Rattlesnake: a Network for real-time Multimedia Communications

Gerard J.M. Smit, Paul J.M. Havinga, Michèl J.P. Smit
University of Twente, dept. Computer Science
P.O. Box 217
7500 AE Enschede, the Netherlands
e-mail: smit@cs.utwente.nl

Abstract

In this paper we describe the design of a local area network suitable for distributed multimedia communications.
Multimedia applications require a communication infrastructure with capabilities beyond the current state of the art: real-time stream traffic, small end-to-end latency with little variation, high bandwidth and high availability. In most state-of-the-art LANs the bandwidth is limited to the link bandwidth. We claim a high aggregate bandwidth with a moderate link bandwidth. Moreover, the performance of many LANs degrades rapidly under heavy load conditions. In our network we use real-time virtual channels to guarantee a bounded latency.
The Rattlesnake network has a star shaped topology. All stations are connected by dedicated links to a central switch box. Inside the switch box we use a network with a Kautz topology, which has desirable features such as: self routing capability, small diameter, fixed degree, fault tolerance and suitable for VLSI implementation.
The proposed network supports ATM based communication. Due to our hybrid time division multiplexing approach, stringent real-time requirements can be met.

Keywords: multimedia, low latency communication switches, real-time, LAN, Kautz topology, hybrid TDM.

1. Introduction

This paper presents the results of recent research in multimedia communication at the Twente University. The research is part of the Huygens project, that started in 1991 [Mullender 91]. This project addresses multimedia systems and applications and embodies all levels between Operating System support and the hardware. Multimedia applications use multiple communication types to combine for instance moving images, sound, animated graphics, photos, graphs and typeset text in a document. Furthermore, current workstations integrate the functions of a television screen and camera, telephone and answering machine, word processor, fax machine, as well as a recording device for all of these. Because workstations are getting faster and more numerous, there is an increasing need for faster, higher-capacity LANs. One or a few modern high performance workstations can use the entire data transfer capacity of the current networks.
Integration of different services on the same network requires LANs to transmit not only data but also voice and video. Each of these services has their own Quality of Service requirements. To meet these diverse demands, it is essential that LANs, will be capable of operating at high data rates and achieve stringent delay requirements.

The bandwidth of the existing networks is by far not enough for distributed multimedia applications. Ethernet, with 10 Mbit/s link bandwidth and FDDI or the Cambridge fast ring [Hopper 88], with a link bandwidth of 100 Mbit/s, are LANs for high-performance workstations. Their throughput and latency are becoming a bottleneck in demanding real-time applications. Furthermore, overall performance strongly depends on the number of transmitting stations [Dykeman 88].
The fundamental advantage of our network is that we claim a greater aggregate bandwidth with a moderate link bandwidth. Where in networks with a ring or bus topology such as: FDDI, Ethernet and the Cambridge fast ring, the aggregate network bandwidth is limited to the link bandwidth, in our project the aggregate bandwidth will be many times the link bandwidth and grows with the number of workstations attached to the network.

A similar approach is found in the Autonet project [Schroeder 90]. This project reports similar advantages. Other networks with a high aggregate bandwidth are: Hubnet [Lee 83], Tree-Net [Gerla 1988], Manhat-

tan Street Network [Maxemchuk 85]. An overview is given in [Abeysundara 91]. A drawback of most of these systems is that a bounded packet delay cannot be guaranteed. In our project, however, the *real-time behaviour* is an essential design issue. We can guarantee a bounded latency, therefore our network can be used by distributed real-time applications such as distributed process control and distributed multimedia applications.

To match the transmission speed of the network links, and to minimize the overhead due to processing of network protocols the switching of messages is done in hardware.
In our proposal we use techniques such as: virtual channels, worm-hole routing and deadlock avoidance, that are well known in the multi-processor field. We have applied these techniques in the Rattlesnake switch. *Virtual networks*, implementing a number of virtual channels on one physical link, were first introduced as a technique to avoid deadlocks in networks. Dally [Dally 90] showed that virtual networks increase the connectivity of networks and have performance advantages. In the iWARP [Borkar 90] processor virtual links implement real-time channels. In the iWARP processor some virtual channels can be pre-defined and fixed during a communication session. So there is a guaranteed bandwidth between nodes, that can be used for real-time communications.
Wormhole routing operates by advancing the head of packets directly from incoming to outgoing channels. Only a few control digits (called flits) are buffered at each node.

2. Kautz networks

We use Kautz networks in our project because these networks have interesting properties [Bermond 89]. Particularly, they interconnect considerably more nodes than the usual topologies, they have a small diameter, and a small and fixed degree.

