ECOLE: A Configurable Environment for a Local Optical Network of Workstations

Gordon Brebner and Rob Pooley

Department of Computer Science
University of Edinburgh
James Clerk Maxwell Building
Mayfield Road
Edinburgh EH9 3JZ
Scotland
E-mail: {gordon,rjp}@dcs.ed.ac.uk

Abstract. ECOLE is a project concerned with the implementation of highly efficient parallel computation on a network of generic workstations connected by a very high speed optical LAN. It is founded on research conducted at BT Laboratories, which has resulted in SynchroLan — a multi-gigabit LAN billed as the fastest LAN in the world. A holistic approach to system design is taken, to identify all possible hardware and software improvements that might allow a workstation to harness fully the raw capacity of a multi-gigabit network. Two important features of the work are the inclusion of configurable hardware and the use of active protocols — both of these are to introduce overall flexibility, without compromising efficiency by more than a small amount. The work benefits from the use of advanced techniques for modelling and simulation of experimental architectures. This paper explains how the ECOLE project acts as a very practical focus for several existing lines of advanced systems research being undertaken at the University of Edinburgh.

Keywords:
NOW, optical LAN, configware, active protocol, modelling and simulation.

1 Introduction

The emergence of low cost and high performance workstations and interconnection networks points to the use of network of workstation (NOW) architectures as a compelling alternative to expensive parallel supercomputers. Ideally, NOW architectures would employ totally off-the-shelf workstations and networking. However, the deliberate generality built into these components leads to greatly reduced performance when attempting to compete with supercomputer speeds. To improve the situation, a holistic approach to systems is desirable, reviewing the efficiency of all system components: workstation and networking hardware and software, and also interaction using communication protocols. It is desirable that the resulting systems do not completely sacrifice generality for efficiency, to allow support for a variety of applications with different needs.

The ECOLE (Edinburgh Configurable Optical LAN Environment) project seeks to apply in tandem several currently active areas of advanced systems research at the University of Edinburgh and BT, to provide novel solutions to the problem of implementing efficient, but programmable, NOW architectures based on off-the-shelf workstations interconnected by a very high speed optical local area network. This paper introduces the different research areas, and indicates how their results contribute to the overall aim; future papers will report on system construction and experimentation. The three most central research threads are:

- SynchroLan: an optical LAN demonstrated by BT Laboratories in 1996/97, and rated as the world's fastest;
- Configware: novel solutions to the problem of integrating soft circuitry based on Field-Programmable Gate Arrays (FPGAs) into total systems, particularly with System Level Integration (SLI), also being investigated at Edinburgh University; and
- Active protocols: a flexible and verifiable approach to protocol construction and use, being developed at the Edinburgh University.

and the experimental application of these is much enhanced by research into:

- system modelling using Stochastic Process Algebras; and
- system simulation of high-bandwidth networks and hierarchical architectures.

To complete the picture, the combined research can draw upon the skills of researchers involved in past development of pioneering and parsimonious operating systems for both mainframe computers and workstation computers. There is also ready access to the state-of-the-art supercomputers (e.g., a Cray T3D) of the Edinburgh Parallel Computing Centre (EPCC), allowing comparative benchmarking, as well as providing conventional brute-force computational power for simulation studies.

A technical impetus for the ECOLE project comes just from the raw communications technology. SynchroLan is already able to deliver data rates of 2.5 Gbit/second to workstations, and the technology allows upgrading to at least 40 Gbit/second transfer rates. The problem is that nothing like such rates can currently be delivered to the applications running on workstations. As many other researchers in this area have noted, a main bottleneck lies in the network interface, since current interfaces are not specifically designed for frequent, high bandwidth communication. Apart from this bottleneck, the communications protocols used over the network are another main source of inefficiency. Given good solutions to these two main problems, further gains can be had from holistic improvement of system architectures.

The remainder of the paper is arranged as follows. Section 2 focuses on the network interface, specifically on how the configware research can bring novel benefits. Section 3 focuses on the communication protocols, specifically on how the active protocols research can bring novel benefits. This paper does not include a more detailed description of SynchroLAN, since its operation is not central here; more information can be found in [8, 12, 16, 17]. Section 4 reviews the modelling and simulation technology which will be brought to bear on the problem of designing and experimenting with new system architectures. Finally, there is a brief review of related work, and an outline of the future work programme of the ECOLE project.

