Deadline Based Network Resource Management

Johnny W. Wong  
Yanni Ellen Liu

Department of Computer Science, University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
jwwong,y5liu@uwaterloo.ca  Fax: 519-885-1208

Abstract—A novel approach to supporting real time applications in packet-switched networks is proposed. The key element of this approach is the use of a tuple (size, deadline) to characterize the performance requirements of a document. This tuple is mapped to deadline information at the network layer, which is carried in packets and used by routers for channel scheduling purposes. A new deadline based scheduling algorithm is developed. Simulation results show that this algorithm is superior to FCFS with respect to the percentage of documents that are delivered on time. Our scheduling algorithm is also efficient as far as implementation is concerned.

I. INTRODUCTION

In recent years, we have seen increased interest in the transport of real time data over a network. Examples include audio and video data in a video conference, state update in a distributed multi-user game, and urgent data in an alarm application. To ensure timely delivery of real time data, quality of service (QoS) support at the transport network is required.

Traditionally, the Internet provides best effort service; its delay performance is good when the traffic is light. However, it does not scale well as the level of traffic increases. To provide QoS support, several strategies for network resource management have been developed. These include fair queueing and buffer management at routers [5], [8], integrated services and resource reservation [2], [11], and differentiated services [1]. These strategies are based on the notion of a flow, and QoS support is provided on a per flow basis. However, they may not be appropriate for certain types of real time data transfer. An example is the timely delivery of a small document. In this case, it is difficult to specify a flow with given bandwidth requirements because data will not be presented to the network for a significant period of time.

In this paper, we propose a novel approach to managing network resources in order to support real time document delivery applications. In our investigation, the term document is used to mean a unit of data transfer that has an associated delivery deadline. For the case of a video conferencing application, an audio or video frame could be considered as a document. Each frame should be delivered no later than its deadline in order to ensure the interactive nature of the conversation as well as good presentation quality at the receiver. For some other applications, a document has the same meaning as a file. The size of the file may range from very small, e.g., an urgent message in an alarm application, to very large, e.g., a business document in a business-to-business electronic commerce transaction. In this paper, only applications with soft real time requirements are considered. This means that it is acceptable if most, if not all, documents are delivered on time.

At the network layer, our focus is on channel scheduling algorithms that are designed to maximize the percentage of documents delivered on time. Of particular interest are deadline based scheduling algorithms. Examples of such algorithms can be found in [4]. These algorithms are used as the point of departure for our investigation.

This paper is organized as follows. In section II, our approach to supporting real time document delivery is described. This involves the specification of a (size, deadline) pair for each document, and the mapping of this specification to information needed by the network layer for channel scheduling purposes. Our deadline based scheduling algorithm is described in section III. A simulation model to evaluate its effectiveness is presented in section IV. Results from the simulation are discussed in section V. Finally, section VI concludes the study.

II. REAL TIME DOCUMENT DELIVERY

We propose a deadline based approach to supporting real time document delivery and to managing network resources. In this paper, only applications with soft real time requirements are considered. This means that it is acceptable if most, if not all, documents are delivered on time. Of particular interest are deadline based scheduling algorithms. Examples of such algorithms can be found in [4]. To support such applications, we define a scheduling algorithm that maps each ADU to a flow at the transport network. In this way, the network can provide QoS support for each ADU in the network.

A. Network Architecture

The network architecture of interest is depicted in Fig. 1. Consider the transmission of an ADU from sender to receiver. This ADU is passed from the sending application to the transport layer, together with its size and deadline (denoted by $s$ and $d$ respectively). At the transport layer, a maximum segment size $M$ is defined, and an ADU is segmented into $m \geq 1$ transport segments. When these $m$ segments are passed to the network layer, segmentation may again be performed since a maximum packet size $P$ is defined at the network layer. For simplicity, we assume that there is an one-to-one mapping between transport segment and packet.
Let $d_i$ be the deadline of packet $i$. Also let $r_i$ and $r_{pi}$ be the processing delays at the transport and network layers at the receiver. We have

$$d_i = d - r_i - r_{pi}$$

for all $i$. In our proposed approach, the deadline $d_i$ is carried in packet $i$, $1 \leq i \leq m$.

B. Network Layer Issues

Our goal is to come up with network resource management strategies such that the percentage of ADU’s that are delivered on-time is maximized. The key element of these strategies is deadline scheduling. Since each packet has a delivery deadline, deadline based scheduling algorithms [4] are considered. In order for routers along a path to make use of the deadline information carried in a packet, clock synchronization between these routers and the destination is required. An immediate question is the precision used to specify the deadline. This will have a direct impact on the packet header length. The answer to this question largely depends on the deadline granularity required by the application. One possibility is to use the timeval data structure that is defined for the UNIX system call gettimeofday. It occupies 8 bytes and provides microsecond precision. We believe this precision is sufficient for most real time applications that are of interest to this investigation.

