ATLAS Internal Note
DAQ-No-102
June 4, 1998

Emulation of the Level-2 trigger, architecture B, on the Macramé Testbed

R.W. Dobinson[a] S. Haas[a,b] R. Heeley[a] N.A.H. Madsen* [a,c] B. Martin[a]
J.A. Strong[c] D.A. Thornley[a,d] M. Zhu[b]

[a] *CERN, Geneva, Switzerland*
[b] *University of Liverpool, Liverpool, UK*
[c] *RHBNC, University of London, London, UK*
[d] *University of Kent, Canterbury, UK*

June 4, 1998

Abstract

This note presents results obtained from emulating the second level trigger, architecture B, on the Macramé Testbed. The five networks of architecture B were emulated separately on a 512 node Clos DS link switching network. An event rate of 125 kHz was achieved for all networks, except for the calorimeter network for which the maximum event rate was nearly 80 kHz. The impact on the throughput of how data sources and sinks are distributed throughout the network is explored and a simple model for the latency across the network is presented.

1 Introduction

The motivation behind the emulation programme was to learn about the performance of networks under Atlas second level trigger, architecture B[1] traffic conditions. Measurements have been made using a large IEEE 1355 DS-link[2] network/switching fabric, the Macramé Testbed[3][4].

The Macramé Testbed consists of traffic generating nodes, timing nodes and a DS-Link network. The configuration of the testbed, used for the measurements, consisted of 512 traffic generating nodes, two timing nodes and the network wired up in a Clos topology. A 512 node 3-stage Clos network is shown in Figure 1. Each terminal stage switch connects with a single link to every centre stage switch. The choice of this topology came naturally because it has previously been shown both theoretically[5] and experimentally[3] to be the most performent. Between fifty and one hundred per cent of each second level trigger subsystem was emulated.

A diagram showing the main components of architecture B is shown in Figure 2.

*email to Nina.Madsen@cern.ch

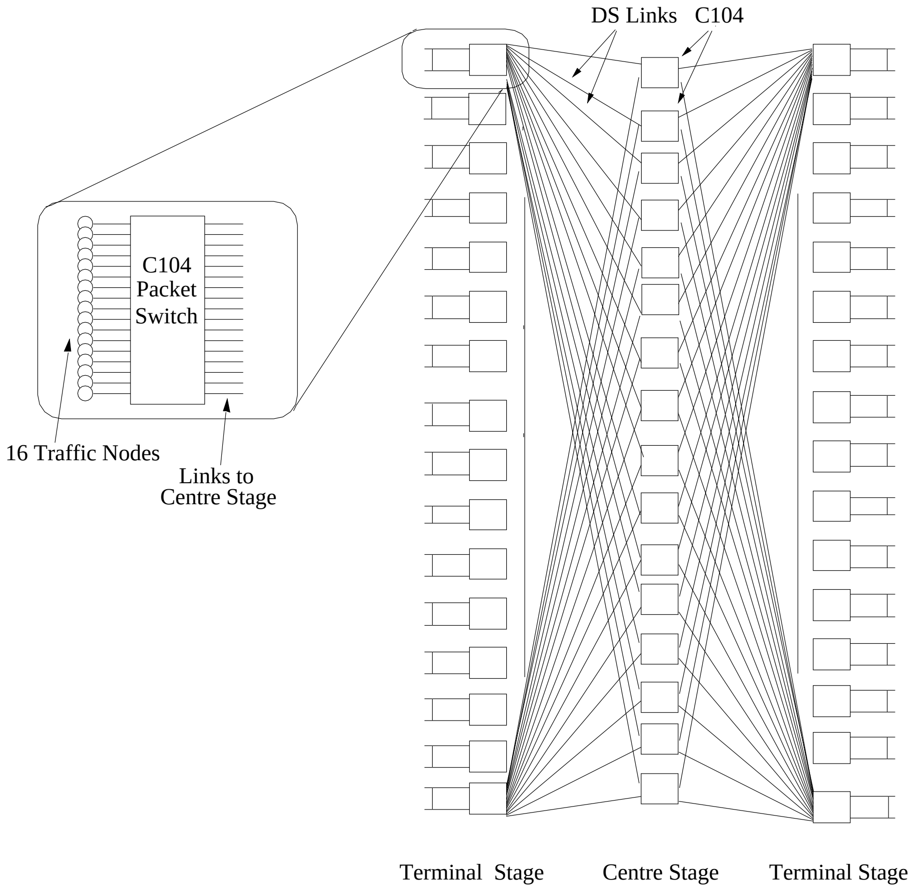

Figure 1: Diagram of a 512 node Clos network.

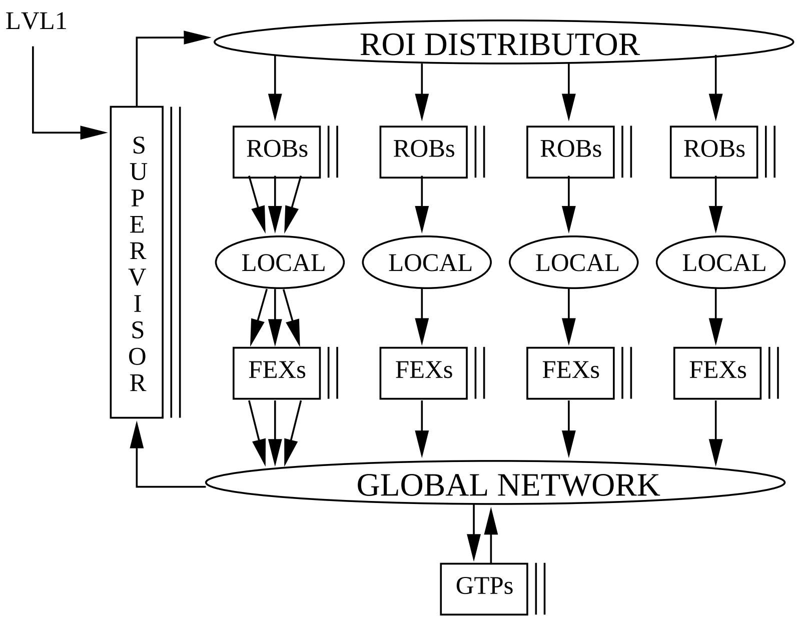

Figure 2: Diagram of architecture B showing the four local networks and the global network.