2 Configware and the network interface

The overall organisation of the experimental SynchroLAN testbed that will be used for the ECOLE project is shown in Figure 1. Although the testbed in

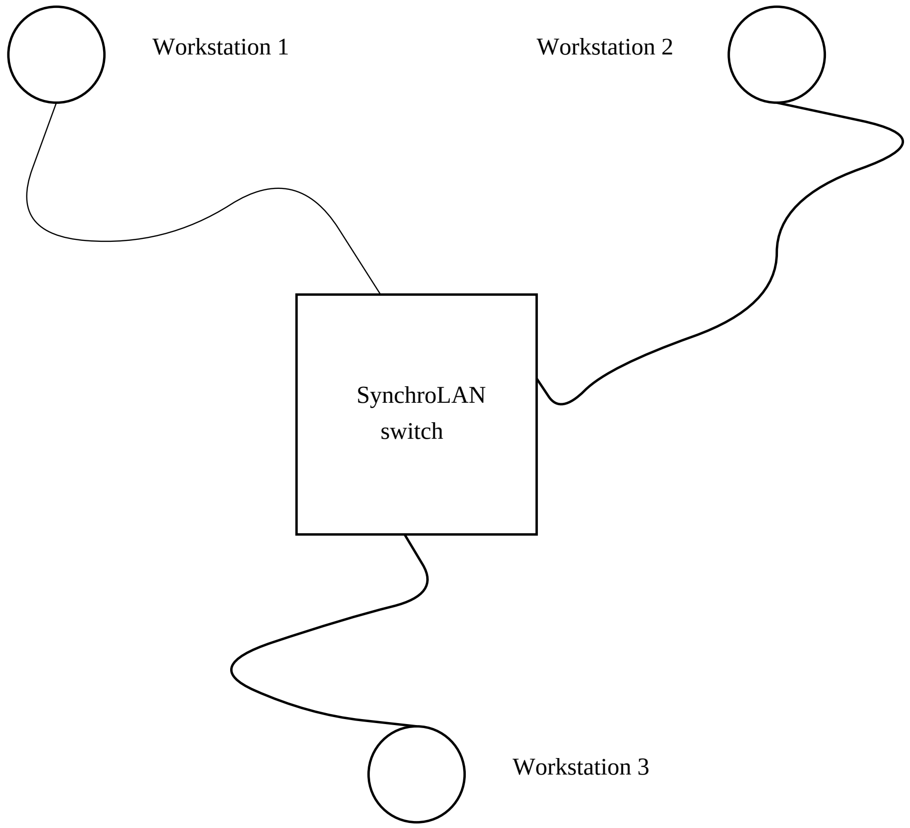

Figure 1. Experimental testbed with SynchroLAN

housed in one laboratory, the fibre optic cabling is routed round the entire building, in order to emulate correctly what would happen in a real local area network.

Research at BT Laboratories is currently targeted at designing a single card that contains all of the components needed for the physical optoelectronic interface. The aim of the ECOLE project is to ensure that any extra hardware components needed to support an efficient, but flexible, NOW architecture can also be fitted on this single card, to be consistent with providing a low-cost solution. The card must interface to the system memory bus of the workstation, to provide the fast interaction with the main processor and the main memory that is needed to support application programs. The overall architecture is illustrated in Figure 2.

For ECOLE, low-cost workstations built from PC components will be used, with direct access to the system memory bus being made possible via the Accelerated Graphics Port (AGP) available with the latest Intel 440LX chipset for Pentium II PCs. This allows a point-to-point channel between a graphics controller and the system RAM, which can be exploited by a communications controller in ECOLE. The current generation of AGP interface (AGP 2X) offers a data rate of 533 MB/second, with a rate in excess of 1 GB/second expected in the future from the APG 4X. These rates are acceptable for the ECOLE prototyping work, the project being deemed a success if they can be fully harnessed.

