling algorithm must then enforce the guarantees promised by the admission control algorithm. This is achieved with with a combination of priority queueing, rate-control (policing), and delay-jitter control.

The focus of this paper is to investigate real applications that utilize such a service. The first component to this analysis is application characterization. Since we limit the analysis presented here to *deterministic* throughput and delay QoS guarantees, the traffic source characterization must be deterministic. For example, to provide a deterministic delay bound, we require knowledge of the maximum number of packets that may be transmitted in any interval of length I.

Once individual sources are characterized, we use the admission control criteria and scheduling policies described in [9] and [8] to provide end-to-end per-connection performance guarantees. Within this framework, we investigate the resulting utilization of the network for various performance bounds. The results have implications in terms of achievable utilizations for real-time traffic requiring deterministic performance guarantees from the network. Moreover, we note that these utilizations are for deterministic traffic only – unused capacity may still be utilized by applications requiring *probabilistic* performance bounds or by best-effort applications that do not explicitly reserve network resources (see [3] for a discussion of these services).

Thus, we provide insight to the traffic characteristics and multiplexing behavior of practical real-time loads. This is achieved by analyzing three applications that are currently under experimentation on a real-time network testbed (see [5] for details on the testbed). The first application is a video conferencing tool based on DAVE (Distributed Audio Video Environment) [7]. DAVE is an object-oriented plug-and-play multimedia programming environment designed to provide an extensible, media-independent model for distributed multimedia. The second application is a combustion modeling tool that uses distributed

[1] This work was supported by the U.S. Department of Energy, DOE, under contract DE-AC-04-94AL85000.

computing. This application uses POET (Parallel Object-oriented Environment and Toolkit) [6] to distribute its immense needs for computing resources across the network. The final application is an MPEG video archival system. This application, also based on DAVE, archives and distributes audio-visual presentations either by multicasting a pre-scheduled presentation to a group of users, or by allowing a user to check out a presentation for interactive viewing on a workstation.

The remainder of this paper is organized as follows. In Section 2 we present a brief review of the analysis techniques used to provide deterministic performance bounds. Next, in Section 3 we describe the three test applications and analyze their deterministic traffic characteristics and the resulting network utilizations when the sources are multiplexed. Finally, in Section 4, we describe future work and conclude.

2 Admission Control for Deterministic Real-time Traffic

This section reviews the analysis techniques in [9, 1] which are used to provide performance bounds for deterministic traffic beyond those achieved with peak rate reservation. That is, even if the sum of the peak rates of all connections is greater than the link speed, deterministic delay bounds can still be provided.

In order to deterministically bound the maximum delay incurred by an arriving packet, a deterministic constraint is needed on each traffic source that requires deterministic service. In [3], traffic source j is deterministically constrained with the $(Xmin, Xave, I, S)_j$ model. In this model, source j is restricted to a minimum packet spacings $Xmin_j$, a maximum packet size of S_j, and a worst case average packet spacing of $Xave_j$. The worst case average spacing constraint requires that over *every* interval of length I_j, no more than $I_j/Xave_j$ packets are sent. With this model, a worst case arrival pattern may be constructed in which a source sends at its peak rate $(1/Xmin_j)$ until it is forced to stop transmitting because of its $I_j/Xave_j$ constraint.

With knowledge of the worst case model of the individual source, a worst-case arrival pattern may be constructed. This worst-case arrival pattern, called a traffic constraint function $b_j(u)$, denotes the maximum number of bits that can arrive from source j by time u. Since the worst case traffic of multiplexed connections is less than the sum of the worst cases of individual connections, deterministic delay bounds may be derived. This is illustrated in Figure 1 which describes the operation of the multiplexer. The horizontal axis is time and the vertical axis is bits. The upper curve represents the total number of bits that have arrived in the queue by time t and the lower curve represents the total number of bits transmitted by time t. The difference between the two curves is the number of bits currently in the queue, or the *backlog* function. When the backlog function returns to zero (the two curves meet), there are no bits in the queue and thus a busy period has ended. The key to this analysis is that if the upper arrival curve is a worst-case arrival curve (i.e., the sum of individual constraint functions), then the maximum delay of any packet can be expressed as a function of the two curves. For example, the following two observations hold: the maximum busy period provides an upper bound on delay for any work-conserving server; the maximum backlog divided by the link speed provides an upper bound on delay for a FCFS server.

