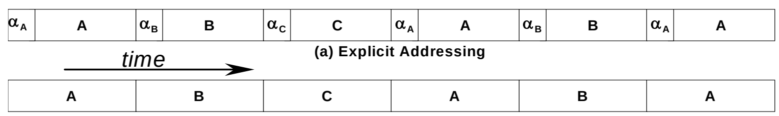
# Summary of SAIN Switching Technology

*Digital communication network technology has evolved in two disparate directions over the past 50 years – one focused on supporting telephone communication and the other on data communication. Attempts to use only one of the approaches to support all communication have proved unsatisfactory. CircuitPath has devised a new paradigm called Synchronous Adaptive INfrastructure (SAIN) networking that combines the benefits of both directions in a simple approach that supports every kind of communication traffic in a single switched-network fabric.*

## The Multiplexing Problem

A fundamental purpose of digital communication is to combine different information flows such as conversations, data file transfers, and streaming audio and video into a single transmission link. The fundamental difference between the way telephone and data networks have evolved rests on two different methodologies by which this combining takes place. The difference is in the way traffic is *multiplexed* onto a transmission medium.

The difference in the two methods is illustrated on the right Multiplexing of data has been based on *explicit addressing* while multiplexing of phone traffic has been based on *implicit addressing*.



(a) Explicit Addressing

(b) Implicit Addressing

In explicit addressing, an information flow is chopped up into pieces called *packets* (or *cells*) where an address *header* designating a packet's destination (and other things) is attached to each packet. In implicit addressing, information is also chopped up into pieces, but in this case, the position of each piece in time designates its address. These pieces are called *cellets*. Each cellet occurs periodically in the same place within a *frame* of cellets. For example, once every eight-thousandths of a second, a one-byte cellet represents a small piece of a telephone conversation.

Early on, implicit addressing was invented to deal with one type of traffic only – voice. Explicit addressing was invented to deal with one type of traffic only – data. In their original conceptions, neither approach anticipated "convergence", the idea that all types of traffic can be accommodated in a single network. Two cultures emerged.

However, about 30 years ago, researchers began to focus on explicit addressing technology alone as the basis for building networks that support all manner of voice, video and data traffic. Although thousands of man-years have been expended in this effort resulting in thousands of articles, theses, dissertations and standards documents, the end result leaves a world of complexity and management difficulties.

Whereas the original explicitly addressed data traffic model was to transfer message and data files from one computer to another, the current need has relegated that stereotype to a minor role. Today's traffic consists of information *flows* – web pages, streaming audio and video, telephone communication and even real-time voice and video conferencing. These traffic flows are far different from the original conception of "bursty" data traffic.

The reason for abandoning implicit addressing has been its focus on setting up *circuits* for voice (or other fixed bandwidth) connections that are expected to exist for relatively long periods of time (minutes to years) without modification. This approach often results in wasted communication resources and an inability to adapt to rapid changes in flow characteristics (increase or decrease in assigned bandwidth) of individual connections.

CircuitPath's Synchronous Adaptive INfrastructure (SAIN) technology combines both implicit and explicit addressing into a single, simple-to-understand and simple-to-implement networking paradigm that overcomes the deficiencies of today's technologies.

SAIN technology makes effective use of the most important aspect of networking based on explicit addressing – the ability to connect a number of flows simultaneously from a single point to one or more distant locations. This is the genius of the Internet where, from a single computer, for example, one is able to simultaneously communicate with distant Web servers, send and receive E-Mail, and receive streaming audio and video programs. The Internet Protocol (IP) enables this capability that promises to extend the Internet to a massive scale even larger than it has become.

SAIN technology promises to overcome the two major shortcomings of today's implementations of both explicit and implicit addressing within large networks. These are:

- Explicit-addressing packet switching (as well as ATM cell switching and MPLS) requires traffic buffering internal to a network in order to smooth out traffic flows that are generated and sent to a network asynchronously. This has resulted in using "routers" that have become increasingly complex as they attempt to handle many flows simultaneously and assure that internal traffic buffering does not produce delay variations that result in undesirable interruptions to streaming traffic. Basic routing functionality will always be required, but can be divorced from traffic buffering.
- Even though implicit addressing, as currently implemented, does not suffer from delay variation problems, it is cannot handle the wide diversity of traffic types that have emerged. Connection setup is not fast enough, and once set up, the amount of bandwidth dedicated to a connection is fixed.

The way that SAIN technology overcomes these severe limitations is:

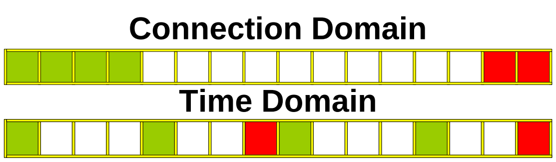
- to place all traffic buffering at ingress port connections to a large network. This eliminates the need for complicated routers internal to a network and
- to implement implicit addressing multiplexing at all ingress, egress and internal switching nodes. SAIN technology uniquely allows connections to use to be set up in connections in milliseconds, and, once established, change their bandwidths can be changed in milliseconds.

Although these two aspects of SAINs are simple to implement, their results are profound.

## How SAIN Multiplexing Works

SAIN multiplexing results in switch uniqueness. It uses Time Division Multiplexing (TDM), but, is unlike presently operational TDM networks. Current systems use *hierarchical-based* multiplexing; simple *transform-based* multiplexing is the SAIN approach;. Hierarchical TDM starts with dividing a large bandwidth into a multiplicity of smaller bandwidths. This process may be repeated in a layered fashion to achieve some mandated standard data rates. Standardized hierarchies are the basis of today's Public Switched Telecommunications Network (PSTN).

Hierarchical Time Division Multiplexing cannot meet the needs of modern networks. CircuitPath's transformed-based multiplexing can.




The basic idea of SAIN technology divides large Time Division Multiplex frames of small time slots called *cellets*. Each cellet can be of any fixed length for a link's TDM frame. The size on a tandem link can be different.

Cellets are addressed in two ways as determined by a very simple mathematical transform. One is called the *Connection Domain* where a contiguous address range of cellets is assigned to establish a single connection.

The other is called the *Time Domain* that is mechanized using the simple SAIN transform. The purpose of the Time Domain is to spread cellets nearly uniformly throughout a TDM frame. By so doing, latency through a multiplexing or switching node is minimized♦.

This simple transform-based method of assigning bandwidth is very powerful. Not only can connections be made within milliseconds, each with a different bandwidth, but, once established, any connection's bandwidth is easy to change in milliseconds as well.

A single multiplexing and switching connectivity *underlayer* (Layer 1.5 within the industry-standard protocol hierarchy) is used end-to-end within a SAIN. The Layer 1.5 building block can be embedded within a family of semiconductor and software products, results in SAIN technology's superior network-wide performance.

♦ *In the Connection Domain, cellet positions are given binary number addresses that start at zero and increase by one for each successive position. Each Connection Domain cellet address is transformed to a Time Domain cellet address by reversing the order of the denoting the cellet's Connection Domain address. This transform works both ways, so that each sequential cellet Address of a TDM frame can be transformed into an Element Address that is easily associated with its Connection.*
*The length of the Connection Domain is 16 cellets that is shortened to 14 cellets by eliminating the last two addresses. The corresponding cellets are set to zero length in the Time Domain so that it is also 14 cellets in length..*