Board for Realizing Active Intelligence in Networks (BRAIN)

Joe Touch
(touch@isi.edu)

May 2, 1997[1]
USC / Information Sciences Institute

Abstract

Current networking research efforts, such as Active Networks and high-performance router development, rely on emerging high-speed switching technology, but can also require programmable processing resources that switches lack. Here we present our vision of a board for realizing this resource for switches, such as ATM. The BRAIN board is a switch- and link- independent processor that provides a pipelined CPU as well as a prototyping area for custom hardware. It takes advantage of recent commodity workstation bus architecture together with existing network interfaces to provide a port processor independent of the underlying networking technology.

1: Introduction

High-speed network research has been focusing on the application of high-performance switches. Such switches form the backbone of ATM networks, and the 'backplane' of recent routers [11]. Alone, switches are insufficient for either routers or emerging programmable networking systems, such as Active Networks (AN) [17]. These systems require a high-performance processor that can keep pace with gigabit link rates.

Here we present our vision of BRAIN, a Board for Realizing Active Intelligence in Networks. Our goal is to design and implement a number of BRAIN boards, to distribute them to the networking research community, and to coordinate their use to enable high-performance programmable networking research.

The BRAIN board is a programmable processor intended to augment switching fabrics, although it does not rely on a particular link or network technology. Instead, BRAIN takes advantage of recent advances in commodity workstations, together with available network interface cards, to provide a processing resource that is not limited by host backplane bandwidth contention.

This document describes our vision of the BRAIN board. First we discuss the processing requirements of network switches. We then present our design goals, and describe the BRAIN board architecture. We compare the BRAIN to alternate designs, and finally discuss our target for implementing and using the BRAIN boards.

2: Processing requirements

Active Networks programs require programmable processing resources inside the network, typically at routers [17]. By contrast, current router designs minimize processor interaction with packets, using dedicated hardware to perform common-case routing [11], [16]. Even emerging router designs based on general-purpose CPUs, *e.g.*, BBN's Multigigabit Router, assume the CPU is dedicated to the common-case algorithm to achieve high performance [2]. Switch-based designs eliminate programmable processing, providing only programmable signalling and configuration management. In each case, a general-purpose programmable processing resource is lacking.

Programmable processing can be added inside a router or switch by placing a CPU near each input or output link (Figure 1). This requires the modification of the switch hardware, which is typical proprietary. A notable recent exception is the NSF-sponsored Gigabit Network Technology Distribution Program at the Univ. of Washington in St. Louis, wherein the switch architecture is openly documented and public [9]. The GNT program will implement and distribute a number (40) of 'switchkits', including a high-speed ATM switch and some PCI host interfaces. However, even in the GNT case, custom, architecture-specific hardware is required to support line-rate processing of cell- or packet-level data.

A typical alternative to these intra-router/switch processors is to use hosts as in-link processors (Figure 2). This solution assumes that the host has both sufficient processing power and sufficient communication bandwidth to both input and output data. For per-packet operations, *e.g.*,

[1] *This document was prepared and originally distributed informally in 1997; it has been prepared as an ISI Research Report in 1998, in its original form, for archival purposes.*

header-based operations, both conditions are met, because the rate of basic operations is dependent on the header rate, rather than the data rate. Furthermore, only header information need be pipelined through the host, reducing the bandwidth requirements.

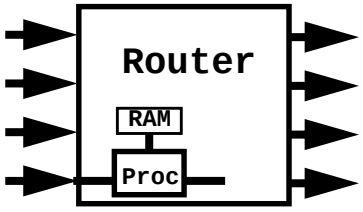

FIGURE 1. Intra-router/switch processing vs. host-based processing

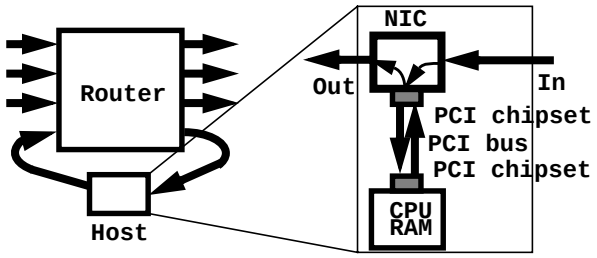

FIGURE 2. PC-based processing

Emerging network capabilities such as encryption, or Active Networks packet data functions such as transcoding or data fusion, tax the capabilities of a host-based solution [3], [4], [19]. For processing-bound functions, multiprocessor hosts may suffice. However, when the entire packet data must be processed inside the host, communication bandwidth can be insufficient. The 32-bit PCI backplane in current hosts supports bandwidths of 1.056 Gbps, supporting OC-3 (155 Mbps) links, but cannot support a fully-loaded OC-12 (622 Mbps) or either GNT (1.2 Gbps) or Myrinet links (640 Mbps-1.28 Gbps [13]), because the PCI bus must support the full line rate in both directions simultaneously (Figure 3, 2-cross). The BRAIN board supports full PCI bandwidth for link processing, supporting a most of the link bandwidths of these new technologies (Figure 3, 1-cross).