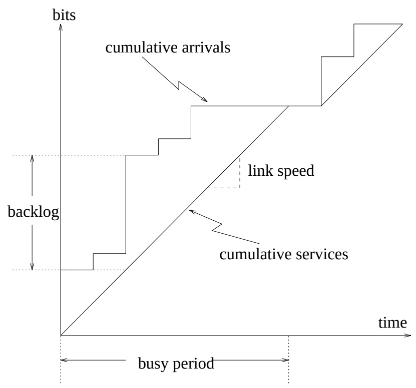


Figure 1: Concepts: Delay, Backlog, and Busy Period

For example, the $(Xmin, Xave, I, S)$ model has a constraint function given by:

$$b(u) = (min(\lceil \frac{u \ mod \ I}{Xmin} \rceil, \lceil \frac{I}{Xave} \rceil) + \lceil \frac{u}{I} \rceil \lceil \frac{I}{Xave} \rceil)S. \quad (1)$$

Theorem 1 of [9] shows that for a FCFS scheduler with $j = 1, ..., n$ multiplexed connections constrained by their respective function $b_j(u)$, a link speed l, and a maximum packet size $\overline{S}$, the resulting delay bound for all connections is given by

$$d = \frac{1}{l} \max_{\forall u \geq 0} \{ \sum_{j=1}^{n} b_j(u) - l \cdot u + \overline{S} \}. \quad (2)$$

In the next section, we characterize a source with the $(Xmin, Xave, I, S)$ for various choices of I. That is, the client's choice of I will determine the value of $Xave$ for a given source. It will be shown that the choice of I

can have a considerable effect on the achievable delay bound and the resulting network utilization.

3 Application Characterizations and Network Utilizations

This section provides a description of three actual real-time traffic sources: a video conferencing tool, a tool for combustion modeling using distributed computing, and an MPEG video archival system. First, the applications are analyzed in terms of their deterministic properties (required in providing deterministic performance bounds). Next, for the given traffic characterization, Equation (2) is used to determine how many connections may be multiplexed in a switch for a given delay constraint. The number of admissible connections determines the utilization of the switches and thus the network.

Both the video conferencing tool and the MPEG video archival system are built on top of DAVE, a model for building distributed multimedia applications. DAVE provides a *plug-and-play* programming environment that is extensible, interoperable, and media independent. The plug-and-play paradigm provides a simple programming environment in which application developers can create multimedia meta-devices on distributed machines running the DAVE daemons. These devices are then plugged together to create an application network that spans the physical network. For example, to send live video with a guaranteed performance connection, a sending application connects a camera device (with the hardware details hidden) to a compression filter, which in turn is connected to a guaranteed performance network device that supports the Tenet Real-time Protocol Suite [2]. The receiving application also uses the plug-and-play environment connecting a receiving network device to a decompression filter and finally to a display window on the receiving host. DAVE supports extensibility by providing a well defined interface between the system kernel and the meta-devices. Extensibility is further supported with the use of dynamic binding and a meta-device class hierarchy which provides the foundation for all meta-devices in DAVE. This provides a simple mechanism for easily adding additional device types to the system. Finally, DAVE is media independent in that there is no data type checking in the system. DAVE uses a message passing system and message class hierarchy to give the user the ability to create new data types as necessary requiring only that every message type inherits from the message base class. In addition, at the connection level, DAVE provides connection management capabilities for developing both peer-to-peer and client-server applications. Thus, DAVE provides a framework for building a wide range of distributed applications such as media servers, video conferencing tools, and remote process control tools.

