Experiments with SuperJANET/SMDS

Steven Simpson, David Hutchison

Computing Department
School of Engineering, Computing and
Mathematical Sciences
Lancaster University
Lancaster LA1 4YR
United Kingdom
Fax: (+44)-1524-593608
Tel.: (+44)-1524-65201 Ext. 3820
ss,dh@comp.lancs.ac.uk

ABSTRACT

A series of measurements of a computer link using the Switched Multimegabit Data Service (SMDS), as provided in the United Kingdom SuperJANET network, are described in this paper. The computer link is subject to the effects of using several access networks at the peripheries of SuperJANET, making the measurements valid only in the context of the link from one computer to another, rather than describing any attributes of SuperJANET in general. However, the measurement mechanism is refined several times in an attempt to filter out non-SuperJANET phenomema.

1 INTRODUCTION

The prospect of high-speed, wide-area computer networks becoming available in the near future is leading researchers to consider and to develop new computer applications that will make use of the larger bandwidths available in the networks, providing the user with digital video and audio, as well as text and file-transfer facilities (so-called *multimedia*).

1.1 Quality of Service

It has been established that continuous media, such as video and audio, require the networks that they use to have characteristics that older networks do not provide, and that older protocols cannot guarantee. Both video and audio require high bandwidth and low jitter. Audio needs high reliability, or else it becomes unusable, whereas video can lose the occasional frame, and retransmission might only serve to congest the network further. 'Quality of Service' (QoS) refers to both the attributes of a network, and the guarantee that service characteristics can be maintained during a connection. QoS can also be described in terms of higher and lower levels: lower levels are bit rates, error rates, jitter etc; higher levels are frames or samples per second, screen size, and so on.

1.2 Motivation

With the appearance of SuperJANET in the U.K., providing access rates of at least $10Mbs^{-1}$, a research project at Lancaster University decided to investigate the possibility of using it to connect a client to a multimedia file server [Lougher, 94]. The measurements presented within this paper were initially made to meet this demand, by determining the tuneable parameters of a SuperJANET connection, and obtaining values for these parameters that were ideal for the transfer of video.

Also considered is the use of local access networks and machines, and how their effects on measurements and results can be isolated.

2 SuperJANET AND SMDS

SuperJANET is a wide area network connecting research establishments around the U.K., and providing them with at least one of SMDS, PDH or SDH. The measurements described in this paper are of a data link through SuperJANET/SMDS, from a machine at Edinburgh University to various machines at Lancaster. The network configuration is shown in fig. 1.

2.1 Installation at Edinburgh

snake is a HP700 series machine, and connects through a backbone ethernet, the FDDI MAN at Edinburgh, and then onto SuperJANET.

2.2 Installation at Lancaster

Local networks at Lancaster University are connected by an ethernet hub (LAX20), which is connected to a router (Cisco AGS+). The router is Lancaster's link to SMDS, and restricts the mean access rate to $10Mbs^{-1}$. Machines used at Lancaster have various connections to the hub, and varying qualities of performance.

dcl-sparx1 is a Sun Sparcstation, and connects to the hub via several ethernet bridges.

fester is a Sun Sparc 5, and additionally has a $155Mbs^{-1}$ ATM link to the hub.

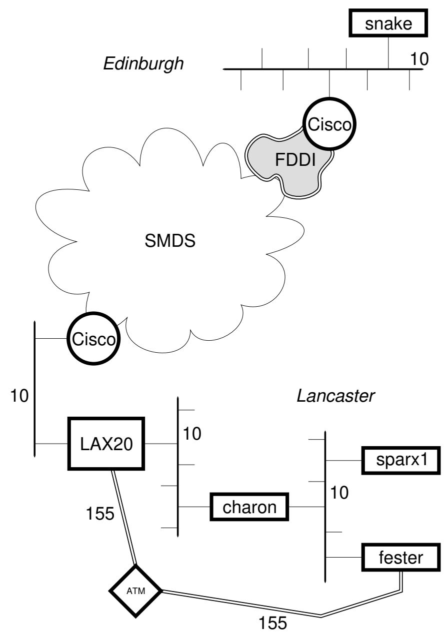

Figure 1: Network configuration

3 END-TO-END MEASUREMENTS

In our experiments, data is transmitted almost entirely from Edinburgh to Lancaster. For each data set, a pre-determined number of fixed-size datagrams are time-stamped, transmitted, and time-stamped on receipt. The *reception* data consists of a sequence of reception times (in milliseconds), some of which will be blank, due to packet loss. The *transmission* data is a sequence of times, but (ideally) with no gaps. Early attempts to obtain this data from the server with no gaps resulted in administrative complications, and so transmission data with gaps had to be accepted.