In the short term, there are two approaches to ensuring that the extra hardware can be accommodated on the same card as the optoelectronic interface. The first is to design a custom ASIC once a solution has been selected from the possible design space; for example, this is the line pursued by the Utah Avalanche project [9]. The second is to include one or more Field Programmable Gate Array (FPGA) chips that can be configured with required circuitry. While the second option is more flexible, and allows prototyping, the performance is likely to be slower and the available number of gates will be rather smaller. For ECOLE, a longer-term perspective is being taken, strongly influenced by very active work on System Level Integration (SLI) at Edinburgh University.

With SLI (alternatively known as 'System On Silicon'), a number of major system components are located on a single chip. This is made possible by recent advances in deep submicron fabrication technologies. The complexity of SLI circuitry poses many challenges for the designer, and therefore much research is being carried out into new design methodologies and tools. One initiative is the Virtual Socket Interface (VSI) [24], being developed within a consortium called the VSI Alliance; this is concerned with providing a standard way to integrate 'virtual components' from different sources, on a single chip. There are three major types of virtual component:

- processor cores (microprocessors, DSPs);
- memory devices; and
- configurable logic arrays.

and these can be integrated with more specialised types of component, such as analogue-digital convertors.

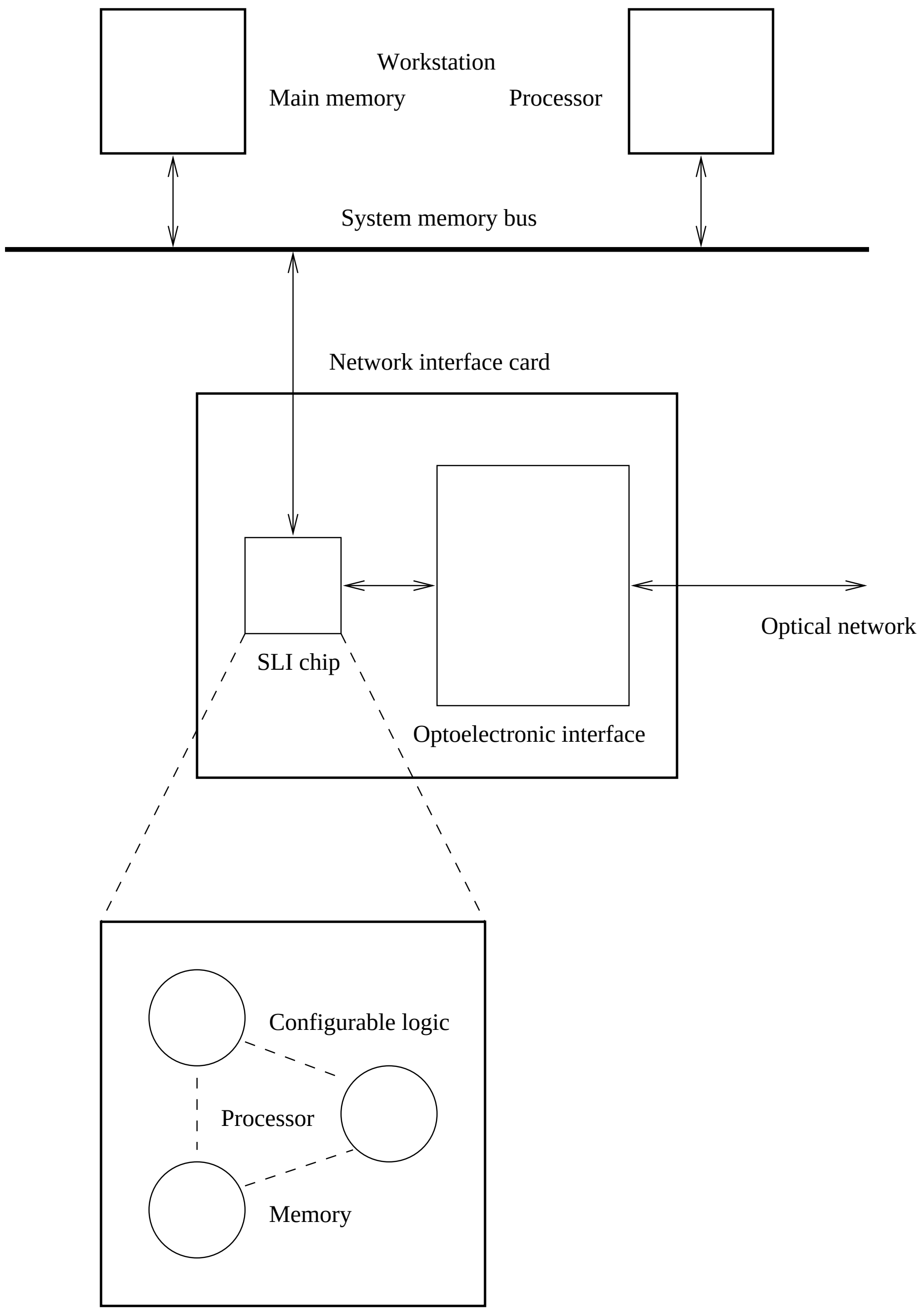

Figure 2. Overall network interface architecture

Apart from the obvious generic speed benefits of having processors, memories, etc., interfaced at chip level, the advent of SLI is particularly promising for the use of configware — configurable hardware — in this project. At present, configware is severely constrained by the pin interfaces to FPGA chips, or by the bus interfaces to boards containing FPGA chips. This penalises not only the speed-ups obtained when using configware as a configurable co-processor but also the time taken to reconfigure configware to change its co-processor function. SLI gives scope for rather more intimate, and fast, configware-processor-memory interactions.

For the ECOLE project, the presence of configware in the network interface is a central feature, in order to add flexible hardware support for communication protocols. This can be useful both for different network environments, for example, different latencies or reliabilities, or for different application requirements. The latter might be variations in model, for example, shared memory, message passing or bit pipe, or in quality requirements. In Section 3, the use of configware for protocol implementation is illustrated by an extended example.

Previous configware research at Edinburgh has focused on how a configurable logic array can best be managed by a processor, resulting in general notions of 'virtual circuitry' — the configware analogue of virtual memory [3, 4]. The fruits of this work gain particularly from SLI, given an integrated processor running a lightweight virtual circuitry operating system. One particular application of virtual circuitry that is currently being investigated results from a convergence with the research described in Section 3. This is where the circuitry implements accelerator components for communication protocols [10]. The general technology developed can be directly applied to particular protocols devised within the ECOLE project.