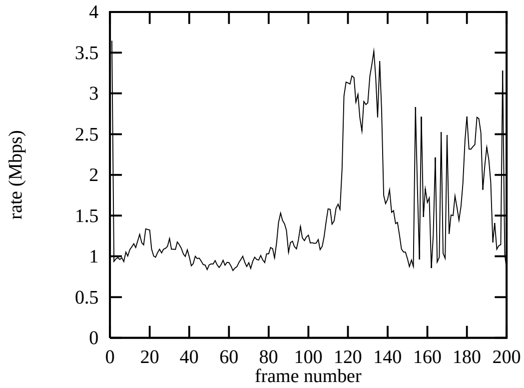


Figure 2: DAVE Video Conference Trace

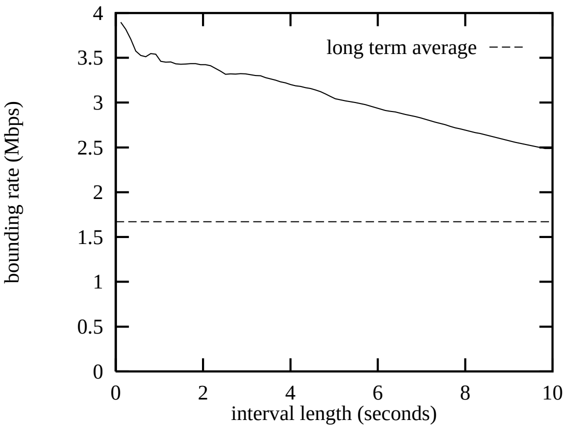
3.1 Video Conferencing

The first real-time application we consider is a video conferencing tool. This application requires performance bounds from the network since if the end-to-end delay becomes too large, if the loss rate becomes too high, or if the throughput is too low, the quality of the conference will degrade to a point where it is worthless.

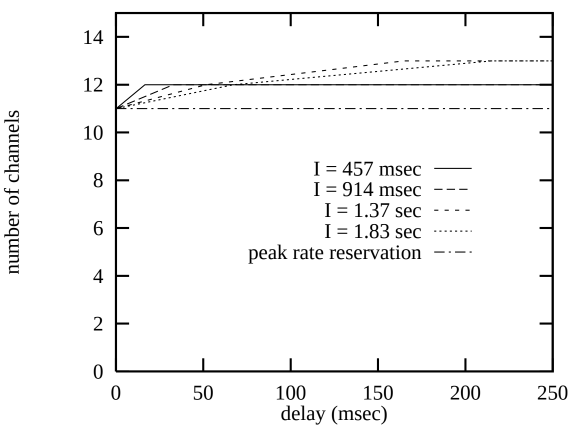
A video conferencing tool in use at Sandia and Berkeley is based on DAVE. The tool uses a software compression algorithm being developed within the DAVE system. This algorithm was designed to provide fast, high quality, medium bandwidth video compression for UNIX workstations. Temporal consistencies in the video stream are exploited by partitioning each frame into cells and then applying a differencing algorithm between frames. Next, the data is encoded using an adaptive Huffman encoding scheme, and finally, it is run-length encoded. Although this algorithm does both inter-frame and intra-frame compression, both JPEG and MPEG *hardware* coders are also in use for their better compression ratios (section 3.3 investigates an MPEG application).

Figure 2 shows a segment of an output trace (Mbps vs. frame number) of a DAVE video conference. The video was compressed at approximately 8.75 frames per second so that rate of Figure 2 is simply the frame size multiplied by the frame rate. The trace shown represents 23 seconds of a 7 minute trace.

In the $(Xmin, Xave, I, Smax)$ model, $Xave$ may



(a)



(b)

Figure 3: DAVE Traffic Characterization and Maximum Number of Admissible Channels

be viewed as a function of I where for a given interval length I, $I/Xave$ is the maximum number of packets that may be transmitted in any interval of length I. Thus, $Xave$ is a maximum of the average packet spacing over every interval of length I. Within this context, for a given I, $Smax/Xave$ may be viewed as the maximum average *rate* over any interval of length I. Figure 3(a) shows the maximum average rate (in Mbps) as a function of interval length I (in seconds). The maximum value of the curve is 3.90 Mbps which corresponds to the source's peak rate which is equivalent to the maximum average rate over an interval of $I =$ one frame time $= 1/8.75$ seconds. As expected, as the averaging interval I increases, the maximum average rate decreases (though not monotonically). Thus, for the $(Xmin, Xave, I, Smax)$ characterization, any $(I, Smax/Xave)$ pair from the curve along with $Xmin$ (from $Xave$ for $I = 1$ on the curve) results in a valid deterministic characterization of the source.

For comparison, Figure 3(a) also shows the long term average bandwidth. This is simply the total number of bits transmitted during the video conference divided by the total time of the conference. The resulting average bandwidth is 1.67 Mbps. There are two remaining points regarding Figure 3(a). First, since the burst ratio or peak-to-average-rate ratio is 2.34, even peak-rate reservation yields an impressive average network utilization of 42.7%. This utilization should be considered significantly high since it is the utilization only for deterministic service — best-effort and statistical service can always utilize remaining resources. Second, of interest is the rate at which the maximum average rate approaches the long term average rate. If the maximum average rate decreases slowly, this indicates that once a source begins sending at its peak rate, it continues to send at high rates for a considerable length of time. If this is the case, as it is in Figure 3(a), then the admission control algorithm (Equation (2)) indicates that the number of admissible connections will not be substantially greater than the number accepted with a peak-rate allocation scheme.