Definition of Kautz graphs [Kautz 68]

The Kautz digraph K(d,k) with in-degree and out-degree d and diameter k is the digraph whose vertices are labelled with words $(x_1,...,x_k)$ of length k from an alphabet of d+1 letters by removing those words in which there are two consecutive identical letters ($x_i \neq x_{i+1}$, for $1 \leq i \leq k-1$). There is an arc from a vertex x to a vertex y if and only if the last k-1 letters of x are the same as the first k-1 letters of y.
A straightforward generic route of length k can be found by simple concatenation of source and destination word. However, in general there are routes with

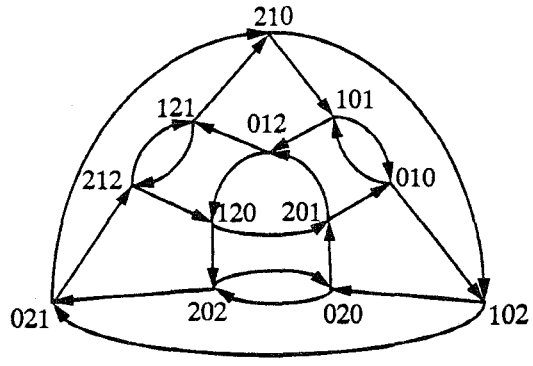


Fig. 1: Example of a Kautz graph (K(2,3)).

length < k [Smit 91a].

Example 1 (see figure 1)
In the graph we find the route Rg = < 120201 > from (120) to (201) via node (202) and (020). This route has length 3 (= k). However, there is a shorter route: Rs = < 1201 > of length 1.
An algorithm for generating all node disjoint routes is straightforward [Smit 91a].

Properties of Kautz digraphs

Some properties of Kautz digraphs are:

- The number of vertices $N = d^k + d^{k-1}$. This implies that Kautz graphs K(d,k) interconnect considerably more processors than the other topologies[1] with diameter k and d=k.
  Table 1 compares Kautz digraphs with 'de Bruijn' digraphs [deBruijn 46] and the binary hypercube. The de Bruijn digraph is selected because its definition is related to Kautz digraphs.

	d=k=4	d=k=6	d=k=8	vertices
hypercube	16	64	256	$N = 2^k$
de Bruijn	16	729	65536	$N = d^k$
Kautz	24	972	81920	$N = d^k + d^{k-1}$

Table 1: Comparison with other graphs.

•The degree of a Kautz graph is *fixed* and independent of N. Networks of arbitrarily large size can be built using (VLSI) components as nodes with a *fixed* number of connections per node. Where Kautz networks have a

1. For the de Bruijn and Kautz digraphs the mentioned degree is the sum of the out-degree and in-degree. Thus a Kautz digraph with in-degree and out-degree of 4 and a diameter of 8 connects 81920 nodes, which is significantly more than the 256 nodes in a hypercube.

fixed degree, other networks, suchas the binary hypercube, require the number of connections per node to *increase* with (the logarithm of) the number of nodes.

- The *diameter* of the network is k ($< {}^{d}\log N$).
- A Kautz network is *fault tolerant*. The connectivity of K(d,k) equals d. The diameter of the network in case of faulty nodes has also been studied by Imase et al. [Imase 86]. They showed the existence of d vertex disjoint paths between any pair of vertices in K(d,k), one of a length of at most k, d-3 of a length of at most k+1 and two of a length of at most k+2. This implies that the performance degradation due to increased routing distances resulting from faults is fairly low.
- Another interesting property of the network is the fact that it admits *self routing* of messages, both if the network is fault free as well as when some nodes or links are faulty.
- A Kautz graph can *emulate standard computation graphs* such as a linear array, ring, mesh and tree.

3. Transport services and transfer modes

Traditional transport protocols have mainly serviced the communication needs of file transfer and low-bandwidth interactive usage. Advances in computer technology and changes in network architecture are rapidly moving us to an environment where communication will be increasingly oriented towards multimedia and real-time services [Biersack 90].
We expect the following two major services to be supported by general purpose LANs:

- *Real-time low-latency services*
  Low-latency services are necessary for voice and video transfer, process control, remote sensing, etc. The data for this service is usually worthless if it does not arrive in time. Furthermore, applications may require reliable connections.
- *Non real-time transactions and bulk data transfer*
  Transactions occur in distributed operating systems [Mullender 90]. Examples include database queries and remote procedure calls (RPC). Transactions require low latency, and have a small or moderate amount of data to be transmitted. Bulk data transfer is a service that carries a large amount of data with relatively loose latency constraints.

The underlying transfer mode must support the above mentioned services. Three known techniques will be presented, and their (dis-)advantages will be shown using figure 2.

- *Synchronous Transfer Mode (STM)*
  The STM or circuit switching mode makes use of frames. Time is divided into equal parts, in which a frame is transmitted. A frame is divided into slots. Each source is allowed to use one or more slots of a frame. STM has the disadvantage that the bandwidth is divided into fixed slots and frames. A major advantage of STM is that it guarantees bandwidth with a fixed, bounded delay. Figure 2 shows that a lot of bandwidth is wasted using STM. A channel of 40 Mbit/s should be reserved while the average usage of the channel is 15 Mbit/s.