The aim of this note is to discuss the constraints and possibilities of the Macramé Testbed, the measurements performed on it, the results obtained and the understanding which emerged from the study of architecture B.

2 Constraints and Possibilities

Macramé is a network testbed, not a parallel computer. This gives some practical limitations to what is possible and what is not. The traffic generating nodes send data out into the network according to traffic descriptors. Each node looks in its memory, to determine when, according to the global clock, it should send a packet with a specified size and destination. The total amount of data sent and received by each node is logged such that, at the end of the emulation run, knowledge of the transmit and receive rates can be obtained, i.e. simple averages. The traffic generating nodes have no ability to act upon either the content, or the number of packets they receive. A packet once received is discarded. A timing node sends trace packets with a specified frequency and length, to a second timing node. The receiving timing node calculates the latency of the trace packet by comparing the time stamp in the packet to the global clock. The packet descriptors of the timing nodes can only be specified once, i.e. with a specific interval, a certain length packet is always sent to the same destination. For more detail on the Macramé Testbed see [4].

Macramé allows each node to independently deliver a predetermined set of packets to the network and can as such be used to model Atlas level-2 architecture B traffic where data is pushed through the network. The size of the testbed is comparable to the size of the networks needed in the trigger.

3 Input Assumptions

This section describes:

- the trigger menus,
- the configuration of the subdetectors,
- the trigger architecture under investigation,
- the simulated event scenarios which were used for driving the emulations.

The input parameters for the emulation originates from many subgroups of the Atlas experiment. In preparation for the different modelling activities during the 'demonstrator project', two internal Atlas DAQ notes [6, 7] were prepared. Their basic aim, was to collect all the important numbers and so establish a set of well-defined assumptions from which to work. In the following section we summarise the parameters used in the emulation.

3.1 Trigger Menu and Selection Strategy

To emulate the trigger, the number of region of interests(RoIs) per event and their types are needed in order to produce the correct traffic through the network. For the emulation studies, the high-luminosity, extended trigger menu has been used. A full listing of this trigger menu is contained in DAQ note 70 [7].

Another important assumption is the selection strategy used by the trigger. For architecture B, event selection is made after all features of all the regions of interest have been determined (in parallel).

3.2 Detector and Trigger Parameters

The calorimeter consists of two subdetectors: the electro-magnetic calorimeter(ECAL) and the hadron calorimeter(HCAL). For the Inner Detector, data from the Transition Radiation Tracker(TRT) and the Semi-Conductor Tracking(SCT) detectors are available to the level-2 trigger. From the Muon Spectrometer, data from the Monitored Drift Tube(MDT) chambers and from the Resistive Plate Chambers(RPC) are used.

Detector System	Sub-detector	# ROBs
Inner Detector	SCT	256
	TRT	512
Calorimeter	ECAL	432
	HCAL	48
Muon Spectrometer	MDT	192
	RPC	22
Total:		1462

Table 1: The number of Read-Out Buffers(ROBs) for the different subdetectors.

Each subdetector feeds the Read-Out Buffers(ROB) with data via the front-end electronics. The number of buffers for the individual subdetectors is detailed in Table 1. All the data in the ROBs are available to the level-2 trigger. The amount of data in the event fragment stored in a buffer depends not only on the detector but also on the type of the trigger; see Table 2. For each local processor participating in an event, 150 bytes are sent to the global processor. This amount of data corresponds to the feature which the local processor has identified. The local processor is also referred to as the feature extractor(FEX).

Detector System	Sub-detector	Jet RoI	em RoI	μ RoI
Inner Detector	SCT	—	1000	1000
	TRT	—	740	740
Calorimeter	ECAL	60	1300	1300
	HCAL	890	890	890
Muon Spectrometer	MDT	—	—	600
	RPC	—	—	100

Table 2: Size in bytes of an event fragment in a ROB for a given RoI type used in the emulation.

Architecture B has five processor farms; four local and one global. The precise size of the processor farms is an unknown at present, since it depends on the processing requirements as well as the technology available at the time of construction. Assuming 500 MIPS per processor and the current trigger algorithms, the 'Paper Model'[8] has estimated the size of the farms; these numbers are quoted in Table 3. Initial emulation measurements had shown that a maximum output on the traffic nodes with this kind of traffic was limited to 8 MBytes/s. To ensure that the data fragments could be accepted by the receiving traffic nodes at a 100 kHz event rate, we adjusted the farm size such that, the output rate would be around 8 MBytes/s at an event rate of 120-130 kHz. In Table 3, the farm sizes used for the emulation have been shown; the number of MIPS needed per processor for current algorithms is also shown.

Farm	Paper Model		Emulation	
	Farm size	MIPS	Farm size	MIPS
Global	18	500	12	750
SCT	37	500	80	231
TRT	41	500	118	174
Calorimeter	74	500	286	129
Muon Spectrometer	5	500	10	250

Table 3: Farm sizes for High-Luminosity/extended Trigger Menu. The MIPS quoted under the emulation is calculated using the adjusted farm size and the current processing estimates.

3.3 Drive Files

The drive files contain the event flow through the level-2 trigger in terms of how much data enters the system and how long the data is being processed at the various stages. The drive files are generated during computer modelling[9] and provide input to the emulation. This ensures that the emulation, the computer modelling and the 'Paper Modelling' start on common ground. The parameters listed previously in this note are the parameters which have been fed into the computer model. The event fragment sizes in Table 2 are only detector data. For the emulation carried out on Macramé, 32 bytes have been added to take into account additional information needed by the level-2 trigger protocol, as is done in the computer model.

In the computer modelling, the trigger item for the current event is picked from the trigger menu[7], where each menu item has a probability of being picked in proportion to its frequency. The menu item specifies the number of RoIs and their individual types. From the spatial position of the RoI and its trigger type the number of ROBs can be calculated from the mapping of ROBs onto the subdetectors[10].

The task of the drive file is to provide the emulation with the distribution of data over the read-out buffers. The drive file includes information that in the real system will come from level-1, i.e. number and position of the RoIs in the detector. As no physics algorithms are executed Monte-Carlo data volumes are used, but with dummy data content.