Figure 3(b) shows the maximum number of channels that may be accepted into the network for a give delay bound. Specifically, for a static priority scheduler (see [8]) and a link speed of 45 Mbps, the curves show how many connections with a given $(Xmin, Xave, I, Smax)$ characterization can be multiplexed while satisfying a deterministic delay constraint. For example, the figure shows that with peak rate reservation, 11 connections can be multiplexed for any delay bound above a small threshold (the sum of 10 cell service times). Moreover, the $I = 914$ msec curve shows that if the $(Xmin, Xave, I, Smax)$ curve is chosen with $I = 914$ msec, 12 connections may be accepted for delay bounds above 33 msec. A 13th connection is accepted only with a delay bound greater than 165 msec for $I = 1.37$ sec. Clearly, since these delays represent the queueing delay of only one switch within a network and propagation delays must also be considered, a user will most likely request a delay bound smaller than 165 msec and thus 12 connections will be the practical limit. Thus, the analysis techniques of [9] provide a 9% improvement in the number admissible channels over peak-rate reservation.

A further observation from the collection of many traces (not shown here) is the significant differences in the time function, deterministic characterization, autocorrelation function, and distribution of the source based on such coarse-grain human factors as the distance the speaker is from the camera, the dynam-

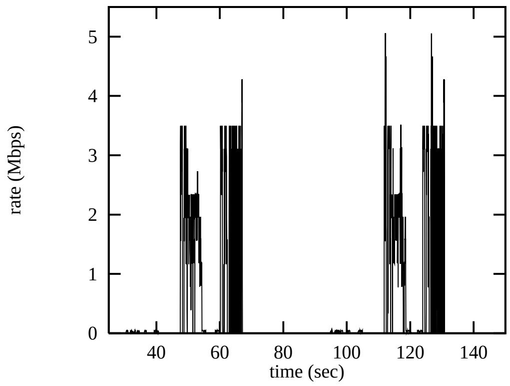


Figure 4: POET trace

ics and mannerisms of the speaker, and the ambient background video noise (e.g., caused by people in the background watching or walking by). This rough observation points immediately to the difficulty of the stochastic modeling problem and indicates that providing deterministic service is promising – especially if the achievable network utilization is high.

3.2 Distributed computing

A second application involves using distributed computing for turbulent combustion modeling. Since accurate modeling of the coupling between the reactive and diffusive processes on a small scale requires vast computational resources, distributing the computing resources throughout the network better facilitates the analysis of this problem.

Parallel Object-oriented Environment and Toolkit (POET), an ongoing project at Sandia, provides the mechanisms for development of applications that require distributed and massively parallel computation. Other existing tools for parallel and distributed software development generally fall into two categories:

1. high-level tools and compilers that hide the parallelization details; such tools are easier to use but often hide the pitfalls that lead to bottlenecks.

2. low-level tools for message passing; though such tools can create scalable distributed code, they are often difficult to use since they require explicit knowledge of the underlying algorithms and software.

POET uses an approach similar to the X Toolkit approach which enables both high and low level control of the computational methods. The user's program instantiates and configures parallel objects that suit the model requirements and then calls a main loop routine. This routine handles the parallel object interactions and associated data transports, and returns when all objects have run to completion.

A time trace of a POET combustion modeling application is shown in Figure 4. This figure represents the sending bandwidth as a function of time for one of the workstations of this distributed application. The peak rate is 5.1 Mbps and the long-term average rate is 429 kbps for a burst ratio of 11.8 to 1.

Figure 5(a) shows the maximum average rate vs. interval length for the POET trace. As expected from visual inspection of the trace in Figure 4 (though this was a segment of the entire trace) the maximum average rate gradually decreases from the peak rate to the long term average rate. Further, although the curve retreats from the peak rate fairly rapidly (Figure 4 demonstrates that the rate of 5.1 Mbps is not sustained), it flattens out at a rate of just above 3 Mbps. Again, the previous figure shows the source nearly sustaining a rate above 3 Mbps for intervals of up to 5 seconds. However, despite this source's bursty behavior, Figure 5(b) shows that the bounds of Equation (2) allow a substantial improvement beyond peak-rate reservation. For example, while only 8 connections connections can be multiplexed using peak-rate allocation (floor(45/5.1)), depending on the requested delay bound and the source's choice of I, up to 13 connections can be multiplexed – a 63% increase. The number of acceptable connections is especially important in this scenario since the number of workstations involved in a typical combustion modeling application is on the order of 10 to 15. A single refused connection means that the entire application cannot run with a guaranteed delay bound.