This analysis also assumes 100% utilization of the PCI bus, and that no other peripherals use that bus, if this assumption is false the available bandwidth is further limited. A recent example uses a host and host-interface to monitor a 1.2 Gbps (OC-24) link [14]. In this system, only packet headers and cell counts can be logged, due to the bandwidth limits of the host.

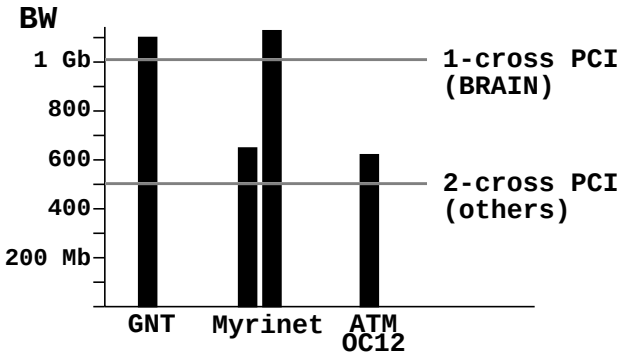

FIGURE 3. Configuration bandwidth limits

2.1: The GNT Program

The Univ. of Washington in St. Louis will implement and distribute a number of ATM switchkits, under the NFS-sponsored Gigabit Network Technology Distribution Program. During a pre-award meeting of potential recipients of these switchkits, we raised the issue of the need for a programmable processing resource as part of a complete platform to support Active Networks research.

The GNT distribution kits include an ATM switch and a number of host interface cards (NICs). The configuration of the switch and the specification and architecture of the system are open and public, permitting user experiments which proprietary ATM hardware precludes. The GNT kit supports experiments in alternatives for ATM signalling and OS integration of NIC management.

A number of attendees indicated a desire for a programmable processing resource, which could operate on packet or cell data at near full line rates, *i.e.*, 1.2 Gbps (OC-24 equivalent) for the GNT links. This would require either custom GNT hardware modifications to include an on-board, per-port CPU, or a processor that could be inserted 'inside' a link. The BRAIN board provides a link-independent implementation of the latter.

3: The BRAIN board

The BRAIN board is designed to provide a programmable processing resource that overcomes the host-backplane bandwidth limit, while providing sufficient computing power to support transcoding and data fusion. The design also supports custom hardware, both in the form of PLDs and custom chips. A major design goal is to achieve these results independent of a particular link, switch, or router technology, which implies that it is not internal to the router/switch architecture, and not dependent on a particular network interface component. The board provides a

general pipelined processing resource, which can be used for network processing, or can also support other pipelined I/O processing, *e.g.*, for 'data ingest' processing prior to disk storage [4].

3.1: Internal architecture

Internally, the BRAIN board is a pipelined processor (Figure 4). It uses separate PCI interfaces for data input and output, and relies on a multi-Harvard central processor, currently indicated[1] as the Texas Instruments TMSC40 DSP [18]. The processor supports two 32-bit full-rate DMA channels, as well as multiple low-speed asynchronous DMA channels which are used here to access an on-board PLD development area and a local PCI bus for disk access, etc. The DSP also contains multiple integer instruction units, which allows parallel processing of the data stream.

On-board, dual-ported RAM is used on both the input and output ports, to decouple the I/O DMA from the bus interface. A separate on-board memory supports scratchpad workspace, program storage, and additional slots that can be used for 'value-added' memory, such as CAMs. The PLD socket supports a high-level programmable device, *e.g.*, an AMD Mach 465 [1]. Earlier versions of this chip were used to implement single-clock IP checksum operations; this version will allow complex logic functions to be implemented in hardware, rather than in an often cumbersome sequence of software operations [20].