In summary, configware is not just included as a useful means of prototyping or implementing glue logic — the typical current uses of FPGA technology — although both of these are certainly of use here. It is included to provide support at hardware speeds for dynamically tailoring protocols to their operating environment. The use of SLI in the longer term should deliver satisfactory performance, as well as the means for enabling a single-card network interface.

3 Active protocols and the network communication

There is general agreement that the use of a standard protocol stack, such as TCP/IP, is inappropriate for a NOW architecture because of its relative inefficiency. This is hardly suprising, since TCP/IP was designed for use over any type of networking, of any quality. Just as general-purpose workstation architectures hinder the quest for speed, so do general-purpose communication protocols. As a result, a number of special-purpose protocols have been devised, each one usually associated with a particular NOW architecture. The problem with this approach is that it results in an inflexible environment that does not allow protocol adaptation to deal with different requirements between applications.

To obtain more flexibility of protocols, research on 'active protocols' at Edinburgh University is a key ingredient of the ECOLE project. An active protocol is one which can be constructed and distributed dynamically. The idea is associated with the active network concept [23], which is currently attracting much attention in the networking world. In an active network, packets can contain executable code as well as just processable data, and this code can be executed by intermediate nodes within the network. Active protocols are defined in terms of a few primitive and orthogonal protocol elements, based on the communication principles described in [3]. They can be distributed in an active network using messages that contain executable forms of the protocol description, that is, appropriate packet-handling instructions. As well as practical benefits, the nature of active protocols makes them amenable to rigorous demonstrations that protocols and protocol stacks have the properties required by their users. This is very important in situations where protocols are being devised rapidly, and are not subject to many years scrutiny by standardisation committees.

The use of protocol elements within an active protocol is best illustrated by an example. Consider end-to-end flow control between two communicating parties. This is a feature of many different protocols, with a few standard mechanisms appearing in various different disguises, and often in combination with other protocol features, for example, error correction. The active protocol framework treats flow control as an individual abstract function, and provides mechanisms whereby this function can be combined in a practical implementation with other functions for efficiency reasons. In this example, an abstract solution based on sliding windows will be considered.