The drive file generated for the emulation contained 140,000 events corresponding to 1.4 seconds at 100 kHz event rate for the second level trigger. Due to memory size constraints in the traffic generating nodes, each emulation run used a large subset (38-108 thousands events) of the full event sample.

The event fragment distributions, which are derived from the drive files, are shown in Figure 3.

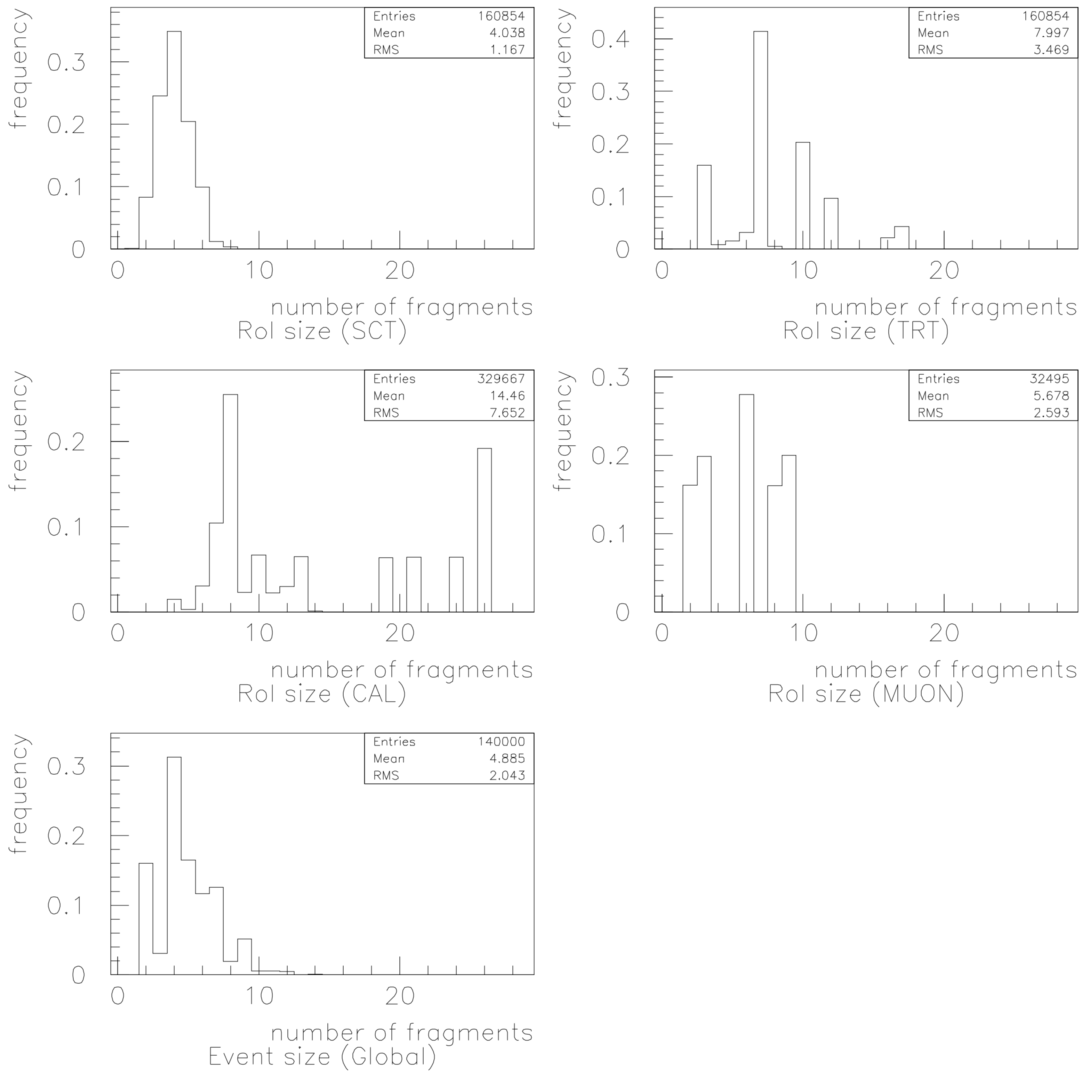

Figure 3: The event fragment distribution from the five networks.

4 Implementation Details

Having provided the reader with an overview of the general input parameters used during the emulation as well as the main constraints, the focus is now turned toward the details of the emulation setups and results. First, the assumptions made will be summarised, this is followed by a description of the measurements performed.

4.1 Assumptions

The time distribution between events is assumed to be constant and the timing of the individual packets associated with a given event is assumed to be synchronised, i.e., all ROBs participating in a given event are instructed to send the data at the same time according to the global clock. In the global network the same assumption is applied to the FeXs. This strict synchronisation will not occur in ATLAS but, due to various smoothing mechanisms in the data acquisition system, it is probably closer to reality than a random timing distribution.

As the size of the available network is smaller than the local TRT and Calorimeter networks we chose to emulate the first N per cent of the network. This means that if an RoI is within the first N per cent it is fully emulated, if it straddles the border only some of the data packets are sent, and if the whole RoI is outside the N per cent it is completely discarded.

4.2 Generation of Traffic Descriptors

For each traffic generating node a program extracts from the drive file the following information for each packet to be sent:

- the time between packets
- the length of the packet
- the address of the receiving node

As there is memory for only 6552 packet descriptors per node, the number of events which the traffic nodes cycle over in the emulation varies, depending on how often data from the given source are needed. Table 4 gives the number of events for each subnetwork emulated. Precautions were taken to ensure that the descriptor lists for all traffic generating nodes wrap around at the same time.

Processor, i.e. receiver, assignment is made using round-robin scheduling. For the global network emulation, the FeXs to be used as sources are assigned by a round-robin scheduler per subdetector.

4.3 The Basic Measurements

So far, we have explained the inputs to the testbed. What happens during the emulation will now be discussed. After initialisation, all traffic nodes are started at the same time and continue to send packets according to their packet descriptors. When the traffic nodes have been sending for some time, such that a steady state is reached, the timing nodes start sending their trace packets. This scenario continues until of the order of 100,000 trace packets have been transmitted. The timing nodes are positioned on different terminal-stage switches as this forces all the trace packets to cross the centre-stage of the network, corresponding to a worst case packet latency.