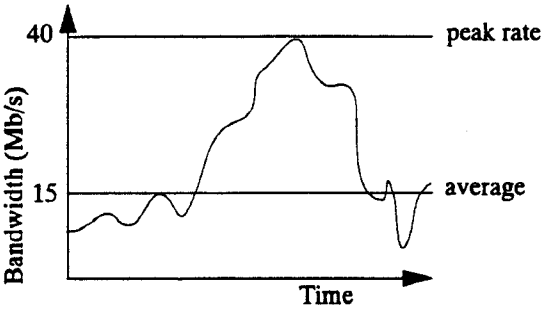


Fig. 2: Variable bit rate video coding of a television channel [Wright 90]

- *Asynchronous Transfer Mode (ATM)*
  The ATM, or packet switch mode, approach abandons the concept of frame references. ATM achieves a more flexible bandwidth sharing by allowing the terminals to seize bandwidth when a packet is ready for transmission. Each packet has a label that identifies the path of the packet through the network. The label can be a virtual channel number or in case of a self-routing network, a destination address. The main disadvantage of ATM is the possible collision of packets trying to seize the same slot. This requires scheduling and buffering mechanisms. Buffering of packets may result in a long delay, and in case of buffer overflow in discarding of packets. The main advantage of ATM compared to STM is the better performance in case of services with variable bit-rates. Furthermore, ATM is not technology dependent. Figure 2 shows the advantage of ATM as a variable bit rate transporter. Bandwidth will only be used when data is available. However, only a percentage of the peak bandwidth will be claimed, say 15 Mbit/s. This will result in longer latency's during peak hours and might cause an unacceptable latency for some applications. Furthermore a percentage of packets will be lost due to bufferoverflow.

- *Hybrid Time Division Multiplexing (Hybrid TDM)* Hybrid TDM is a combination of STM and ATM switching techniques [Hui 90]. It combines the flexibility of ATM with the capability of assigning time slots of STM, see figure 3. Each frame has a fixed number of slots. Part of these slots are assigned to hard real-time services (STM with guaranteed bandwidth), and the rest to non hard real-time services (ATM packets). If a real-time slot is not needed by its source, it can be used by non real-time services. The size of a slot is independent of the ATM packet size. This means that one ATM packet is fragmented into several slots. Figure 2 shows the real advantage of a Hybrid TDM scheme. The real-time video service will claim a STM channel of 40 Mbit/s, and have a guaranteed latency. Whenever the video services is not using its assigned peak bandwidth, ATM packets can seize the available claimed but not used bandwidth.

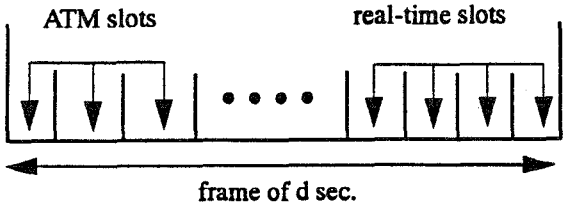


Fig. 3: Hybrid Time Division Multiplexing [Hui 90]

The CCITT has recommended ATM as the target transfer mode for broadband ISDN networks. It is expected that most devices and applications will generate ATM packets in the future. Our network can be used as a platform for ATM communication, and will provide extra facilities for demanding real-time applications. The transfer mode we will use is a kind of Hybrid TDM.

4. Global architecture

4.1 Overview

A distributed multimedia system should support a wide range of communication types and primitives. In these systems workstations are connected to a various number of services such as: high-performance file servers, communication servers (gateways to Wide-Area-Networks), servers for manipulation of voice video and animation, fax machines, telephone and answering machines, interactive video (CD-i), etc (see fig. 4). Our network typically provides communication facilities within a building or campus. In each building or floor there are one or more switch boxes (often called *hubs*). Each switch box can connect about 100 workstations located in a distance of up to 250 meters from the switch box. High speed links interconnect the switch boxes. In this paper we mainly discuss the communication within a switch box.

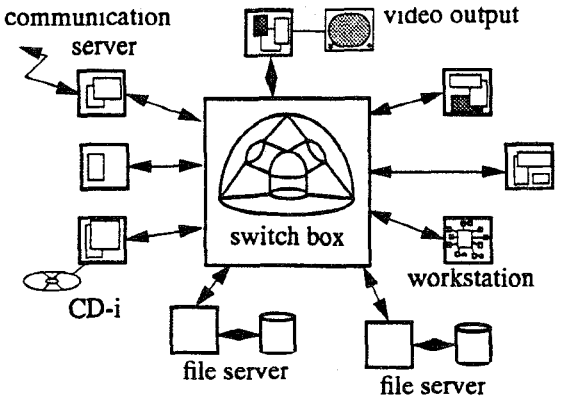


Fig. 4: Global architecture of a distributed multimedia system.