A second issue is the clock synchronization among the sender, the receiver, and the routers along the path. Because we are not targeting hard real time scenarios, clock offsets may be tolerated. Many systems on the Internet today use NTP (network time protocol) to keep their system clocks synchronized with each other and with international standards [9]. If each clock differs from a standard time by at most $\alpha$, then two clocks are at most $2\alpha$ apart. We can therefore design any time-related algorithms to accommodate this offset. As an example, when we compare a local time $t$ with a deadline $d$ (carried by a packet) to see if the packet is late, instead of using the predicate “if $d < t$”, we can use “if $d + 2\alpha < t$”. Conversely, if we want to make sure that a packet is early, instead of using “if $d \geq t$”, we can use “if $d - 2\alpha \geq t$”. We believe this type of compensation is acceptable to most soft real time applications.

III. Deadline Based Channel Scheduling

At the network layer, an important consideration is to deliver as many packets by their deadlines as possible. This will have a positive impact on the percentage of ADU’s that are delivered on time. In this paper, an existing and a new channel scheduling algorithm are evaluated in terms of their effectiveness in delivering ADU’s in a timely manner.

Deadline based scheduling algorithms have been studied extensively in the context of job-shop models [4]. These models are applicable to packet-switched networks. Promising algorithms include “earliest due date”, “operation due date” (end-to-end deadline divided by number of operations), and “slack” where scheduling decisions are based on the time left minus the remaining service time [4].

Motivated by the work reported in [4], we developed a new scheduling algorithm which is based on the ratio $T/H$, where $T$ is the time left (or deadline - current time) and $H$ is the number of hops to destination. $T/H$ is calculated when a packet arrives at a router; it can be viewed as the urgency of a packet. Specifically, a packet with a smaller $T/H$ means that it is more urgent. In terms of implementation, $T$ can readily be calculated because the deadline is carried within a packet and the current time can be obtained by reading the clock at the router. The value of $H$ is readily available also if link state routing [10] is used. In this case, each router periodically computes a spanning tree to all other routers, and the number of hops to each destination can be determined from this spanning tree.

There are two variants of our algorithm: $T/H$ and $T/H$-L:

- $T/H$ - smallest $T/H$ first. Packets with the smallest value of $T/H$ is given the highest priority. This variant may not be very efficient because serving packets in order of their $T/H$ values requires $O(\log n)$ operations for a backlog of $n$ packets.
- $T/H$-L - $T/H$ with threshold. This is a head-of-the-line priority queueing system [7] with 2 queues. A threshold parameter $L$ is defined. A packet is entered into the high priority queue if its $T/H \leq L$, otherwise it is entered into the low priority queue. The advantage of this scheme is simplicity when compared to $T/H$. The time complexity for its implementation is $O(1)$ instead of $O(\log n)$.

For both variants, refinement of algorithm is possible if ADU information is available in each packet. The idea of packets carrying ADU information was proposed in [3] as application layer framing (ALF). In our case, if each packet carries an ADU id, then whenever a packet is already late at a router (before it gets to the destination), all subsequent packets belonging to the same ADU can be downgraded to best effort (by setting their deadline to infinity). This will help increase the chance of other ADU’s being on time. We call this scheme “late ADU downgrading”.

Note that scheduling algorithms that make use of deadlines have also been investigated (see for example [6]). However, in these algorithms, the deadlines are dynamically determined by the routers. This is different from our approach where deadlines are specified by the application.

IV. PERFORMANCE MODEL

In this section, a performance model is developed to evaluate the effectiveness of our scheduling algorithm. It is a discrete event simulator which supports arbitrary network topology and has extensible components, such as traffic models and scheduling algorithms. As mentioned previously, the performance measure of interest is the percentage of ADU’s that are delivered on time.
We first consider the modeling of document delivery from a given sender to a given receiver. This is illustrated in Fig. 2. ADU’s (or documents) are generated at the sender. Each ADU is characterized by the following attributes: size, sender and receiver addresses, deadline and arrival time. Two types of ADU’s are considered: real time and best effort. Best effort ADU’s are characterized by a delivery deadline of infinity. Segmentation of an ADU into packets is performed at the sender before the packets are admitted to the entry node. The processing time at the sender to prepare the packets for transmission is assumed to be negligible.

Packets are routed through the communication network until they reach their exit node. They are then delivered to the receiver where packet re-assembly is performed. For convenience, we assume that the processing time to receive a packet is negligible (or $r_t = r_s = 0$), fixed shortest-path routing is used to route packets from sender to receiver, no packets are lost due to error or buffer overflow, and that packet transmissions are not delayed by any flow or congestion control mechanism.