The basic idea is that, when information is transmitted, there is an associated sequence number that indicates the place of the information within some (possibly pseudo-infinite) information stream. The receiver controls the flow by informing the transmitter of its current position within the information stream; in general, this is behind the current position of the transmitter. The transmitter is obliged to ensure that the receiver does not fall too far behind. Two possible ways of running the scheme are:

1. the receiver periodically automatically sends back sequence numbers indicating its current position (common in many protocols);
2. the transmitter periodically asks the receiver to send back a sequence number indicating its current position (used in some protocols, for example, the ATM Service Specific Connection Oriented Protocol (SSCOP)).

In both cases, the protocol element at the transmitter sends sequence numbers corresponding to information transmitted. In the first case, the protocol element at the receiver periodically sends back sequence numbers as information is consumed. In the second case, the protocol element at the transmitter periodically sends 'marked' sequence numbers corresponding to information transmitted, and the protocol element at the receiver responds by sending back a sequence number corresponding to information received and consumed. Note how this abstracts away unnecessary detail to give the fundamental protocol operation.

It can then be used as a building block for more complex protocols, with the abstract form being useful for verifying that the construction achieves the desired effect. The implementation of this example will be considered further, after the next paragraph.

The active protocol work is being linked with the configware work through the notion of 'circlets' (small circuits, roughly the hardware equivalent of software applets). The virtual circuitry research provides an operating system mechanism for the management and use of a collection of circlets; one application of this is to the case where active protocols are represented by circlet descriptions carried in packets. While the mobility of active protocol circlets is not likely to be useful in a NOW architecture (since it appears more efficient to have a small collection of defined protocols, pre-distributed statically), the capability for defining protocols and then deriving executable circlets is exactly what is required to harness the configware described in Section 2.

Now, continuing the sliding window flow control example, it is possible to sketch an implementation based on circlets. A first point to make is that circlets can be parameterised, to create families of circuits. Here, a natural parameterisation is on the number of bits used for sequence numbers, so a family of circuits reflects a family of protocol elements. The basic circlet layout is shown in Figure 3. It has one register (to the right) used for storing a received sequence

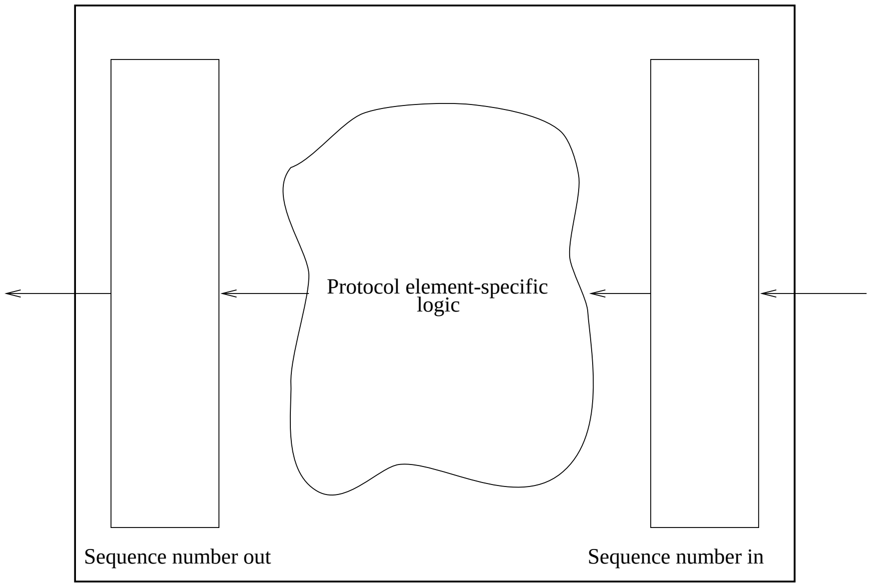

Figure 3. Basic circlet layout

number, and another register (to the left) used for storing a transmittable or transmitted sequence number. The height of these registers (and hence of the circlet) is the parameterised feature. The central logic is specialised to give the correct effect, depending on protocol and depending on whether the circlet is being run at the information transmitter or the receiver. The circlet then fits within a surrounding system to implement the context of the protocol element, and this may either involve other circuitry supplying the inputs and using the outputs or some program that delivers the inputs and collects the outputs.