3.1 Mechanisms

The mechanisms used throughout the experiments have common features:

- UDP/IP sockets are used,
- they model constant bit rate (CBR) video,
- a server runs continuously at one machine, waiting for the client to contact it. This first contact specifies the attributes that the connection should have. The server then generates packets according to the specification,
- after starting up, the client establishes contact with the server, and waits for the arrival of packets, storing details about them in a table, then writing to a file with the raw data. In fact, two files are generated - one for transmission times, the other for reception times,
- a set of utilities is used to format the raw data, such as to calculate bit rates and error rates, and to generate graphs,
- packets contain a sequence number and a transmission time, placed just before submission to lower protocols.

A *measurement* is one connection between the server and client, and all the results generated from it.

Two algorithms are used for transmission of packets (TA1, TA2), and two are used for reception (RA1, RA2).

3.1.1 Transmission

Algorithm 1 (TA1) requires the server to know the number of packets *n*, the inter-packet delay or send-delay *d*, and the packet size in bytes *s*. *d* is measured in cycles round an empty loop. The server transmits *n* packets of size *s*, executing an empty loop *d* times between each packet.

Algorithm 2 (TA2) requires packet rate *p* Hz, bit rate *b* Mbs^{-1}, and *n* as before. After preparing a packet, the server calculates how long it must wait before transmission, and delays that long using `usleep` or `select`. *b*/*p* gives the mean packet size, while 1/*p* gives the mean inter-packet period. A `step` object, which always gives an integral value when evaluated, but on average is rational, is used to represent these values.

Another method for timing of packets involves `setitimer`, and is described in [Ismail, 95].

3.1.2 Reception

Both of these algorithms prepare a table for *n* packets, each entry holding a transmission and reception time, and a flag to indicate that the packet was received.

Algorithm 1 (RA1) has a current packet number *c*, initially zero, and an incoming packet with sequence number *i* is compared with it. If $i = c$, entry *i* of the table is filled in with the packet's details (the flag is set to indicate that the packet was received), and *c* is incremented by 1. If $i > c$, entry *i* is filled in as before, but entries $c...(i - 1)$ are marked as unreceived. If $i < c$, then a packet greater than *i* must have been received (*i.e.* this one is out of sequence), so the packet's details are stored, and the flag indicates that the packet was late. If a (short) timeout occurs while waiting for a packet, the flag for the awaited packet *c* is set to indicate loss, and *c* is incremented. Once $c = n$, the connection is assumed to have terminated.

Algorithm 2 (RA2) sets all flags to 'lost' initially. It then uses a *long* timeout to wait for incoming packets. As each one is received, its details are placed in the table. Connection is assumed to have terminated when the timeout occurs, or all packets are received.

3.2 Sites

Several machines are available for use for experimentation. They are described as *sites* as the place in the networks is important, in addition to their capabilities.

One site is available at Edinburgh, *snake*, and is used throughout as a server. *sparx1* and *fester* are used as clients at Lancaster.

3.3 Experiments

Three configurations were used, specified by [*transmission algorithm*, *reception algorithm*, *server–client*]. For each configuration, there are several types of information to be discovered:

- *Traffic patterns* – the behaviour of traffic throughout the day.
- *Tuneable parameters* – the attributes of a connection which can be altered to improve performance.
- *End-to-end delay* – this requires that the clocks of the client and server be synchronised. As this facility has not been provided, no experiments regarding this characteristic have been done.
- *Round-trip delay* – this is observed as the response time (the arrival of the first packet) at the client, after transmitting the specification. As an approximate rule, the end-to-end delay could be calculated as half of this delay.

Measureable values are calculated as follows:

- *Bit rate/throughput* – for reception, the total amount of data sent, divided by the time difference between the first and last packets received.
- *Error rate* – the number of packets regarded as lost (and with RA1, late; packets cannot arrive late with RA2) as a percentage of the total number of packets. For RA1, this can give separate loss and late rates, but has not been used.
- *Inter-packet (jitter) delay* – the time between two adjacently sequenced packets. The mean, minimum, maximum and standard deviation of each measurement can be calculated. Also, histograms can be generated for a range of jitter delays, to broaden the description of a measurement. Note that when a packet is lost, two values of inter-packet delay are lost, one with each of the packet's neighbours.

3.3.1 [TA1, RA1, *snake–sparx1*]

Traffic Patterns

For one day in 1995, 24 measurements were taken, one every hour, starting at 11:00 and spilling over to 10:00 the next day. *n* was 10000, and *s* was 1KB. *d* was 19000.