The communication architecture should provide:

- Support for high performance *client/server* interaction such as remote-procedure calls.
- The ability to establish *hard real-time* connections (i.e. bandwidth reservation using real-time virtual channels).
- Support for *stream traffic* such as voice and video. Stream traffic is characterised by a continuous bit stream flowing from one user to the other(s), mostly for a relative long period of time. They impose strong requirements on the network in terms of delay, bandwidth, real-time behaviour, latency and transmission quality. There is no time for retransmission. A video sequence, for instance, must be retrieved at a high and constant rate; frames retrieved too late are no longer useful. A hypertext document, in contrast, is retrieved in the order in which the user happens to browse through it, and need not be retrieved in real-time.

We have chosen a *star* shaped network, in which all stations are connected by dedicated links to a central switch box. A connection between two stations is established through the switch box.
This approach has a number of advantages:

- Within the switch box a number of connections can be handled simultaneously.
- Because of the point-to-point connection from the station to the central switch box the achievable throughput is high.
- A point-to-point connection is suitable for an optical fiber based implementation.
- State-of-the-art components such as TAXI chips can be used the point-to-point links.
- The interfaces in the workstations can be simple

and low cost (most of the complexities is within the switch box) and the design is relatively independent of the technology of the physical layer of the links.

A direct consequence of this configuration is that the *performance* and the *availability* of the switch box are critical design factors that must be studied carefully. Communication should be possible even if some of the links or workstations fail. Therefore separate links will connect workstations to two or more switch boxes. If one of the links (or the switch boxes) fail an other link (or an other switch box) can be used.

4.2 Architecture of the switch box

A switch box consists of a number of switching elements interconnected via bi-directional links. The distances within the switch box are small, so the standard techniques can be used. Because the switches are physically relatively close together (typically in one cabinet) the links between the switches need not be serial. For performance and cost reasons parallel connections will be used.

Fig. 6 shows the internal structure of the data-path of a switching element. A typical switching element has 3 input and 3 output links. With this element several physical network topologies with in-degree and out-degree 3 can be built, such as: torus, mesh, deBruijn networks and Kautz networks.
Although the switching elements can be configured in several network topologies, we advocate a Kautz network topology [Smit 91a] because of its valuable properties. Particularly, Kautz graphs have a small diameter, a fixed and a small degree.

A connection between two arbitrary stations is made via two or more switching elements in the switch box. A message generated by a source station travels through these switching elements to reach a destination station. Switching elements [Smit 91b] contain logic to forward messages from an input link to an output link, as directed by the destination address in each message header. The input and output links can be interconnected in several ways. Feasible implementations are a cross-bar, a bus and a slotted ring.

Switching elements forward messages using a wormhole routing [Dally 87] technique that minimizes switching latency. We use the routing capability of Kautz networks. This means that given the Kautz addresses of source and destination all link (and node) disjoint path can be generated easily. There is no need for routing tables or algorithms to generate routing tables. Furthermore there is no need for reconfiguration whenever links fail.

4.3 Interface with workstations and servers

Each switching element is connected via point-to-point serial links (> 100 Mbit/s) to a (work)station or a server. Figure 5 shows this interface. We plan to use standard serial link drivers like the AMD TAXI chip set [TAXI 87], leaving the problems of phase-locked loops and data encoding to others.

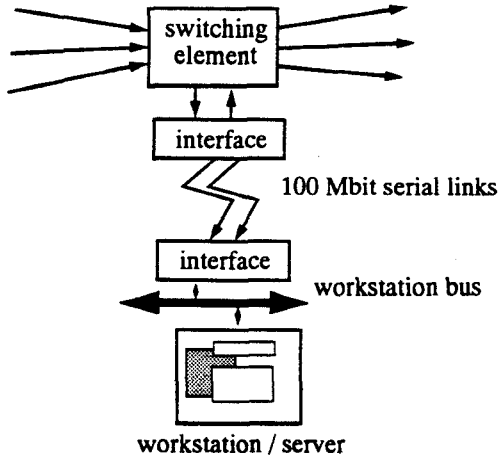


Fig. 5: Interface with workstations.

In the first design we plan to use uni-directional coaxial links. The coaxial technology limits the distance of workstations and switches to 250 m. Fiber optic links might be used between buildings because of their longer length limit. Every workstation has an interface between the serial link and an (internal) workstation bus. As there is no general standard bus for workstations and because we want to support a wide range of workstations types, a number of different interfaces will be designed. Interfaces will be designed for: EISA bus, VMEbus and Future bus. For most buses standard glue components are available so the design will be reasonable straightforward.

5. The Switching Element

A Rattlesnake switching element will support a Hybrid TDM scheme, see section 3 and figure 3. The switching element has two main functions. Firstly, it supports *hard real-time traffic* in a circuit switching fashion (STM). Secondly, it supports the *lower priority traffic* in a packet store-and-forward fashion (ATM). Figure 6 depicts the general architecture of the Rattlesnake switchThe switching elements are connected via bi-directional links. They communicate with each other via these links by exchanging Rattlesnake frames

A switching element contains k buffers at each input and output link [Smit 91b]. The buffers are connected with each other via an internal bus, ring or a crossbar.