For a real time ADU, the delivery deadline is modeled as follows. Let $x$ be the end-to-end latency when there is no queueing and no segmentation. Also let $x_y$ be the end-to-end propagation delay, $y$ the size of the ADU, and $c_j$ the capacity of the $j$th channel along the path based on shortest-path routing. Then $x$ can be estimated by $x = x_y + \sum_y y/c_j$. The allowable delay is assumed to be proportional to $x$. Hence, the delivery deadline for the ADU is given by $d = \text{arrival time} + kx$, where $k$ is a “deadline parameter” ($k > 1$). In general, a smaller $k$ means that the ADU has a more urgent deadline.

Since $r_t$ and $r_s$ are assumed to be zero, $d$ is also the deadline for each packet belonging to the ADU (see equation (1)).

Consider now the complete network model. The network topology used in our simulation experiments is depicted in Fig. 3. In choosing the topology, we take into consideration factors such as connectivity, versatility and size. The value shown on each channel is the distance in miles; this information is used to determine the propagation delay. The capacity of each channel is assumed to be 155Mbit/sec.

ADU’s may be generated by applications from any point in the network. We assume that the interarrival time of ADU’s is exponentially distributed, and the aggregate arrival rate is $\lambda$ (in number of ADU’s per second). For each arriving ADU, the entry node and exit node are selected at random. The size of each ADU is assumed to belong to one of two ranges: [500, 1500], and [1500, 500000], in bytes. The first range reflects the sizes of small ADU’s, i.e., one packet per ADU. The proportion of small ADU’s is kept at 25%. ADU size is assumed to be uniformly distributed within each of these two ranges. At the network layer, $P$ is assumed to be 1500 bytes (the maximum ethernet frame size, which is widely used on the Internet).

Finally, all system and router clocks are assumed to be perfectly synchronized. Therefore, the clock compensation scheme discussed in Section II is not included in our model.

V. Simulation Results

Simulation experiments have been performed to study the performance of T/H and T/H-L. We also consider FCFS which is used in traditional router design where best effort service is provided. In terms of implementation, FCFS is more efficient because there is no need to compute $T$ and $H$.

For each experiment, the simulation runs were for a duration of 30 seconds.

A. Effect of ADU Arrival Rate

In our first experiment, we vary the ADU arrival rate $\lambda$ from 300 to 1400 ADU’s per second. The deadline parameter $k$ is selected to 1.5. 50% of the ADU’s generated are of type real time; the other 50% are best effort. For T/H-L, simulation runs were made to determine a good value of $L$, and the value selected is 0.018.

In Fig. 4, we plot the percentage of on-time ADU’s against $\lambda$. The effectiveness of deadline based scheduling is clearly shown by these results. For T/H and T/H-L, the percentage of on-time ADU’s is higher than 90% when $\lambda \leq 1100$. In contrast, for FCFS, this percentage drops from 95% at $\lambda = 300$ to 43% at $\lambda = 1100$. It is also observed that T/H performs slightly better than T/H-L, and that the performance difference between T/H (or T/H-L) and FCFS increases with $\lambda$.

For all three schemes, the performance degrades quickly when $\lambda > 1100$. This is a result of the increased load on the network, leading to longer delays at the routers. Based on this observation, we conclude that as the load increases, some kind of admission control at the application or transport layer is required in order to maintain a high percentage of ADU’s delivered on time. The issue of admission control will be addressed in a future paper.
B. Effect of Deadline Parameter

Our next experiment is to study the effect of the deadline parameter $k$ on performance. We fix $\lambda$ at 1000 ADU’s per second, and vary $k$ from 1.25 to 3.5. The results are shown in Fig. 5. We observe that as $k$ increases, more ADU’s are delivered on-time for all three schemes. However, the performance of T/H and T/H-L is significantly better than that of FCFS.

The results in Fig. 4 and Fig. 5 indicate that the interaction of $\lambda$ and $k$ may be significant in terms of their impact on performance. To investigate this interaction, we use simulation to obtain the various combinations of $\lambda$ and $k$ that would result in 95% of the ADU’s being delivered on time. Our approach is as follows. For each value of $\lambda$ in the range of 300 to 1000, we perform simulation runs for different values of $k$ until a 95% on-time delivery is found. The results are shown in Fig. 6. For T/H, the deadline parameter $k$ needed is 1.035 for $\lambda = 300$, and increases to 1.57 when $\lambda = 1000$. This increase is much smaller than that for FCFS, where the corresponding values of $k$ are 1.46 and 7.5 respectively. T/H-L achieves similar performance to T/H. This is a further confirmation of the effectiveness of our baseline scheduling algorithm.