The error rate was found to rise exceptionally (38% of packets lost) around 12:00, and moderately (about 10%) between the hours of 13:00 to 17:00. Other times had error rates less than 1%.

The 10:00 measurement was lost for some unidentified reason, possibly being the loss of the first packet from client to server which would initiate the connection. The bit rate achieved with a low error rate was about 4.31Mbs^{-1}. The corresponding inter-packet delays at reception are also presented, in terms of their minima, maxima, means, and standard deviations. The standard deviations and maxima broadly followed the pattern of error rate, peaking at about 27ms and 1400ms respectively, with about 1.5ms and 90ms during low error-rate periods. Minima were within the range 0.415-0.454ms, and means hovered around 1.9ms (range: 1.893ms to 2.956 at 12:00).

Tuneable Parameters

Using the 'quiet' periods identified by the previous experiment, this experiment was carried out while the chances of being disturbed by other traffic were low.

A measurement was taken for all *s* in {1, 2, ..., 7, 8} KB, for all *d* in {1, 3, ..., 15, 17} 10000 cycles, while *n* was varied with *s* to keep the total amount of data transmitted constant (*sn* = 84000KB). The results show that for a particular value of *s*, there is a value of *d* that gives optimum performance: if *d* is lowered from this value, the error rate increases; increasing *d* decreases the bit rate (which it must, clearly) without significant gains in reliability. For example, when s = 8KB, error rate rises from almost 0% to over 95% as *d* is lowered from 90000 to 50000 cycles, remains high when $d < 50000$, and remains low when $d > 90000$.

From the optimum delay of about 90000 cycles, an optimum throughput can be derived, in this case giving a bit rate of about 7·5Mbs^{-1}, and for the range of packet sizes, giving the graph shown in fig. 2:

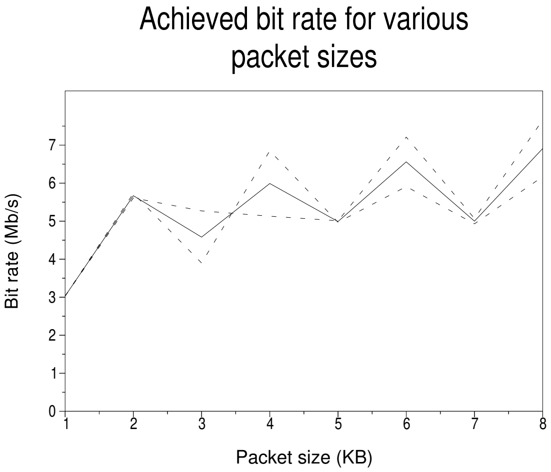

Figure 2

The solid line is the mean of the two broken lines, which were taken at 21:00 on 22^{nd} February 1995 and at 03:00 the following day. It has been suggested that the pattern, which appears to favour even packet sizes (when expressed in KB), is due to the leaky bucket algorithm in the Cisco credit manager (see fig. 1).

The graph shows that 8KB packets are preferred by the connection as a whole. However, it is not clear if this is an aspect of SuperJANET, of the local access networks, or of the machines in use.

3.3.2 [TA2, RA2, *snake-sparx1*]

Traffic Patterns

To improve on the earlier experiment, this experiment covers a week, at half-hourly intervals, and averages the readings for each time of the day to give a profile of the day.

The measurements commenced at 00:00 on Sunday, 26^{th} March 1995, and use a 1·5Mbs^{-1}, 24 packets s^{-1} connection. Most error rates were below 3.5%, though there were three peaks at 03:30 (5.9%), 04:30 (13.8%) and 07:00 (6.5%). The results also demonstrated that, using TA2, the server was not a huge source of jitter, though the range was about 38-50ms most of the time, with a widening at 10:00 to 30-60ms. At the client, this range has spread out further, with the maxima reaching over 1200ms, and some negative minima (*i.e.* some packets have been re-ordered in transit).

Tuneable Parameters

Here, a 1·5Mbs^{-1} stream is modelled, and the only parameter tuned is *p*, from 4 to 24 packets per second. (The range is limited by the maximum IP packet size, and the reliability of the timing algorithm at the server.) We attempt to find the best packet size for one particular bit rate that is expected to be used most often, while the earlier experiment tried to find the best bit rate achievable for a range of packet sizes.

Measurements were taken for seven days, starting on 30^{th} May 1995, at quiet periods identified by the traffic patterns of this configuration, namely 06:00, 08:00 and 22:30 (whose error rates were less than %1), and at the apparently busy period of 04:30.