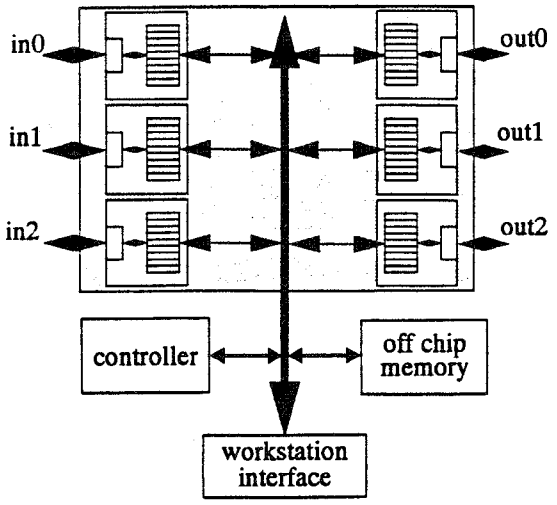


Fig. 5: Structure of the Rattlesnake Switching Element.

Switching elements contain logic to forward messages from an input buffer to an output buffer. When establishing a connection from source to destination of a hard real-time virtual circuit, each switch assigns an input buffer and an output buffer for this circuit.

The number and size of the buffers is limited by the implementation of a switching element. The current prototype, implemented in a Field Programmable Gate Array (FPGA), has 3 input and 3 output links (see section 6). Each link has 16 buffers of 16 bits.

5.1 The Rattlesnake frame

The Rattlesnake frame slightly differs from the Hybrid TDM frame (see figure 3). A Rattlesnake frame consists of a head, a data-part and a tail, see figure 7. The data-part can be seen as a variable length Hybrid TDM frame, and has a maximum of k slots. While a Hybrid TDM frame has a fixed number of slots, a Rattlesnake

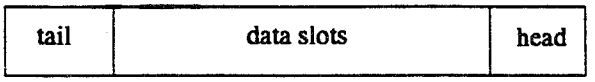


Fig. 7: The frame format of a Rattlesnake switching element.

frame has a variable number of slots, i.e. it will become shorter when there is no pending data for that output link. This will result in higher bandwidth availability in low-traffic situations.

Real-time virtual channels have priority access to their reserved slot. Reserved slots that are not actually used and free slots are used for non real-time ATM slots.

The *header* of the frame consists of k bits indicating which real-time channels use a slot in this frame. A bit set in the header at position d, indicates that virtual channel number d occupies a slot in the frame.
The *tail* is used for flow-control and has a length of k bits as well. It indicates end-of-frame. The bits set indicate which real-time channels in the opposite direction of the link can receive a new data item.

Hard real-time communication
Each link supports a maximum of k hard real-time virtual channels. For each of those channels an input and an output buffer at either side of the link exists. An end-to-end connection is established by reserving subsequent virtual channels (= buffers) of the route from source to destination. In each node a translation table from input buffer to output buffer exists.

Non real-time communication
Non hard real-time packets are segmented, such that each segment fits into a slot of the Rattlesnake frame. Several frames may be necessary to forward a complete packet. Packet segments are moved from the input buffer at the link and stored in an input buffer at an off-chip memory, until the complete packet has arrived. ATM packet header information is used to route the packet. Priority queues can be implemented in this memory. Because packets are segmented, variable length packets can also be handled.

5.2 Rattlesnake frame protocol

The data transfer process between two nodes, is a constant exchange of frames up and down a link. The next frame is sent when the tail of the frame from the opposite node has arrived.
At the start of a new frame cycle, a switching element will collect k bits from the real-time buffers, indicating whether the buffer has output data. These k bits and the last received tail are used to compose the new header. The new header indicates which real-time channels are allowed to sent data in this frame.
First the new header is transferred, than the data slots are transferred and finally a tail is generated and transmitted.
This frame is parsed when it arrives at the input link of the next node. The head indicates how many and which slots are to be stored in the input buffers. The tail is used to construct the head of the new frame in the opposite direction.

6. Current status

An initial implementation of the Rattlesnake switching element, having one hard real-time virtual channel and

6 uni-directional links, has been realized with a XC3042 Field Programmable Gate Array (FPGA) of Xilinx [Xilinx 91]. The implementation, described in VHDL, has been simulated and subsequently a gate array has been synthesized by the VHDL synthesizer from VIEWlogic. It used 126 of the 144 available CLB[1]s [Smit 91c] and ran at a clock speed of 10 MHz. Currently we are implementing a Rattlesnake switching element with 16 real-time virtual channels and 6 bi-directional links in a XC4005 FPGA.

The presented architecture has been described in VHDL. The architecture phase is completed, and we are now implementing the design. A prototype is expected this summer.

7. Conclusion

In this paper we have presented a proposal for a high-performance, low-latency network suitable for hard real-time multimedia applications. Our Rattlesnake switch uses real-time virtual channels to guarantee a bounded latency. It can be implemented with standard components such as FPGAs.

The communication network of the switch is based on a Kautz topology. Kautz graphs form a class of interconnection networks with interesting properties such as: small diameter, large number of nodes ($N = d^k + d^{k-1}$), the degree is independent of the network size, the network is fault-tolerant, it can embed standard computation graphs and has a simple routing algorithm.