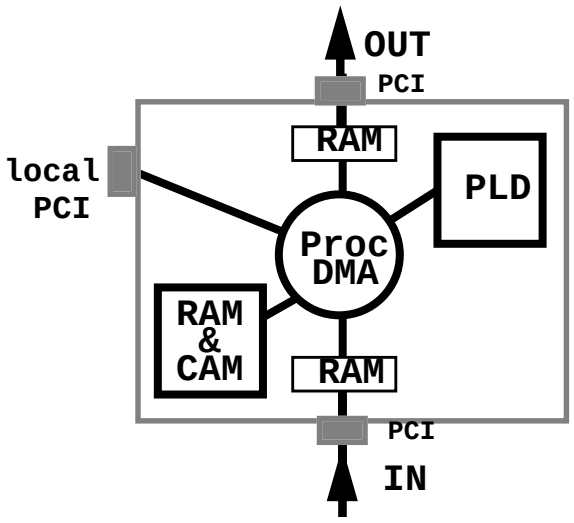

FIGURE 4. Internal design

1. Available dual-Harvard CPUs supporting dual DMA will be compared and selected during the final design.

3.2: Interface architecture

The BRAIN board uses an existing PCI-based host as a controller. There are two possible configurations, which will be compared as part of the proposed work. In the first, the input and output PCI interfaces of the BRAIN board plug directly into a dual-PCI host (Figure 5). In a dual-PCI host, a bridge chip isolates the two PCIs [8]. Network input passes over PCI #1 into the BRAIN board, and out onto PCI #2 to the network output. Because the address spaces of the input and output side of the BRAIN board are distinct, traffic never crosses the bridge chip. In a sense, the BRAIN is performing as a intelligent PCI bridge, in which data is processed as well as being transferred across the busses. In this configuration, the BRAIN's local PCI is an on-board slot, the card edge plugs into a PCI #1slot, and has a tethered daughtercard edge that plugs into a PCI #2 slot. The signal properties of PCI support this configuration inside a single host easily, which is why the newly-available dual-PCI configuration enables this design [15].

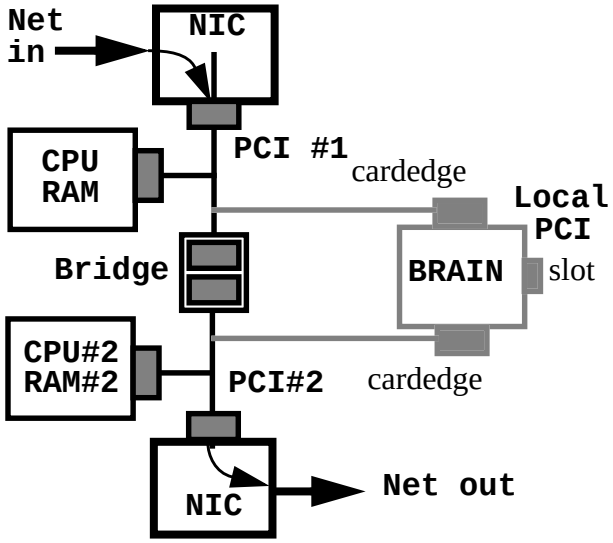

FIGURE 5. BRAIN uses dual PCI, existing NICs

In the second possible configuration, the BRAIN board is controlled via a single-PCI host, and uses two on-board slots, one for each NIC, and the local PCI is connected to the card edge to the host PCI (Figure 6). The design does not rely on the more complicated dual-PCI host architecture, but also requires more complicated power and clock engineering to support multiple card slots on-board. The only difference between these configurations is the use of card edge connectors vs. slots, and the final design will be based on engineering evaluation.

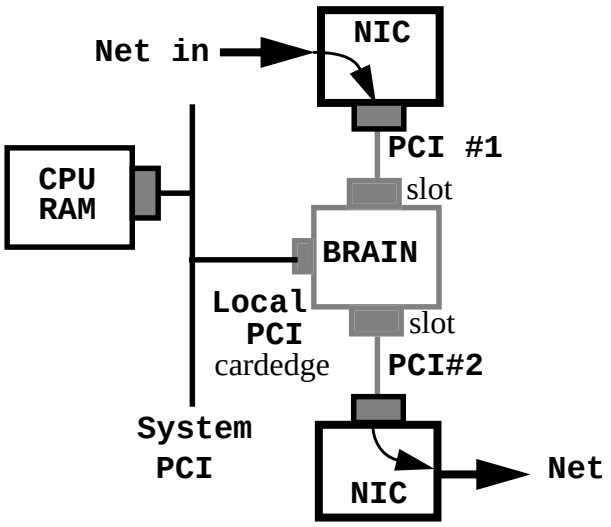

FIGURE 6. BRAIN uses single PCI, existing NICs

3.3: Design benefits

The BRAIN board uses an existing multi-Harvard architecture processor to provide high-performance processing resources while maintaining simultaneous high-performance input and output bandwidth. To the network, the BRAIN board is similar to the conventional link-based host processor (Figure 2), but avoids the bandwidth limitations of the latter. Our design also uses existing NICs, supporting any switch or link technology. It also decouples the design of the BRAIN board from the design of the router or switch internals, unlike internal processing solutions (Figure 1).

The BRAIN board replaces the assumption of a particular link or network interface with that of a standard bus interface, 32-bit PCI. It is therefore also useful as an intelligent bridge between differing link technologies, or for the pipeline processing of any PCI stream. In the latter case, the BRAIN board can be used for data ingest between a network interface and permanent storage, or rendering graphics prior to display.