The timing node discriminates between data-packets coming from traffic nodes and trace-packets coming from another timing node. Whereas, a traffic node monitors how many bytes per second it transmits and receives, a timing node records only the latency of incoming trace packets. This latency is referred to as the 'single packet latency'.

As the network approaches saturation the traffic nodes cannot send their packets at the specified time intervals. Two possibilities arise corresponding to a transient and a permanent state. In both cases the traffic nodes knows that it has a back-log of packets to send. In the first case, it manages to make up for the lost time by sending the next few packets back to back. In the second case, the node cannot catch up and the synchronisation between the nodes is irreversibly lost once a buffer counter inside a traffic node overflows. In this case the 'offered' and 'accepted' data rates will differ.

5 Configurations Emulated

A number of emulation runs were performed. During each run only one of the level-2 trigger networks was emulated. Table 4, shows the number of components emulated for the individual

Network	% emulated	Sources	Sinks	# events
SCT	100 %	256 ROBs	80 FeXs	101,442
TRT	80 %	414 ROBs	94 FeXs	105,729
Calo	66 %	286 Ecal ROBs 32 Hcal ROBs	190 FeXs	38,171
Calo	50 %	216 Ecal ROBs 24 Hcal ROBs	143 FeXs	38,171
Muon	100 %	192 MDT ROBs 22 RPC ROBs	10 FeXs	67,863
Global	100 %	80 SCT FeXs 117 TRT FeXs 288 Calo FeXs 10 Muon FeXs	12 GTPs	108,874

Table 4: Configuration of the emulated networks.

networks. Initially, sixty-six per cent of the calorimeter network was emulated. This network was later emulated with a different component to node mapping — allowing 50% of the network to be emulated, using three traffic nodes per HCAL ROB.

Two basic mappings of components to traffic nodes have been used for the emulation; 'grouped' and 'distributed'. In the grouped case, the first sixteen sources are placed on traffic nodes connected to a single terminal-stage switch; the following sixteen are placed on the traffic nodes connected to the next terminal-stage switch and so on, for all sources, thereafter the same mapping is continued with the destination components. For the distributed mapping, the first thirty-two[1] sources are placed, one on each of the terminal-stage switches, then the next thirty-two sources are placed etc.

[1]There are thirty-two terminal-stage switches in the Clos network.

until all the components have been accounted for, see Figure 4.

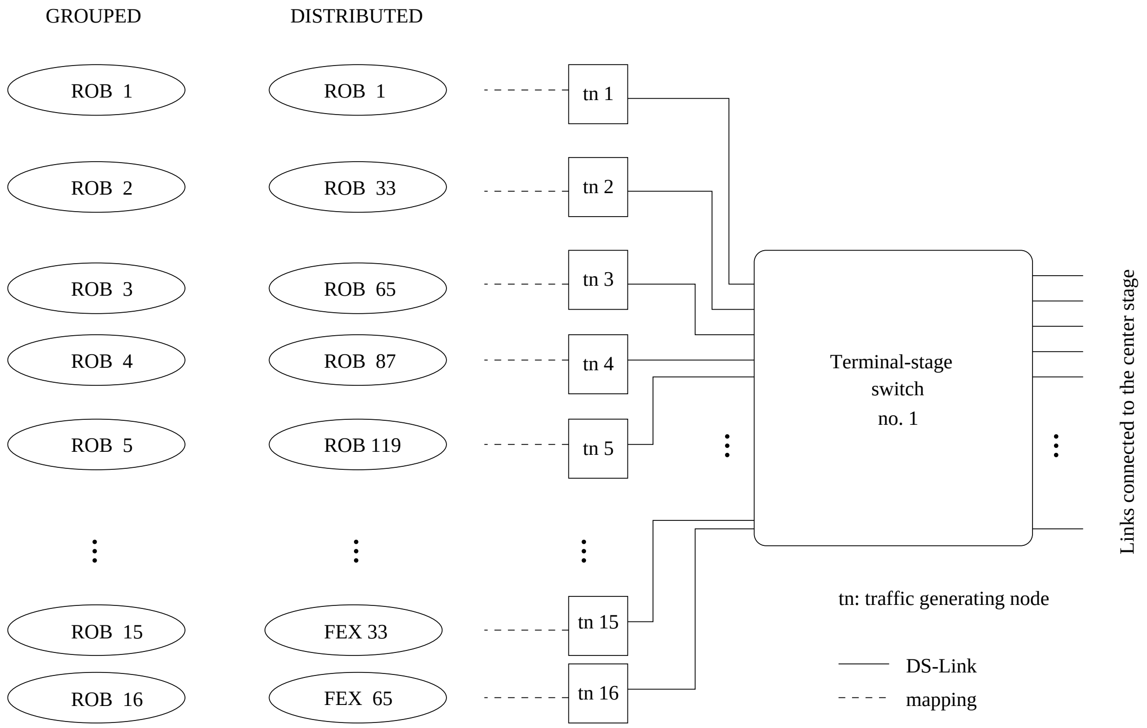

Figure 4: The two different mappings, grouped and distributed for terminal-stage switch number 1 in the case of the SCT subdetector. See Figure 1 for a diagram of the Clos network.

6 Results

In this section the results and conclusions from the emulation are presented.

6.1 Distributed versus Grouped Mapping

When comparing the distributed versus the grouped mapping, it was observed that for both the single packet latency and the achieved event rate, the distributed mapping gave a better performance, independent of the frequency or the emulated network, see Table 5. The throughput scales linearly with the event rate as long as the attempted event rate is achieved. We have used $l_{1\%}^{10\mathrm{kHz}}$ as a measure of the latency where $l_{1\%}^{10\mathrm{kHz}}$ is defined as the time for which only 1 % of the packets exceed this latency at an event rate of 10 kHz.

In Figure 5, the network throughput versus attempted event rate for both the distributed and the grouped mapping have been plotted for the TRT local network. Since the throughput scales linearly with the event rate as long as the attempted event rate is achieved, it is seen that the emulation of that network achieved an event rate of 125 kHz in the distributed case, and only 80 kHz with the grouped mapping. In the same figure, a single packet latency plot at 10 kHz event rate of the TRT network have also been shown. Here the probability that the single packet latency is greater than a certain time is plotted. The evidence for the distributed mapping is clear, e.g., for

Network	Event Rate [kHz]		$l_{1\%}^{10\text{kHz}}$ [μs]	
	Grouped	Distributed	Grouped	Distributed
SCT	80	125	599	408
TRT	80	125	984	600
Calorimeter	25	25	945	830
Muon	100	142	358	264
Global	100	125	103	74

Table 5: Achieved event rates for the various network for both the grouped and the distributed mapping.