The fundamental advantage of our network is that we claim a high aggregate bandwidth with a moderate link bandwidth. In networks with a ring or bus topology such as: FDDI, Ethernet and the Cambridge fast ring, the aggregate network bandwidth is limited to the link bandwidth. In our project the aggregate bandwidth will be many times the link bandwidth and will grow with the number of workstations attached to the network.

Our network can be used as a platform for ATM communication, and will provide extra facilities for demanding real-time applications. We use a kind of Hybrid TDM transfer mode.

We have chosen a *star* shaped network, in which all stations are connected by dedicated links to a central switch box. This approach has a number of advantages: high aggregate bandwidth, low cost point-to-point connections, suitable for an optical fiber based implementation, and a simple interface to workstations.

One of the areas that needs more research attention is the performance evaluation in integrated service environments. In such environments, different types of communications may coexist in a switch and heavy concentration of traffic may occur.

1. A CLB (Configurable Logic Block) is the basic building block of a Xilinx FPGA. It contains programmable combinatorial logic and two storage registers.

References

[Abeysundara 91] Abeysundara B. W., Kamal A.E.: "High-Speed Local Area Networks and Their Performance", ACM Comp. Surveys, June 1991, pp 221-264.

[Bermond 89] Bermond J.C., Homobono N., Peyrat C.: "Large Fault-Tolerant Interconnection Networks", Graphs and Combinatorics, 1989.

[Biersack 90] Biersack E.W., Feldmeier D.C.: "Transport Protocol Issues for ATM-based Networks", proc. EFOC/LAN 90, June 1990, pp 104-113.

[Borkar 90] Borkar S. et al.: "Supporting Systolic and Memory Communication in iWarp", Proc. 17th ACM/IEEE Symposium on Computer Architecture, 1990, pp 70-81.

[deBruijn 46] de Bruijn N.G.: "A combinatorial problem"; Koninklijke Nederlandse Academie van Wetenschappen Proc. A49, pp 758-764; 1946.

[Dally 87] Dally W.J.: "A VLSI Architecture for Concurrent Data Structures", Ph.D. thesis, Computer Science, California Institute of Technology, 1987.

[Dally 90]] Dally W.J.: "Virtual-channel Flow Control", Proc. 17th ACM/IEEE Symposium on Computer Architecture, 1990, pp 60-67.

[Dykeman 88] Dykeman D., Bux W.: "Analysis and tuning of the FDDI media access control protocol", IEEE J. Selected Areas Commun. 6, July 1988, pp 997-1010.

[Gerla 88] Gerla M., Fratta L.: "Tree structured fiber optics MANs", IEEE J. Selected Areas Commun. 6, July 1988, pp 934-942.

[Hopper 88] Hopper A., Needham R.M.: "The Cambridge fast ring networking system", IEEE Trans. Comput. 37, Oct. 1988, pp 1214-1223.

[Hui 90] Hui J.Y.: "Switching and traffic theory for integrated broadband networks.", Dordrecht, The Netherlands: Kluwer Academic Publishers, 1990.

[Kautz 68] Kautz W.H.: "Bounds on directed (d,k) graphs. Theory of cellular logic networks and machines", AFCRL-68-0668 Final report, pp 20-28, 1968.

[Imase 86] Imase M., Soneoka T., Okada K.: "A fault-tolerant processor interconnection network" (original in Japanese); translated in Systems and Computers in Japan, vol. 17, no 8 pp 21-30, 1986.

[Lee 83] Lee E.S., Boulton P.I.P. et al.,: "The principles and performance of HUBNET: A 50 Mbit/s glass fibre local area network.", IEEE J. Selected Areas Commun. 6, July 1983.

[Maxemchuk 85] Maxemchuk N.F.: "The Manhattan street network", Proc. of the IEEE GLOBECOM Conf., 1985, pp 255-261.

[Mullender 90] Mullender S.J. et al.: " Amoeba - a Distributed Operating System for the 1990s", IEEE Computer 23, May 1990.

[Mullender 91] Mullender S.J.: "The Huygens Project", internal memo Twente University dept. Computer Science, 1991.

[Schroeder 90] Schroeder M.D., Birrell A.D. et al.: "Autonet: a High-speed, Self-configuring Local Area Network Using Point-to-point Links", Digital Systems Research Center, Palo Alto, CA, April 1990.

[Smit 91a] Smit G.J.M., Havinga P.J.M., Jansen P.G.,: "An algorithm for generating node disjoint routes in Kautz digraphs", Proceedings Fifth International Parallel Processing Symposium, Anaheim, CA, 1991.

[Smit 91b] Smit G.J.M., Havinga P.J.M., Jansen P.G.,: "On the design of a reconfigurable network switch", Proceedings Euromicro 91, Vienna, 1991.

[Smit 91c] Smit G.J.M., Havinga P.J.M., Jansen P.G.,: "A programmable network switch for Kautz networks", Proceedings Parallel Computing 91, London, 1991.