4: Prior work

There is only one other known available dual-PCI controller board, the Cyclone Microsystems PCI-914 Intelligent I/O Controller [5]. The IIOC contains an Intel *i*960 processor which uses a single-Harvard architecture, and so does not support simultaneous input and output. Its primary PCI is a cardedge, but its secondary PCI supports only "IQ Modules," an open but non-standard PCI interface. As a result, the Cyclone board is not suited to general use with generic existing PCI NIC cards.

There have been many notable prior outboard network processor boards [6], [7], [10], [12]. In most cases, these boards relieve the host CPU of handling interrupts, scheduling data transfers, or packet processing, *i.e.*, they reduce the processing load on the CPU. The BRAIN processor addresses the bandwidth limitation of a central, single-Harvard architecture processor, providing a pipelined processing resource mapped directly to the NICs.

5: Summary

Active Networking and high-performance in-band packet processing requires high-bandwidth programmable computational resources. The BRAIN board would provide this much needed resource, in a link- and switch-independent fashion. The board is also useful for high-speed pipelined data transformations supporting data fusion, satellite image ingest processing.

Both the overall system architecture and the internal board architecture of BRAIN are simplified by the emergence of multiple-PCI PC hosts. BRAIN also provides both a pipelined CPU as well as a prototyping area for custom hardware.

We envision an effort to produce and develop the BRAIN board, and to distribute and coordinate its utilization by the networking research community. The boards would be provided to qualified researchers in a manner similar to (or using the same mechanism as) the GNT Distribution Program. This effort includes the adaptation of TMS320C40 development and debugging tools to the BRAIN board, to provide a suitable programming environment for network researchers.

Extended research using BRAIN would involve the design of a CAM module, and development of transcoding and encryption algorithms that utilize the on-board PLD. The board can also be used for direct-to-disk packet logging.

6: References

[1] Advanced Micro Devices AMD Mach 465 part information, <http://www.vantis.com/products/overview/c17466d.html>

[2] BBN Multigigabit Router, information <http://www.bbn.com/>

[3] Bhattacharjee, B., Calvert, K., Zegura, E., "An Architecture for Active Networking," Technical Report CIT-CC-96-20, College of Computing, Georgia Institute of Technology, 1996.

[4] Cromp, R., Campbell, W., and Short, Jr., N. ("An intelligent information fusion system for handling the archiving and querying of terabyte-sized spatial databases," International Space Year Conference on Earth and Space Science Information Systems, Pasadena, CA, Feb. 1992.

[5] Cyclone Microsystems PCI-914 Intelligent I/O Controller board information, <http://www.cyclone.com/pci914.htm>

[6] Dalton, C., Watson, G., *et al*., "Afterburner," IEEE Network, V7 N4, July 1993, pp. 36-43.

[7] Davie, B., "The Architecture and Implementation of a High-Speed Host Interface," IEEE JSAC, V11 N2, Feb. 1993, pp. 228-239.

[8] Dual-processor motherboard specification (Intel), see <http://www.intel.com/design/pro/datashts/242016.htm>

[9] Gigabit Network Technology (GNT) Distribution Program, information <http://www.arl.wustl.edu/~jst/gigatech/ResDist.html>

[10] Kalmanek, C., *et al*., "Xunet 2: Lessons from an Early Wide-Area ATM Testbed," IEEE/ACM Transactions on Networking, V5 N2, Feb. 1997, pp40-55.

[11] Minshall, G., Lyon, T., and Huston, L., "IP Switching and Gigabit Routers," IEEE Communications Magazine, Jan. 1997.

[12] Mockapetris, P., "Communication Environments for Local Networks," Ph.D. Dissertation, UC Irvine, Dec. 1992.

[13] Myricom home pages, <http://www.myri.com/>

[14] Parulka, G. "ATM Tap Project," Wash U. in St. Louis, pending project.

[15] PCI Local Bus Specification, Rev. 2.0, PCI Special Interest Group, Hillsboro, OR, 1993.

[16] Rekhter, Y., *et al*., "Tag Switching Architecture - Overview," (working draft), Jan. 1997.

[17] Tennenhouse, D.L., *et al*., "A Survey of Active Network Research," IEEE Communications Magazine, Jan 1997, pp. 80-86.

[18] Texas Instruments TI TMS320C40 part information, <http://WWW.TI.COM/sc/docs/dsps/products/c4x/>

[19] Touch, J., *et al*., "ISI's High-Performance Networking Research," submission to the GNT Workshop, St. Louis MO, June 1996 <http://emperor.arl.wustl.edu/~jst/gigatech/whitepapers/touch.ps>

[20] Touch J., Parham, B., "Implementing the Internet Checksum in Hardware," RFC-1936, ISI, April, 1996. <ftp://ftp.isi.edu/in-notes/rfc1936.txt>