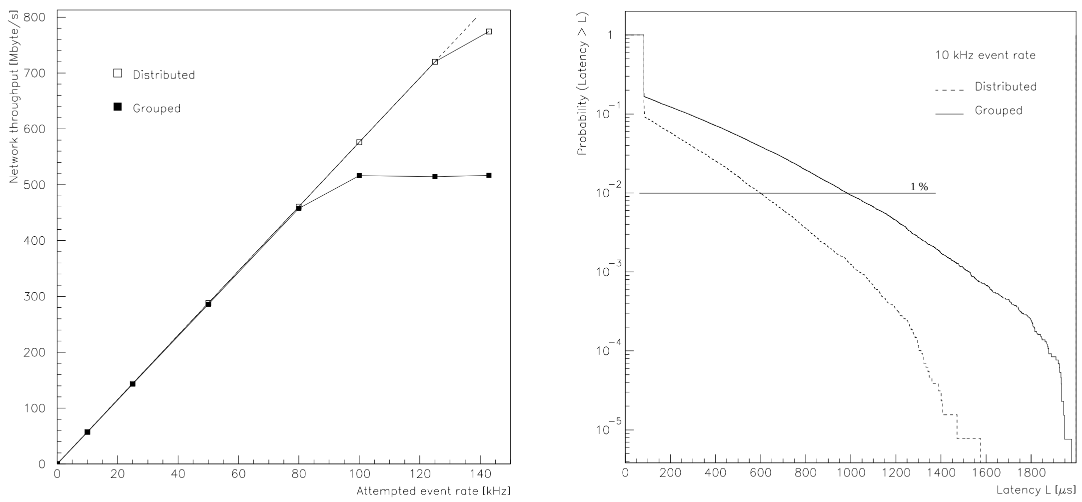

Figure 5: Network throughput and the single packet latency distribution for 80 % of the TRT network. Both the grouped and the distributed mapping of emulated components onto the traffic generating nodes are shown.

the distributed case 99 % of the packets have a latency less than 600 μs; the corresponding number for the grouped case is 984 μs.

A network hot-spot occurs in the grouped case when packets from up to 16 consecutive events are heading for destination nodes connected to the same switch, in the distributed case the packets are funnelled to a new terminal-stage switch for every event. The improvement can thus be attributed to reduction of congestion in the terminal stage switch for the receivers.

It must therefore be concluded that a distributed mapping of the emulated components onto the network nodes is preferable to the grouped mapping in terms of network performance. The results quoted in the rest of this note have all been obtained with the distributed mapping.

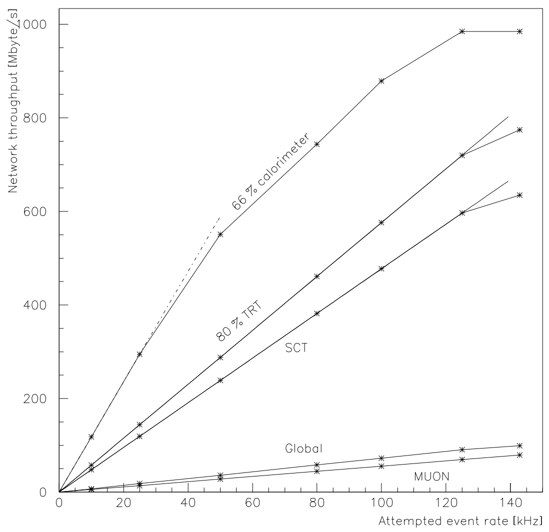

Figure 6: Throughput versus attempted event rate for the distributed runs.

In Figure 6, the achieved network throughput as a function of the second level event rate is shown for all the emulated networks. Any deviation from a straight line when the attempted event rate is increased means that the network is congested and can not sustain the rate. One observes that the 512 Clos network could sustain the traffic emulating the global, the SCT and 80% of the TRT network at up to 125 kHz and the muon network even up to 142 kHz. The network could, however, only deal with an event rate of 25 kHz for 66% of the calorimeter network.

6.2 Optimising the Hadron Calorimeter ROB Mapping

From the 'paper modelling'[8] it was known that the HCAL read-out buffers were heavily loaded at 100 kHz and we expect the output DS link to saturate at around 60 kHz event rate. The deterioration in performance above 25 kHz was not expected and needed explanation.

In order to try to overcome the poor performance we tried two additional configurations. The basic idea was to enhance the HCAL ROBs interface to the network. First, an increased link speed was emulated by using three traffic nodes to send the data for one HCAL ROB, where for an event each traffic node sends one third of the data. Since they all fire at the same time, this scenario is called ‘Salvo’. In the other scenario, three traffic nodes per HCAL ROB were also used, but here the traffic nodes took turns to send an event, i.e. one traffic node sends all the data, but only participates in one third of the data transfers. This scenario we call ‘Round Robin’. All three scenarios were run with 50% of the calorimeter being emulated.

Measurements on the three configurations are shown in Figure 7. The top row shows the transmit rate as a function of the traffic node number, the second row shows the corresponding hit rates; all the plots correspond to a 10 kHz event rate. The spikes are the HCAL ROBs, the ECAL ROBs appear as the steady ‘background’.

The first column *a*) contains data where the calorimeter has been emulated, with a distributed mapping. Here it can seen from the transmit rate that, by scaling up to 100 kHz, the network will not be able to sustain such an event rate, since this would demand an transmit rate of 13-15 MBytes/second out of the HCAL ROBs well beyond the theoretical available link speed for data of 9.5 MBytes/s, (which is why the performance is expected to saturate above 60 kHz).

The first approach was to use the ‘Salvo’ scenario described earlier. Looking at the transmit rate in the second column *b*) in Figure 7, one observes that the link bandwidth is not a limiting factor even at a 100 kHz event rate. Neither at the output from the buffers nor at the input to the feature extractors are the links saturated. The third column *c*) shows the transmit and hit rate for the ‘Round Robin’ scenario.