[TAXI 87] Advanced Micro Devices. TAXIchip integrated circuits. AMD7968/ AMD 7969. Publication 07370, Sunnyvale, CA, May 1987.

[Wright 90]Wright D.J., To M.,:"Telecommunication applications of the 1990s and their transport requirements", IEEE network magazine, March 1990, pp 34-40.

[Xilinx 91] "The Programmable Gate Array Data Book", Xilinx Inc., 1991.

Rattlesnake

a network for real-time multimedia communications

Gerard J.M. Smit, Paul J.M. Havinga, Michèl J.P. Smit

University of Twente

Contents

Introduction

Real-time local area communication network

- Purpose: Real-time communications for:
  - multi-computer systems
  - distributed process control
  - multimedia systems
- Network has Kautz topology
- Hybrid TDM switching
- Network offers platform for ATM switching

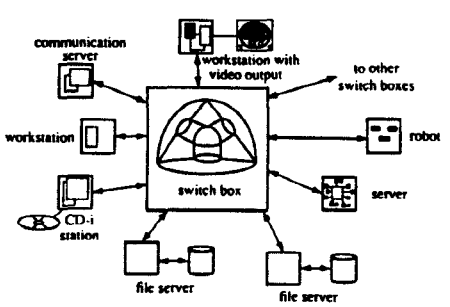
Research at the Twente University SPA group

- ***Pegasus***: Operating system support for distributed multimedia systems
  - Cambridge
  - Twente
- ***Broadcast***: Distributed systems support for very large systems
  - Newcastle
  - INRIA
  - Bologna / Pisa
  - INESC
  - Twente
  - Laas
  - Grenoble
- ***Huygens***: umbrella project, real-time communication architectures
  - partially sponsored by HP, Esprit, Xerox Europe

Global architecture

Local Area Network

- ± 100 workstations or servers per switch box
- high aggregate bandwidth with a moderate link bandwidth
- support for RPC's, hard real-time traffic, stream traffic



- switch box may be interconnected to form a Campus Area Network (± 2000 nodes)

Star shaped network

+ complexity encapsulated in the switch box

  small distances ⇒ parallel connection

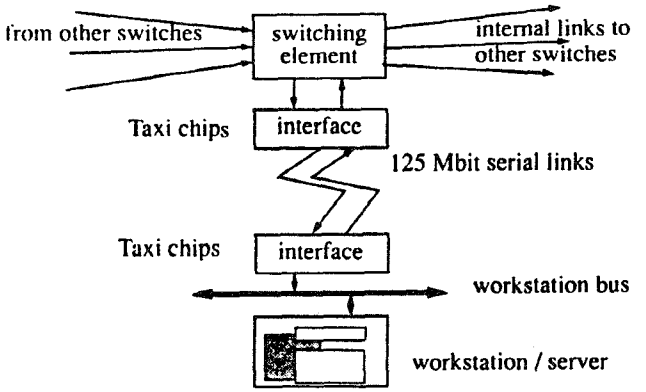
+ interface with workstations / servers simple and cheap

- availability of the switch box ⇒ alternative links to other switch boxes

Global architecture (2)

Workstation / server ↔ switch box interface

- serial 125 Mbit/s links (coax, fiber, etc.)
- standard link drivers (Taxi chips)
- interface technology independent of switch box technology

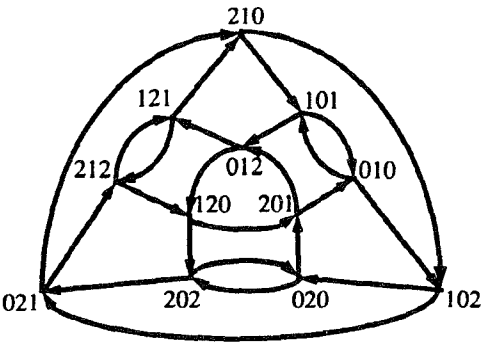


Kautz graphs

Definition

The Kautz digraph K(d,k) is the digraph whose vertices are lab with words $(x_1,...,x_k)$ **of length k** from an alphabet of d+1 lette removing those words in which there are two consecutive ider letters.

There is an arc from a vertex x to a vertex y if and only if the k-1 letters of x are the same as the first k-1 letters of y.



K(2,3)

Alphabet = { 0, 1, 2 }.