For the transmitter protocol element, the circlet input corresponds to a sequence number sent back by the receiver and the circlet output corresponds to a sequence number sent to the receiver. For the receiver protocol element, the circlet input corresponds to a sequence number sent by the transmitter and the circlet output corresponds to a sequence number sent back to the transmitter. In all cases, the key operation of the central logic is to determine when, and with what value, the output register is updated.

At the transmitter, the essential decision required is that there is information to transmit, and that the values contained in the two registers are not further apart than the current sliding window size.

In the case where the receiver automatically sends back sequence numbers, the central logic at the receiver must update the output register when information has been consumed, possibly tempered by the two registers being further apart by more than some threshold and/or that some time period has elapsed since the consumed information was received. In the case where the transmitter solicits a response, the central logic at the receiver is simplified so that the output register is updated precisely when a marked sequence number arrives. The central logic at the transmitter is more complex, however, since it must decide when to send a marked sequence number, on the basis if the two registers being further apart than some threshold and/or that some time period has elapsed since information was transmitted.

A significant point to note from the above is that only a few different operations occur in the different variants of the central logic:

- sensing new information to send at the transmitter
- sensing new information consumed at the receiver
- comparing the difference of the two register values against a threshold
- running a watchdog timer.

This simplifies the creation of a range of circlets, and fits with the general philosophy of simplicity and flexibility of circuitry. This example has shown how the decomposition of protocols into simple protocol elements facilitates implementation using circlets with desirable properties.

Of course, an important concern is to ensure that introducing the generality of active protocols does not inflict new speed penalties that cancel out the benefits gained from some specialisation of the system architecture and the protocol sets. This should not be the case, since the use of circlets is targeted at various high speed networking operations, fast packet switching for example, that demand very efficient performance.

In summary, active protocols are involved both to provide a convenient framework for design and testing of novel protocols and to provide an implementation route that allows a collection of protocols to be dynamically instantiated on configware.

4 Modelling and simulation of novel architectures

Even with the flexibility in prototyping provided by the configware approach, developing and proving designs by physical prototypes is time consuming and may not deliver sufficient detailed information about bottlenecks, performance and other characteristics of an architecture. It is particularly important to identify serious flaws early in the design process and then to concentrate on only those designs worthy of detailed study.

Edinburgh has considerable expertise in two, complementary, methods for such studies, *stochastic process algebra* and *hierarchical discrete event simulation*. Recent developments in both techniques and tools have greatly increased the power of these techniques, particularly when used in combination, to address this type of design problem [22, 15].

It is a well known property of architectures that many decisions can be made at a high level of abstraction, independently of the detailed working of components. It is, unfortunately, also well known that the detailed working of components may introduce further, equally significant, problems and that these may only become apparent when the full, detailed design is tested. In system testing these are addressed by high level, unit and integration testing respectively. In using performance modelling, a similar strategy is needed. It can be achieved by modelling in a component based manner, using process algebras to provide a hierarchical framework for integration which is fully understood.

4.1 High level modelling

Early modelling of outline designs is greatly facilitated by the use of suitable tools. Edinburgh has considerable experience in developing and using such tools for architectural simulation. By developing models in terms of components, where descriptions can be refined as more details become available, a consistent approach is guaranteed.

Along with the development of outline simulation models, a more formal modelling approach is possible, based on process algebras [20, 21]. This allows equivalent behavioural models, capable of revealing deadlocks and other logical flaws, to be generated from the schematic version of an architecture. This also allows further consistency checking as the design is refined. By enhancing these models with performance information, numerical solutions can also be obtained [11].

4.2 Unit modelling of components

Pre-solution of component models can be used to avoid the complexity of solving a detailed design model. This may then allow these results to be reused within a simpler overall model. The use of such techniques depends on a mixture of understanding of stochastic modelling and behavioural equivalence. Again, work at Edinburgh has shown how such aggregated models can be safely derived, using stochastic process algebra as a framework.