C. Effect of T/H-L’s Threshold Value

In our third experiment, we investigate the impact of the threshold value $L$ on the performance of T/H-L.

Consider the case of $\lambda = 500$ and $k = 1.5$. The results in Fig. 4 indicate that the percentages of ADU’s delivered on time for T/H and FCFS are 98.77% and 89.85% respectively. For the case of T/H, we found that during the simulation run, the T/H values calculated by the various routers range from about -0.005 to 0.212 (a negative value means that the packet is already late). In our third experiment, the same range is used for the threshold value $L$ in the T/H-L scheme.

In Fig. 7, we plot the percentage of on-time ADU’s against $L$. The performance of T/H and FCFS is not affected by $L$, and is shown by horizontal lines at 98.77% and 89.85% respectively. As expected, we found that the performance of T/H-L falls in between that of T/H and FCFS. We also found that T/H-L achieves its best performance when $L = 0.01875$. This best performance is only about 0.15% worse than that of T/H. More importantly, over the range of $L$ considered, the performance of T/H-L is close to that of T/H. Between these two schemes, T/H has slightly better performance, but T/H-L is significantly more efficient as far as implementation is concerned (O(1) vs. O(log n) for a backlog of n packets). We conclude from this observation that T/H-L is the best scheme among the three schemes considered.

D. Effect of Maximum Packet Size

In our next experiment, we investigate the impact of the maximum packet size $P$ on performance. Our results are based on $\lambda = 1000$ and $k = 1.5$. The percentage of on time ADU’s when $P$ is increased from 1500 to 10500 bytes is shown in Fig. 8. It is observed that the value of $P$ has very little impact on performance.

E. Effect of Late ADU Downgrade

In section III, a “late ADU downgrade” (LAD) scheme was suggested. Under that scheme, once a packet of a particular ADU is detected late by a router, all subsequent packets belonging to the same ADU will be downgraded to best effort. A simulation experiment is performed to evaluate the merit of this suggestion. We chose $\lambda$ in the range of 400 to 1000. The deadline parameter $k$ is selected to be 1.5. The percentage of on-time ADU’s is shown in Table I. It is observed that the performance improvement resulted from the late ADU downgrade scheme is not significant (less than 1%).

F. Effect of Different Deadline Parameters

Our last experiment is concerned with scenarios where different documents may request different deadline parameters. The following two cases are considered:

- Case 1 - the deadline parameter $k$ is assumed to be 1.5 for all real time ADU’s, and
- Case 2 - we assume that 50% of the real time ADU’s use $k = 1.25$ (denoted by RT1) and the other 50% use $k = 1.75$ (denoted by RT2). The average value of $k$ remains at 1.5.

We selected the ADU arrival rates to be 400, 800 and 1200, and presented the results on the percentage of on-time ADU’s.
for T/H, T/H-L, and FCFS in Tables II, III and IV respectively. It is observed that the percentage of on-time ADU’s for RT1 is lower than that for Case 1, while that for RT2 is higher. The average performance, over RT1 and RT2, is not as good as that for Case 1, although the performance difference is not significant. This can be explained by the increased difficulties in meeting deadlines when the value of $k$ is small (e.g., for RT1), thereby lowering the overall performance.

VI. Conclusion

In this paper, we proposed a novel approach to managing network resources in order to support soft real time applications in packet-switched networks. The key element of this approach is the use of a tuple (size, deadline) to characterize the performance requirements of a document (or ADU). A mapping from this tuple to deadline information for each packet was also developed. This information is carried in the packet, and is used by routers for channel scheduling purposes. Two variants of a new deadline based scheduling algorithm, referred to as T/H and T/H-L respectively, were developed. The performance of FCFS, T/H and T/H-L were evaluated by simulation. It was found that T/H and T/H-L are superior to FCFS in terms of the percentage of ADU’s that are delivered on time. Between these two schemes, T/H has slightly better performance, but its implementation complexity is higher, especially when the backlog size is large. We thus conclude that T/H-L is the preferred scheme.

T/H and T/H-L achieve good performance until the arrival rate exceeds some maximum value. This suggests that some kind of admission control is needed in order to maintain a high percentage of on-time ADU’s. The performance evaluation of admission control will be presented in an upcoming paper.

Finally, our scheme can be extended without much difficulty to support hard real time applications on the same network. Specifically, a “hard real time” class may be added, and the head-of-the-line scheduler always gives this class the highest priority.

ACKNOWLEDGMENT

This work was supported by the Natural Sciences and Engineering Council of Canada.

REFERENCES