Properties of Kautz graphs K(d,k)

- the graph is *degree-regular*
- the degree is *fixed*
- the *diameter* of the network (k) is small ($< {}^{d}\log N$)
- the number of vertices $N = d^k + d^{k-1}$

	d=k=4	d=k=6	d=k=8	number of nodes
k-cube	16	64	256	$n = 2^k$
de Bruijn	16	729	65536	$n = d^k$
Kautz	24	972	81920	$n = d^k+d^{k-1}$

Comparison of the number of connected nodes under given degree and diameter

- a Kautz network is *fault tolerant*
- the *connectivity* of K(d,k) equals d. There are d **vertex disjoint** paths between any pair of vertices in K(d,k), with length of at most k+2.
- a Kautz graph can *emulate standard computation graphs*, such as ring, tree, torus

Routing in a Kautz network

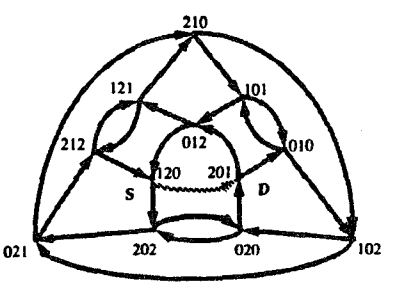
- the network admits *self routing* of messages. This me: that a connection-less services is easy to implement
- algorithm for generating node disjoint routes is straightforward

Example A: simple concatenation

A route from node **120** to **201** of length k=3 is: (120 2C

Example B: shortest route

A route from node **120** to **201** of length 1 is: (1201)

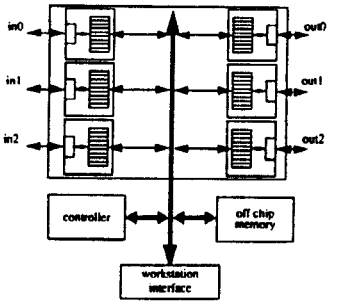


K(2,3)

Rattlesnake switch

Rattlesnake switching node

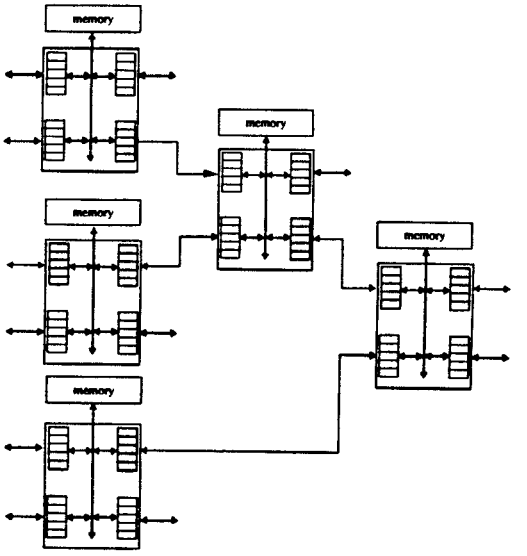
- 6 links per node
- input and output buffering (hard real-time channels)
- off chip memory to buffer ATM cells
- implementation in VLSI (FPGA)



Rattlesnake communication services

- hard real-time traffic: ⇒ video, audio, real-time control, ...
- ATM traffic (non hard real-time): ⇒ file transfer, R.P.C., ...

Rattlesnake communication

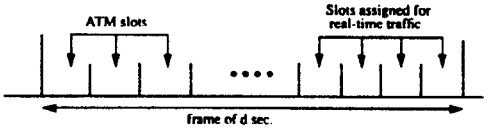


- connection establishment by reserving subsequent buffers
- Red: Hard Real-time Virtual Channel
- Blue: Not used HRVC
- Green: ATM traffic

Rattlesnake transfer mode

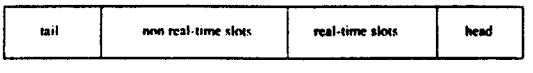
Transfer modes

- Synchronous Transfer Mode (circuit switching)
- Asynchronous Transfer Mode (packet switching)
- **Hybrid Time Division Multiplexing**



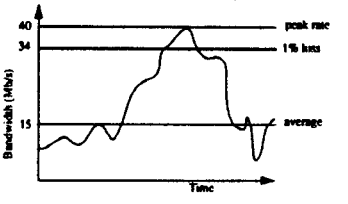
Rattlesnake frame

- switches communicate by exchanging frames
- frame is of variable length
- ***head*** (k bits) indicating full real-time buffers
- real-time slots (≤ k)
- ATM slots
- ***tail*** (k bits) flow control information



Advantages of the rattle transfer mode

- guaranteed bandwidth for hard real-time traffic
- all free bandwidth can be used by ATM
- higher bandwidth available in low-traffic situations



Variable bit rate video coding of a television channel

Realization / current status

- Rattle switch implementation in a FPGA
  - Xilinx XC4005 FPGA (196 CLBs)
  - implementation described in VHDL
- Current implementation restrictions
  - 6 links per switching node
  - 16 hard real-time virtual channels per link
- Time schedule
  - one switching node implemented by the summer of '92
  - network of 6 switches implemented by end of '92
  - demonstrations for industry by spring '93

Conclusions

- High performance, low-latency network for real-time multimedia communications.
- Network based on Kautz topology
  - small diameter
  - fault-tolerant
  - large # of connected nodes ($N = d^k + d^{k-1}$)
- High aggregate bandwidth with a moderate link bandv
- Hybrid TDM scheme used to support real-time communications in a packet switched environment
- Star shaped network
  - low cost point-to-point connections
  - simple workstation interface
- Realization with 'cheap' off-the-shelf components