The most significant results from this configuration are the standard deviation values for inter-packet submission and reception times. The 06:00 measurements are used as an example.

The values for standard deviations of submission times demonstrate a characteristic of the transmitting machine: the scheduling mechanism favours inter-packet delays of 50ms and 100ms, which correspond to packet rates of 20Hz and 10Hz respectively, and give deviations less than 0.5ms, while all other values are over 3ms. Clearly, *for this particular type of machine*, these two packet rates should be favoured. The corresponding graphs for the other three times show almost identical patterns.

By subtracting the submission values from the reception values, we can derive values for the quality of the connection excluding the server machine. The resultant data is very similar to the reception data, due to the differences in scale between the original graphs (6-32ms for reception, 0-5ms for submission), though it does have some troughs in different positions.

Comparing the 'difference' data with its counterparts for the other times, we see that there is always quite a low trough around 22Hz. In fact, the data for 04:30, 06:00 and 22:30 all show a trough at this point, while 08:00 shows a trough a 23Hz.

The only other obvious pattern is that for low packet rates, the standard deviation is consistently high, at about 25ms.

3.3.3 [TA2, RA2, *snake-fester*]

The experiments of the previous configuration are intended to be repeated using the new configuration. This configuration has not been set up yet, so no experiments have been carried out using it.

4 ASSESSMENT

These are mainly critical observations of the algorithms and network configurations used.

4.1 TA1 and TA2

There are several faults with the earlier algorithm. Firstly, it uses an inter-packet gap that defines the time from the *end* of the submission of one packet, and the *beginning* of the next, and specified in imprecise units (loop cycles). TA2 improves over this by using `gettimeofday` and specifying the time in microseconds between the start of two adjacent packets.

This time is determined from the user-supplied bit rate, and as it could be rational when expressed in microseconds, a `step` object holds this value, which evaluates to an integer whose mean value over several evaluations is the same as the rational value. The `step` object is incorporated into a `ticker` object, which stores a time, waits for the time to be reached, then increments by the value of its `step`. This is a more rigorous timing mechanism, as demontrated by the submission statistics, but could be improved by being used in a lower protocol layer.

4.2 RA1 and RA2

RA1 was designed without fully understanding how to interpret packet arrival, and this is reflected in the way that it regards many packets as late when a higher-sequenced packet arrives – precluding an opportunity by the application to choose to store the early packet while waiting for the later ones.

An earlier version of RA1 treated all late packets as lost (discarding their information if they arrive late), and the current RA1 still does this if the out-of-sequence packet that marks the other packets as late happens to be the last one (the algorithm assumes the end of the connection has been reached).

RA1 also requires its timeout value to be carefully set: too long, and a failed (unsupervised) connection will take a long ($n \times$ timeout) time to finish, and allow the next automatic measurement to begin; too short, and packets are missed too frequently.

RA2 has a much simpler design, removing the notion of late packets, and deciding that the end of the connection has been reached when no packets arrive within a long period of time. It also has no concept of a current packet, which RA1 uses to determine whether a packet is late. In an extreme case using RA1, the last packet may arrive first, causing the algorithm to discard all other packets. It should at least have marked them as late, but as it would terminate under that condition, they can only be regarded as lost.

If RA2 were to be evolved into a new protocol that regularly delivered packets (or perhaps frames, as it is for video) in sequence to an application, it may be helpful if the application could specify that some packets would be too late for processing if they arrived now, so that the protocol could avoid attempting to reserve space for such packets and re-ordering them. It may also be able to signal immediately to the application not to expect certain packets.

Enhancements of RA2 would include enabling it to identify frames from the packets. If it also knows when these frames are to be displayed, and therefore when the component packets for late frames are not needed, it can drop the packets, and make space for the next complete frame, rather than allowing a buffer to overflow with useless frames. It could also signal to the transmitter to reduce its frame rate. Such “frame-knowledge” enhancements are complicated with MPEG by the difference in transmission order and playback order [MPEG 1, 93].

4.3 Network Configurations

It is unclear how much effect purely local traffic has on the measurements, though by ‘subtracting’ purely local measurements from the combined measurements, it may be possible to attribute some characteristics to SMDS. For example, if a purely local measurement demonstrated that the end-to-end delay of travelling packets was constant, then any variance in a combined SMDS-local measurement could be said to be due entirely to SMDS.

Ideally, we should avoid local traffic altogether, though not necessarily *other* SMDS traffic, as this is what is intended to be measured by ‘traffic patterns’ experiments. Note that ‘local traffic’ also includes processes, that have nothing to do with the measurements, sharing the machines being used.