At the level of component modelling, conventional discrete event simulation techniques are generally adequate. It is important, however, to take advantage of well defined experimental techniques, where multiple runs can be used in parallel to explore a design space [14]. The notion of experimental support in a distributed computing environment is a key to effective use of simulation in design.

4.3 Integration modelling of system level designs

Use of a well-understood basis for structuring and unit solving of designs makes the testing of complete systems feasible. Since aspects of component behaviour which are not capable of exerting a global influence on the design will have already been isolated, they can be safely ignored in the complete model. This, combined with the same experimental methodology used in component level simulation, allows a thorough exploration of a complete design, and a good understanding of the results.

Since the ECOLE project is able to exploit state of the art high performance computing, such as the Cray T3D, in the more demanding stages of the work, it will be possible to have considerable confidence in any candidate designs before moving to physical prototyping and testing.

5 Related work, and the future ECOLE work programme

The ECOLE project is one of many that are looking the problem of achieving fast parallel computation using a network of workstations. The adoption of a holistic approach is a central feature, something that is relatively uncommon in other research, one notable exception being the Utah Avalanche project [9]. A major physical difference from Avalanche and other work is that a next-generation general purpose LAN forms the interconnection mechanism, rather than an interconnect such as Myrinet [2]. This is to ensure that workstations can still be used in their more conventional independent manner, as well as just in a NOW architecture.

Another physical aspect of ECOLE is that normal workstations are used, in contrast to more specialised architectures of other work, such as the MIT Alewife [1] or the Stanford FLASH [13] architectures. Indeed, the choice of a PC-based architecture is targeted at the most commonly available workstation — and one that can deliver much-improved performance these days, rather than the HP multiprocessor workstations or Sun SPARCstations used in other work, for example Avalanche, HP Lab's Hamlyn [7] or Illinois Fast Messages [19]. The aim is to add just one extra card, responsible both for physical network interfacing and for tailored hardware/software functions inside the network interface.

A final innovative physical aspect is the inclusion of configurable logic as an integral part of the network interface design, to provide scope for dynamic configurability of hardware. This has not been a feature of other NOW research, due to the relative inflexibility of much current FPGA hardware and to the speed penalties compared with custom ASICs. ECOLE seeks to benefit from emergent improvement in configurable logic technologies that will greatly reduce current performance penalties.

Reduced-weight communication protocols are also important to the holistic approach of the ECOLE project. These have also featured in other NOW research, for example, the DD protocol of the Avalanche project [9] or the PRP protocol of the PARMA project [18]. Gains over more conventional protcols, such as TCP/IP, come from introducing application specificity and from simplied physical channel assumptions, such as error freedom or perfect sequencing. For ECOLE, the framework of active protocols offers a way of experimenting with new stripped-down protocols to find those that fit the application/physical environment efficiently. Moreover, the configurable interface gives a means to avoid the need of selecting a single protocol for all uses.

The work plan of the ECOLE project involves extensive use of modelling and simulation of design alternatives, before any prototyping using PC-based workstations and the SynchoLan optical LAN. The first stage of the holistic approach involves an examination of high-level architectural design alternatives, together with alternative communication protocols. The second stage involves examining major system components. Two particular aspects of this are an investigation of the uses of reconfigurable logic as part of an overall architecture, and an investigation of improved software architectures to support applications. The third stage involves prototyping of chosen designs, by which time it is intended that SLI chip facilities will be available to allow a complete system implementation.

Acknowledgement

The authors thank BT Laboratories, particularly David Cotter, for inspiring this work, and for promising access to experimental optical network prototypes.

References

1. Agarwal A., R. Bianchini, D. Chaiken and K. Johnson, "The MIT Alewife Machine: Architecture and Performance", Proc. 22nd Annual International Symposium on Computer Architecture, June 1995, pp.2–13.
2. Boden N. et al, "Myrinet — A Gigabit-per-second Local Area Network", IEEE MICRO Vol 15 No 10, February 1995, pp.29–36.
3. Brebner G., *Computers in Communication*, London:McGraw-Hill 1997.
4. Brebner G., "A virtual hardware operating system for the Xilinx XC6200", Proc. 6th International Workshop on Field Programmable Logic and Applications, Darmstadt, September 1996, Springer LNCS 1142, pp.327–336.