While Figure 4 showed that the POET time trace is nearly periodic with a period of approximately 64 seconds, not shown is the fact that traces from different workstations are highly correlated with each other. With the high interaction between workstations (one cannot continue until it receives information from another), network delays will seriously degrade the performance of the POET application. Further, unlike the video-conferencing application, all information must be received so that if data is lost due to buffer overflows, it must be retransmitted. Therefore, this application requires low average delay and reliable transport. Unfortunately, because of the source's high burst ratio and a maximum average rate curve that does not decrease sharply enough, deterministic guarantees for this source are costly to the network. For 13 multiplexed connections and a delay bound of 120 msec, the average utilization of the network for deterministic traffic is 12.4% (again, the remaining re-

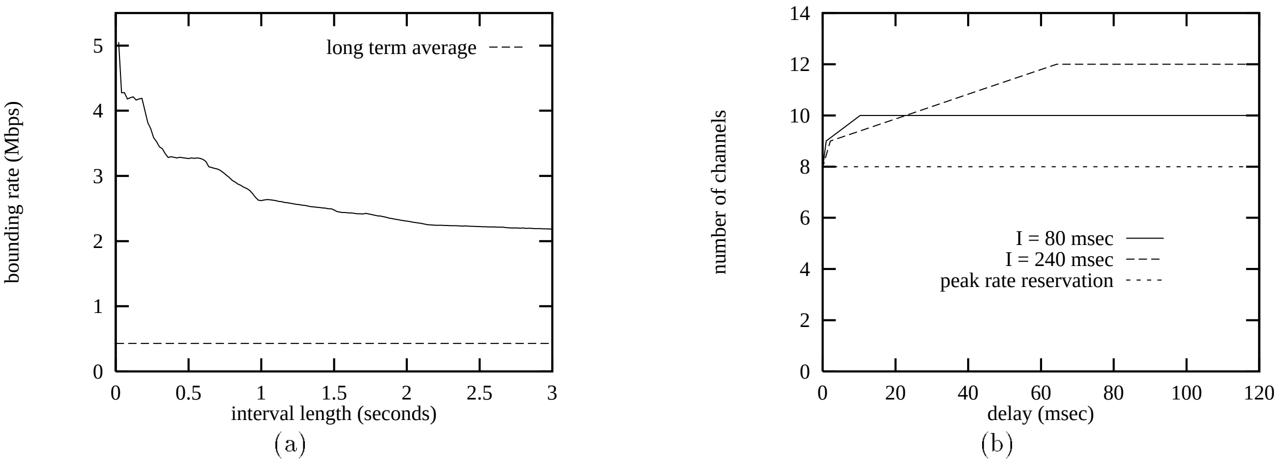


Figure 5: POET Traffic Characterization and Maximum Number of Admissible Channels

sources may be used by statistical or best effort traffic). One may be tempted to improve on the 12.4% utilization by giving the traffic probabilistic bounds on delay rather than deterministic bounds. However, most analysis techniques assume statistical independence of sources which is clearly not the case here. Indeed, each of the POET sources is synchronized since the model of processing is a computing period on all nodes followed by several seconds of massive message exchange.

3.3 MPEG Video Archival System

The final application considered in this paper is a production audio-video archival and distribution system under development at Sandia. This system is being built to archive and play back audio-video presentations with two modes of operation. In the first mode, the audio-video will be multicast at a scheduled time when all users at Sandia may view the presentation. In the second mode, a user will check out a presentation which is then unicast to the user's workstation where it may be viewed and displayed interactively with fast-forward and rewind capabilities. Since the archival system stores and transmits MPEG encoded presentations, receiving workstations must perform MPEG decompression. This is done in hardware or software, depending on the workstation's hardware capabilities.

The video server provides storage and access to several hundred audio-video presentations and is designed to support at least 10 concurrent audio-video streams. Movies stored on a hierarchical storage space using RAID disk arrays and an 8mm jukebox will be available on demand to the users. The server has the capability of caching approximately two hours of video information into RAM memory. Network access to the server is through FDDI, ethernet, and ATM interfaces that will be connected to Sandia's internal network as well as to the Xunet testbed (see [5]).