The cause of the problem requires a more detailed study of the data transfers to the FEXs. For the calorimeter data, there are on average 14 event fragments to be collected, with a minimum of 4 and a maximum of 26 fragments, see Figure 3. On average two of the 14 fragments will come from the HCAL RoBs and each contain 890 bytes of data plus the 32 byte protocol header. The rest of the fragments come from the ECAL ROBs and have sizes according to the RoI type, see Table 2. From the trigger menus, approximately half the RoIs are jets. Collecting the 14 fragments takes of the order of:

$$2 \text{ fragments} * 922 \text{ bytes} + 12 \text{ fragments} * 1/2(1332 + 92) \text{ bytes} = 1\ ms$$

This implies an average wait-to-send time of 0.5 ms. During this waiting time a sending node participating in the event is stalled and any subsequent events are queued. This is known as ‘head-of-line-blocking’. The wait-to-send time should be compared to the hit-rate. At 100 kHz event-rate the hadron calorimeter read-out buffer hit rate is 16 kHz giving an average separation for sending of data fragments of only 63 μs. This is clearly much smaller than the average 500 μs wait-to-send time calculated above. However, queued packets for subsequent events will suffer a shorter wait-to-send time when they do access the network since some other fragments for that event will have been transferred already. Measurements on the test-bed show that problems arise in the case of the calorimeter, when the hit separation is less than half the average wait-to-send time.

The Salvo technique does very little to alleviate this problem as it mainly reduces the time a link is active. The small improvement in performance, see Figure 8, is due to the increase in probability of transferring part of the data with a reduced wait-to-send time. The Round-Robin technique reduces the hit-rate for any one HCAL ROB links by a factor of three and therefore improves the rate limit by a similar factor. With this technique the network operates up to nearly 80 kHz.

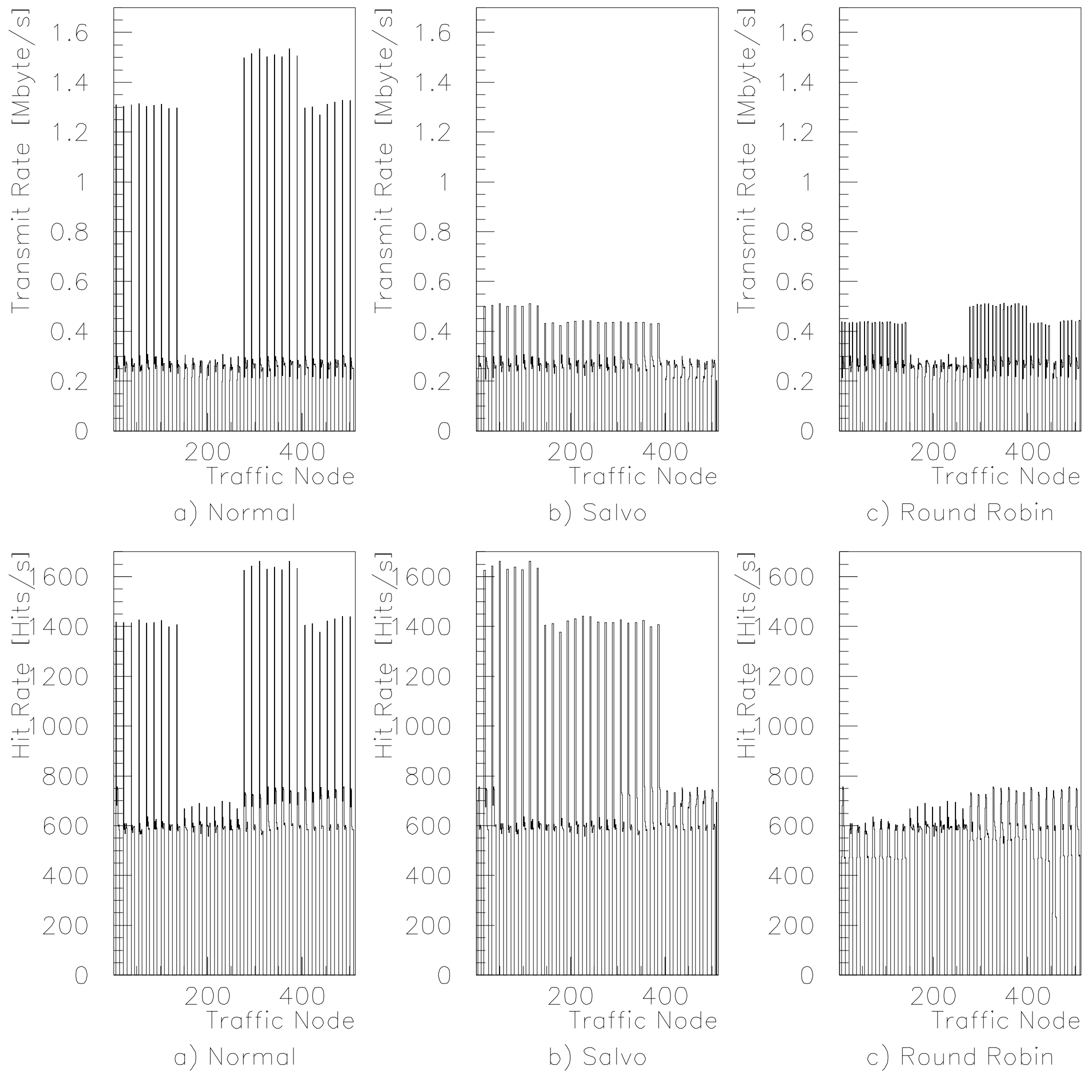

Figure 7: The first row shows the transmit rate for the calorimeter ROBs at 10 kHz event rate, the second row shows the corresponding hit rate. The first column *a*) is from emulating half the calorimeter using the normal distributed mapping. The second *b*) and third *c*) column is for the 'salvo' and 'round-robin' scenario respectively. The spikes in the histogram corresponds to the HCAL ROBs, the 'background' are the ECAL ROBs. The rearrangement of the spikes along the node axis is because the exact mapping was not the same for the emulations, but the overall distributed mapping was still applied.