5. Brebner G., “The Swappable Logic Unit: a Paradigm for Virtual Hardware”, Proc. 5th Annual IEEE Symposium on Custom Computing Machines, Napa, April 1997, IEEE Computer Society Press.
6. Brebner G. and S. Haeck, “Active Protocols”, in preparation for submission to SIGCOMM’98.
7. Buzzard G. et al, “An Implementation of the Hamlyn Sender-managed Interface Architecture”, Proc. 2nd Symposium on Operating System Design and Implementation, October 1996.
8. Cotter D., J. Lucek and D. Marcenac, “Ultra-High-Bit-Rate Networking: From the Transcontinental Backbone to the Desktop”, IEEE Communications Magazine, Vol 35 No 4, April 1997, pp.96–102.
9. Davis A., M. Swanson and M. Parker, “Efficient Communication Mechanisms for Cluster Based Parallel Computing”, Proc. 1st Internation Workshop on Communication and Architectural Support for Network-Based Parallel Computing, San Antonio, February 1997, Springer LNCS 1199, pp.1–15.
10. Donlin A., “A Dynamically Self-Modifying Processor Architecture and its Application to Active Networking”, Internal report, Department of Computer Science, University of Edinburgh, September 1997.
11. Gilmore S. and J. Hillston, “The PEPA Workbench: A Tool to Support a Process Algebra-based Approach to Performance Modelling”, Proc. 7th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation, Vienna, May 1994, Springer LNCS 794.
12. Gunning P., J. Lucek, D. Moodie, K. Smith, D. Pitcher, Q.Badat and A. Siddiqui, “40 Gbit/s optical-TDMA LAN over 300m blown fibre”, Proc. 23rd European Conference on Optical Communications, Edinburgh, September 1997, IEE Conference Publication 448, IEE London, pp.61–64.
13. Heinrich M. et al, “The Performance Impact of Flexibility in the Stanford FLASH Multiprocessor”, Proc. 6th Symposium on Architectural Support for Programming Languages and Operating Systems, October 1994, pp.274–285.
14. Hillston J., A. Opdahl and R. Pooley, “A case study using the IMSE experimentation tool”, Proc. 3rd Conference on Advanced Information Systems Engineering, Trondheim, Springer LNCS 498, pp.284–306.
15. Ibbett R., P. Heywood and F. Howell, “HASE: A Flexible Toolset for Computer Architects”, Computer Journal, Vol 38 No 10, 1995.
16. Lucek J., D. Cotter, K. Smith and P. Gunning, “Ultrafast Photonic Data Networks” (Invited Paper), IEEE LEOS’96 Annual Meeting, Boston, 18–21 November 1996.
17. Lucek J., P. Gunning, D. Moodie, K. Smith and D. Pitcher, “SynchroLan: A 40 Gbit/s optical-TDMA LAN”, Electronics Letters, Vol 33, 1997, pp.887–888.
18. Marenzoni et al, “An Operating system Support to Low-overhead Communications in NOW Clusters”, Proc. 1st Internation Workshop on Communication and Architectural Support for Network-Based Parallel Computing, San Antonio, February 1997, Springer LNCS 1199, pp.130–143.
19. Paikin S., M. Lauria and A. Chien, “High Performance Messaging on Workstations: Illinois Fast Messages (FM) for Myrinet”, Proc. Supercomputing ’95, San Diego, December 1995.
20. Pooley R., *Formalising the Description of Process Based Simulation Models*, Ph.D. Dissertation, University of Edinburgh, 1995.
21. Pooley R., “Integrating Behavioural and Simulation Modelling”, Proc. 8th International Conference on Techniques and Tools for Computer Performance Modelling, Heidelberg, September 1995.

22. Pooley R., "Exploiting Functional Properties to Reduce Complexity in Simulating ATM", Proc. Telecommunication, Distribution and Parallelism (TDP '96), La Londe les Maures, France, June 1996, pp 23–34.
23. Tennenhouse D. and D. Wetherall, "Active Networks", Proc. 15th Symposium on Operating System Principles, December 1995.
24. VSI Alliance Architecture Document, VSI Alliance 1997.