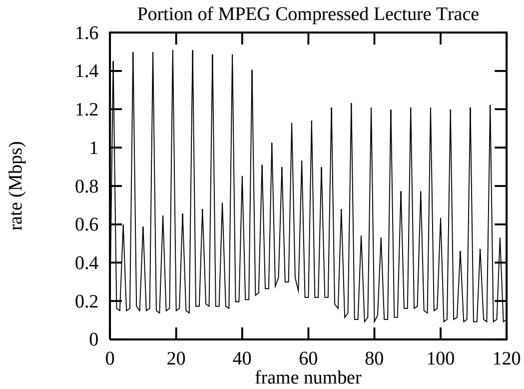


Figure 6: MPEG Video Server Trace

The system is designed to support a wide range of desktop workstations. The receiving workstation has the ability to tune into a single channel accessed via software remote control. Typical programs include presentations from guest speakers, directory information, and newscasts. The approximately 2000 target destination machines are widely dispersed between Albuquerque, Livermore, and Berkeley. Since real-time protocols and multicast capabilities are not supported in all of Sandia's switches and routers, tunneling is required. Additionally, because of several slow network links, preliminary experiments are being performed with low bandwidth sources at rates generally less than 300 kbps. Future investigations will utilize the higher bandwidth links by sending larger and higher quality image streams. This work will necessarily address the issue of providing different QoS to different multicast receivers, an issue beyond the scope of this paper.

Figure 6 shows a trace of the rate in Mbps for a recent Sandia talk that is stored on disk. For this trace, the frame rate is 30 fps so that the trace of Figure 6 is also a trace of the frame size times 30. Although in general, the parameters of the generated traffic depend on many factors such as those mentioned in section 3.1, there will certainly be less action and fewer scene changes in a lecture or presentation than in entertainment type video. The general shape of the trace may be explained in terms of the mechanisms used in the MPEG standard. The coder generates three types of frames, I frames that use only *Intraframe* compression, and P and B frames that are transmitted between I frames and use *interframe* compression. While P frames (Predicted frames) are coded based on only past frames, B frames (Bidirectional frames) are coded based on both past and a future frame. With P and B frames, higher compression ratios are achieved since the interframe coding makes use of motion compensation techniques. The trace shown in Figure 6 was generated using a frame pattern of IBBPBB. More details of the MPEG algorithm may be found in [4].

Figure 7(a) shows the maximum average rate vs. interval length for the MPEG source. As shown, the source's peak rate is 1.51 Mbps (from $I = 1$ frame time) and the long-term average rate is 338 kbps (from $I =$ entire sequence) for a burst ratio of 4.5. The fast rate at which the maximum average rate decreases to the long term peak rate is indicative of the desirable multiplexing properties of this source. Indeed, Figure 7(b) shows a remarkable increase over peak-rate reservation. For example, for the $I = 67$ msec specification, 53 connections with the same specification may be multiplexed with a deterministic delay bound of 29 msec. This represents a 40% average network utilization. Additionally, if the source uses the $I = 200$ msec characterization, 76 connections can be multiplexed with a delay bound of at least 120 msec resulting in an average utilization of 57%. Although this delay bound may be too large for an interactive application (again, this is the queueing delay of one switch), it may be feasible for play-back applications such as this video server. One final note is that we have not addressed buffer requirements in this paper. Clearly, as the delay bound increases, the buffer size requirements of the switch also increase.

4 Conclusions

In this paper, we have presented and analyzed three applications under development at Sandia National Laboratories: a video conferencing tool, a tool for combustion modeling using distributed computing, and an MPEG video archival system. Analysis of these traffic sources has provided several insights to the traffic characteristics and switch utilizations for practical real-time loads.

There are two components to achieving high network utilization for deterministic traffic. First, is the peak-to-average rate ratio. For example, with the DAVE video conferencing application's low peak-to-average rate ratio of 2.34, even a peak-rate resource allocation scheme yields a significant utilization of 42.7%. Second, is the rate at which the maximum average rate decreases (with increasing interval length) to the long-term average rate. For example, even though the MPEG source has a higher burst ratio of 4.5 to 1, because of the rapid decrease in the maximum average rate curve, high network utilizations of up to 57% are achievable for this source.