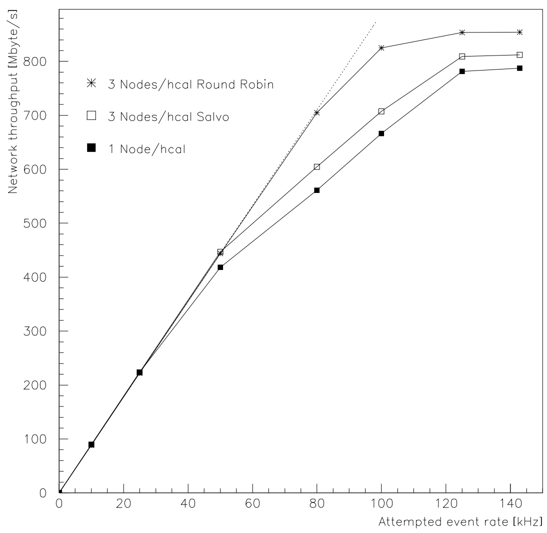

Figure 8: Throughput measurement on the three different emulations of half the calorimeter network.

Alternatives to the round-robin technique are:

- Using faster links throughout the network. This will decrease the wait-to-send time for the buffers since the data are removed faster from the links.
- Structure the data dispatch times so that the wait-to-send time is substantially reduced, i.e. traffic shaping.

6.3 Modelling the Latency Distributions

In an attempt to understand the latency distributions a simple model has been developed. The latency distributions are measured with trace packets sent from one timing node to another timing node. The main idea behind the model is that all the packets compete equally for access to the destination node and have to queue until they are successful. This is modelled by assuming that there is one effective point of contention in the network and is consistent with the assumption that a Clos switching network is non-blocking as long as there are not two or more nodes wishing to talk to the same node.

The model assumes that:

- all packets including the trace packets are the same size and require a time t_p for transfer across a single link,
- all data packets from an event attempt to access the network simultaneously,
- there is a maximum of two events in the network at any point.

Consider a time interval where the first competition time is at time 0, the second at t_p, the third at $2t_p$ etc. At the first competition, there are N data packets, at the second one $N-1$ etc. Where N is the total number of data packets for this event. One overlapping event is taken into account by increasing N with M, the number of packets in the next event, after some random time. The probability for overlapping events has been taken from the following distribution

$$\text{Prob} = \frac{1}{X}\int e^{-t/X}\,dt$$

where X is the mean time between events. The probability of having more than one overlapping event is considered negligible. The trace packet, which arrives at the network at a random time, waits for the next competition when it competes equally with all the remaining data packets. The latency of the trace packet is calculated from when it first enters the network until it 'wins' a competition; to this is added the nominal transfer time. Figure 9 provides a sketch of the model.

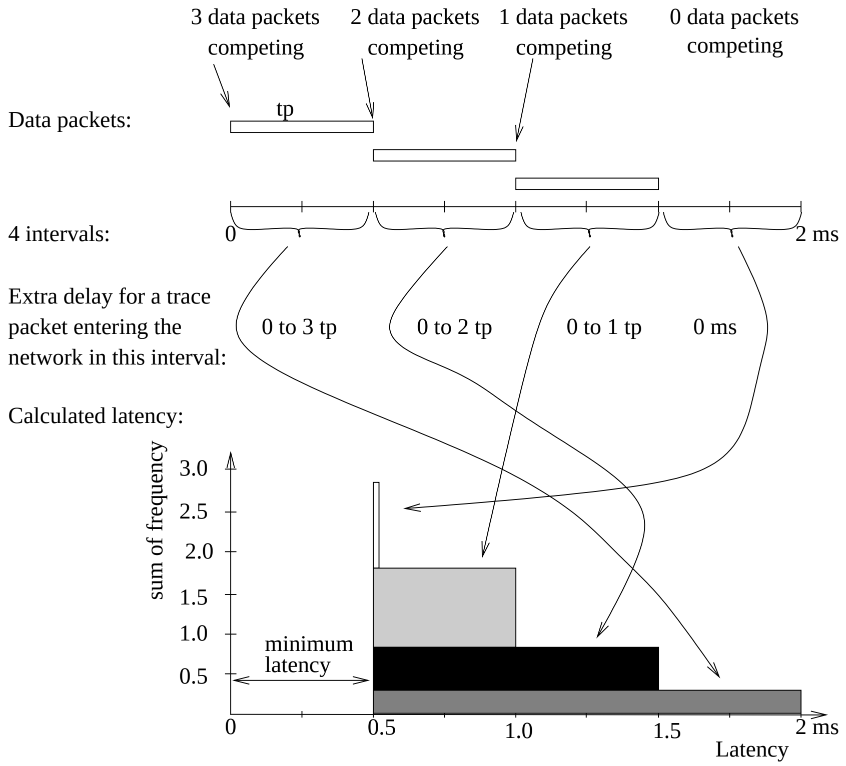

Figure 9: Sketch of how the latency model works. In the figure the event has 3 fragments, and there is no overlapping event. The trace packet can arrive at any time in the four intervals. If it arrives during the last interval it can be transfered across the network with the minimum latency; giving rise to the first peak. If the trace packet arrives in the first interval it has 1/4 chance of getting transferred first, second, third or last; giving rise to the base in the calculated latency. The latency histograms are calculated for each possible number of fragments and then summed with the appropriate weights (see Figure 3).

The results from applying the model to the emulated networks are shown in Figure 10. The model reproduces the shape of the latency distributions, both for different networks and at different frequencies. The local calorimeter and muon networks could not be modelled in this simple way because they have several sizes of event fragment packets.