4.4 Composition of Results

We have rather arbitrarily combined two data sets to form fig. 3.1, which were taken at different times of the day (it was assumed that both times being ‘quiet’ would be enough to regard them as identical). Several more measurements should have been taken, and at the same time of day, to give better statistical performance. That opportunity was overlooked in favour of improving the mechanisms used, after which, more extensive tests would be conducted.

5 SUMMARY OF RESULTS

Assuming that the effect of local networks and machines is not significant, the data indicate that connections of packet rates (or smaller packet sizes) around 22Hz at 1·5Mbs^{-1} have better jitter characteristics, and reasonable error rates. The lower packet rates (7, 8Hz) consistently do badly, while intermediate rates seem unpredictable. If a video stream has a variable bit rate, it is most likely that it will have to vary its packet rate to give an ideal packet size which the connection favours (1·5/22 Mb is about 9KB, just below the Cisco credit manager limit, 9188 bytes).

Initial experiments showed alarmingly high error rates in places, but by more carefully controlling the injected bit rate, *i.e.* lowering it rather than aiming for an indeterminate, high value, error rates are considerably reduced. (SMDS will accept 1·5Mbs^{-1} quite well.)

6 FURTHER WORK

These ideas are intended to meet the shortfalls of the work done so far.

Isolated SMDS

The fact that there is other traffic on SMDS, and that, at Edinburgh, the connection also covers an FDDI network, affects all the results so far gathered. Although traffic analysis experiments are supposed to measure other traffic, the remoteness of *snake* from SMDS taints even this.

So for non-traffic analysis experiments, an isolated SMDS network, where the amount of traffic on it can be controlled, would be ideal for such experiments. Local traffic could be completely removed.

Dedicating the machines in use is also required, unless the kernel can be improved to give the performance of a dedicated machine. Either way, the effect of irrelevant processes is reduced to insignificance.

Comparison of Machine Types

Switching from one machine to another (*e.g. sparx1* to *fester*) to make use of better network facilities affects the experiments if the machines are not of the same performance. It makes the source of any change in performance of the connection unclear.

It would be helpful in these cases if one could look up an entry for the two machine types in a table, and read off the relative performances. This process would be further complicated by various hardware extensions to the machines that give them networking capabilites.

Greater Control of Lower Protocols

To reduce the effects of scheduling when attempting to send packets at a regular rate, more control of the lower protocols, or if necessary, a new protocol, would help to bring the meanings of 'transmission' and 'submission' of packets closer.

Credit Manager Model

A model of the credit manager in the Cisco router may shed light on the shape of the graph in fig. 2.

MPEG Model and MPEG Data

As the multimedia file server deals with MPEG streams, a refinement of the model may increase the significance of the measurements. Instead of producing a steady 1·5Mbs^{-1} stream (*i.e.* of fixed packet size), the packet size and rate could be varied according to the properties of real MPEG streams, pehaps using probability tables to determine the next packet size from the previous ones. Better than that is of course to use real MPEG data.

Statistical Arithmetic

This involves working out how to ‘subtract’ one set of measurements from another. For example, if two separate networks have certain jitter properties, can these values be combined to describe a connection across both networks layed end-to-end?

Direct Measurements

It may also be possible to isolate SMDS from local access networks by monitoring packets at the connections to SMDS: for example, at Lancaster, this is at the Cisco router.

Data gathered in this way would demonstrate a link between performance and usage of the network if any existed, without the need for knowledge of the performance of local networks and machines.

ACKNOWLEDGEMENTS

This research is carried out in the context of an EPSRC-CASE studentship award (ref. 9456391X), with sponsoring from BT Labs, and in the context of the BT-URI (Management of Multiservice Networks) project.

REFERENCES

[Ismail, 95] Ismail, N., Hailes, S. and Wilbur, S. “Multimedia traffic tests over SMDS network”, Department of Computer Science, University College London, Gower Street, London WC1E 6BT, UK, February 1995.

[Lougher, 94] Lougher, P., Shepherd, D. and Peglar, D., “The Impact of Digital Audio and Video on High-Speed Storage”, Proceedings of the Thirteenth IEEE Symposium on Mass Storage Systems, Annecy, France, June 1994.

[MPEG 1, 93] ISO IEC JTC 1; “Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media up to about 1·5Mbit/s”, International Standard ISO/IEC IS 11172, 1993.

Additional diagrams from this work are available separately from:

[Simpson, 95] Simpson, S., Hutchison, D., “Experiments with SuperJANET/SMDS” (internal report complete with diagrams), Computing Department, Lancaster University, Lancaster LA1 4YR, UK, 1995.