Investigation into the distributed modeling application showed that with the higher peak-to-average rate ratio of 11.8 combined with the long burst lengths, utilizations of only 12.4% were achievable. However, because of the synchronization of the sources involved in the application, we postulate that even with a statistical performance guarantee rather than a deterministic one, high network utilization will be difficult to achieve.

Ongoing and future work involves many aspects of both the real-time applications and the network. For the applications, more analysis is needed using mean opinion scores in order to better determine the loss and delay requirements for different real-time applications. Additionally, we are investigating new traffic models to provide insights to the problem of providing statistical performance guarantees.

References

[1] R. Cruz. A calculus for network delay, part I : Network elements in isolation. *IEEE Transaction of Information Theory*, 37(1):114–121, 1991.

[2] D. Ferrari, A. Banerjea, and H. Zhang. Network support for multimedia: a discussion of the Tenet approach. *Computer Networks and ISDN Systems*, 26(10):1267–80, July 1994.

[3] D. Ferrari and D. Verma. A scheme for real-time channel establishment in wide-area networks. *IEEE Journal on Selected Areas in Communications*, 8(3):368–379, April 1990.

[4] D. Le Gall. MPEG: A video compression standard for multimedia applications. *Communications of the ACM*, 34(4):46–58, April 1991.

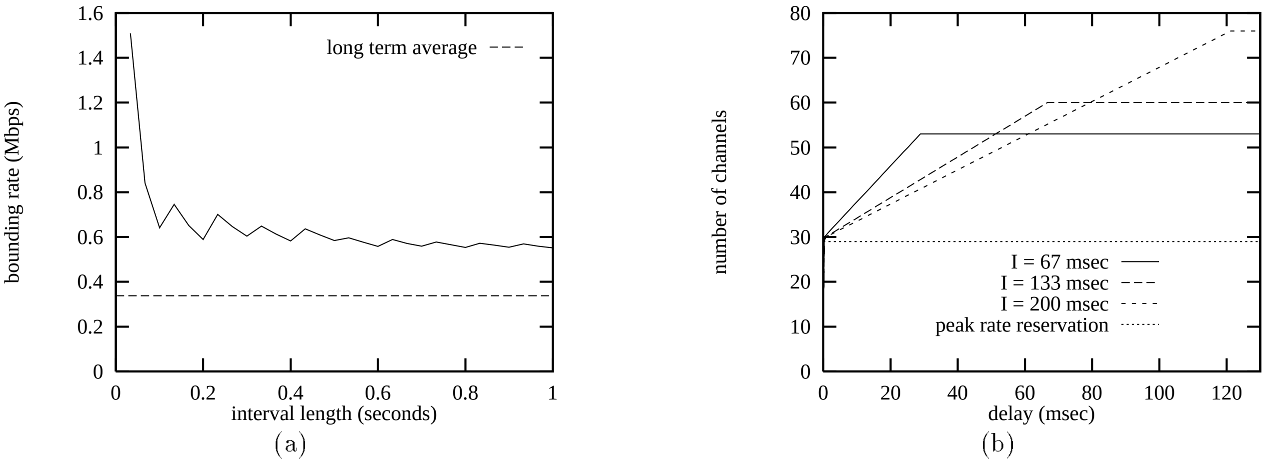


Figure 7: MPEG Traffic Characterization and Maximum Number of Admissible Channels

[5] E. Knightly and R. Mines. Test applications for a heterogeneous real-time network testbed. In *1994 International Conference on Computer Communications and Networks (ICCCN '94)*, pages 183–189, San Francisco, CA, September 1994.

[6] J. F. Macfarlane, R. C. Armstrong, R. E. Cline Jr., and M. L. Koszykowski. Application of parallel object-oriented environment and toolkit (POET) to combustion problems. Technical report, Sandia National Laboratories, Livermore, CA, September 1992.

[7] R. F. Mines, J. Friesen, and C. Yang. DAVE: An object-oriented multimedia programming environment. In *ACM Multimedia 1994*, San Francisco, CA, October 1994.

[8] H. Zhang and D. Ferrari. Rate-controlled static priority queueing. In *Proceedings of IEEE INFOCOM '93*, pages 227–236, San Francisco, CA, March 1992.

[9] H. Zhang and D. Ferrari. Improving utilization for deterministic service in multimedia communication. In *1994 International Conference on Multimedia Computing and Systems*, pages 295–304, Boston, MA, May 1994.