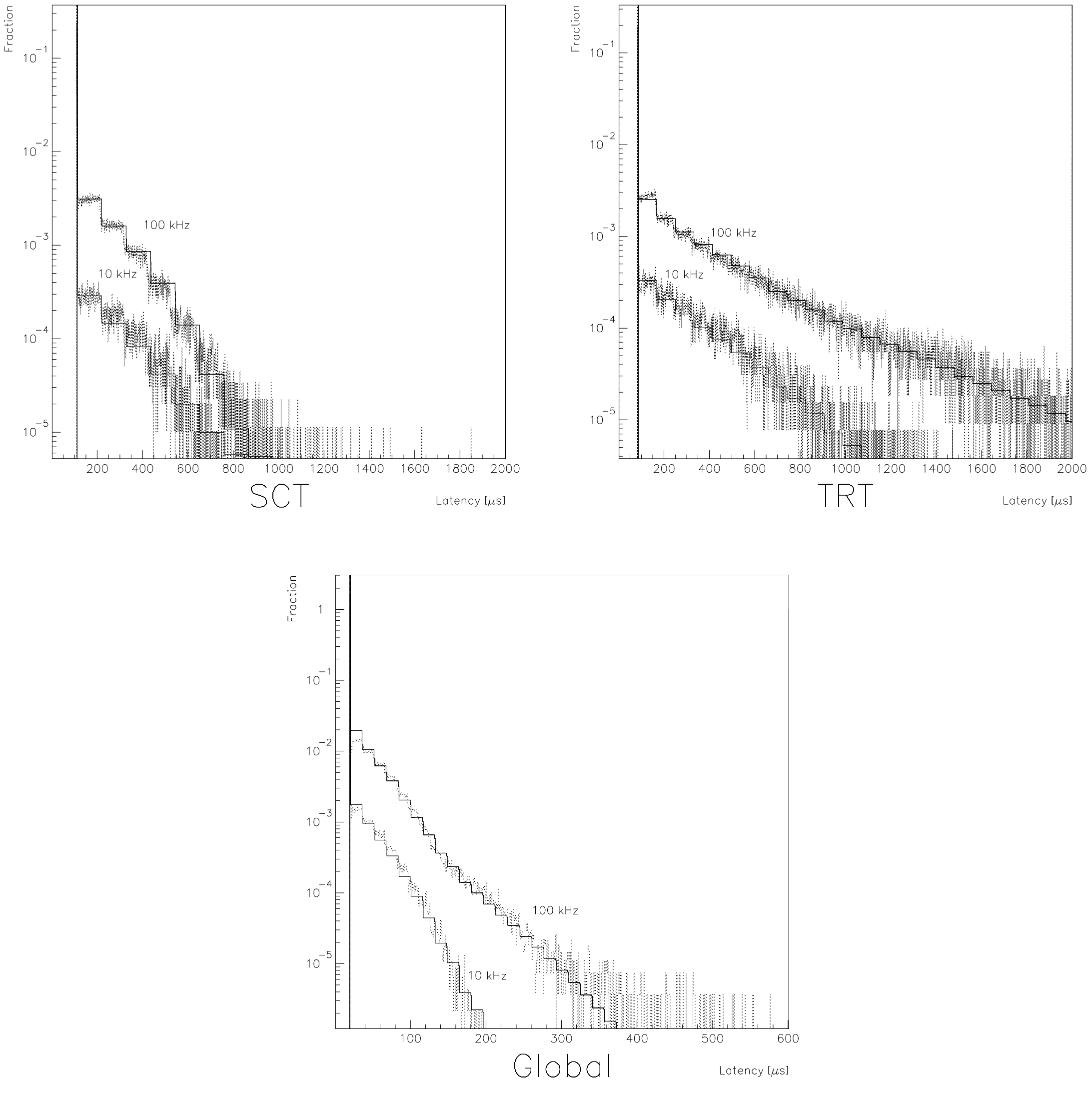

Figure 10: Latency with fit for the SCT, TRT and global networks with distributed mapping.

7 Conclusions

The emulation studies of the ATLAS second level trigger, architecture B on the Macramé Testbed have shown that distributing the sources and sinks evenly over the network rather than grouping them is important for enhancing the usable bandwidth of the network. Emulating the calorimeter network showed that decreasing the hit rate may be necessary to improve the throughput and thereby the sustainable event rate. The latency distribution can be understood by a simple model for how data blocking can occur throughout the switching network.

The local muon and SCT networks and the global network were emulated full size. Substantial fractions of the TRT(80%) and the calorimeter(50%) networks were emulated, so that only a limited extrapolation is necessary for scaling to full size. The Clos networks have been shown to scale for systematic traffic from 64 nodes up to 512 nodes using Macramé[3], and theoretical considerations of the Clos topology show it should scale further[11].

Finally, it should be noted that the emulation of the calorimeter local network ran at an event rate of almost 80 kHz when multiple links for the hadron calorimeter read-our buffers were used. The other networks ran with an event rate of 125 kHz.

8 Acknowledgements

A special thanks to J. Vermeulen for providing the drive files and to R. Hauser for a library to read them with. Also thanks to the demonstrator B community for valuable discussions and support.

References

[1] Atlas Level-2 Group. Options for the atlas level-2 trigger. In *International Conference on Computing in High Energy Physics*, April 1997.

[2] Standard for Hetrogenious Inter-Connect (HIC). Low-Cost Low-Latency Scalable Serial Interconnect for Parallel System Construction, 1995.
`http://www.1355-association.org/`.

[3] M. Zhu et al. Realisation and performance of ieee 1355 ds and hs link based, high speed, low latency packet switching networks. In *Beaune 97 X^{th} Real Time Conference*, September 1997.
`http://www.cern.ch/HSI/dshs/`.

[4] R.W. Dobinson et al. Realisation of a 1000-node high-speed packet switching network. In *Proceedings of the International Symposium on Problems of Modular Computer Systems*, June 1995. `http://www.cern.ch/HSI/dshs/`.

[5] C. Clos. A study of non-blocking switching networks. *Bell Syst. Tech. J*, 32:406–424, 1953.

[6] R. Bock and P. LeDu. Detector and readout specifications, and buffer-roi relations, for the level-2 trigger demonstrator program. Atlas Internal Note; DAQ-No-062, Jan 27 1997.

[7] J.R. Hubbard S. George and J.C Vermuelen. Input parameters for modelling the atlas second level trigger. Atlas Internal Note; DAQ-No-070, June 12 1997.

[8] J.A. Strong M. Dobson, S. George and J.C Vermeulen. Paper models of the atlas second level trigger. To be published as Atlas Internal DAQ Note.

[9] J.C. Vermeulen. Simdaq documentation. Private communication.

[10] J.C. Vermeulen. Calculation of the number of robs per roi and of maximum and minimum roi request rates relative to the average rate. http://www.nikhef.nl/www/pub/expriments/atlas/-daq/ROBspreRoI.html, March 27 1997.

[11] A. Klein. Interconnection network for universal message-passing system. In *Proceesings of Esprit Conference '91*, pages 336–351, Nov. 1991.