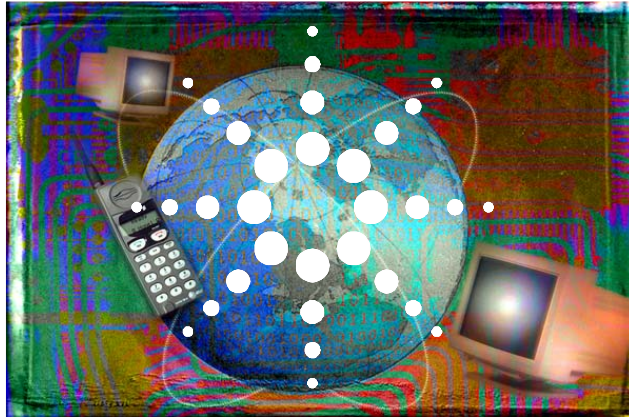


# Communications Bus Analysis - Real-Time Trace of Packet Data



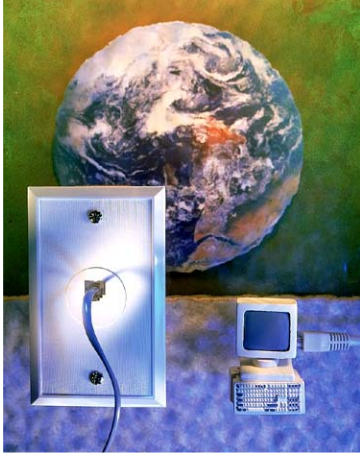
## Authors/Speakers

Scott Ferguson earned his BSEE and MSEE degrees from Bucknell University in 1986 and 1989, respectively. He has spent some years doing scientific computer graphics at Exxon Corporation as well as the Cray Computer Corporation. Scott joined Hewlett-Packard Co/Agilent Technologies in 1995 and spent 3 years developing system software for the HP16505 and 16700 logic analysis systems.

At the introduction of the 16700 system I spent 6 months performing application engineering duties and sales force training in the European Marketing Operation in Amsterdam. Scott created the DataComm Analysis Toolset, In his spare time, Scott plays guitar and saxophone in a local rock band, as well as enjoying golf and tennis.

John Marshall has worked for Hewlett-Packard Company/Agilent Technologies for over 25 years in both R&D and marketing positions. He was involved in the original work on logic analysis and on the HP 64000 development system. He received his BSEE from Texas Tech University. John is currently the Program Manager responsible for the emulation and debugging products at the Colorado Springs division. In this position he has developed and delivered presentations and published articles in the U.S. Europe and Asia. In his spare time, John enjoys photography.

# Introduction



## Major Developments in the Comms Industry

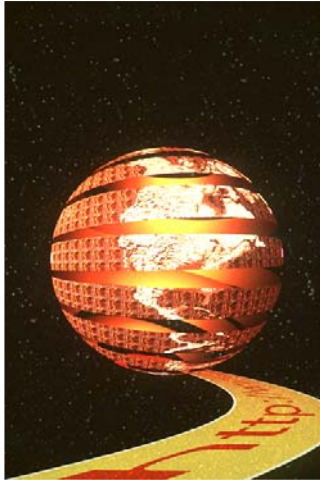
- **Rapidly accelerating advances in technology driving faster line rates**
- **High line speeds require parallel buses on board**
- **Protocol & Network analyzers only see the outside of the box**
- **New tools and strategies required**

The development of ever faster telecommunication networks is driving a need for new tools for designers of network switching systems and host adapters. As line data speeds continue to rise, it becomes impossible to transfer data between chips and across circuit boards as a single serial line.

Therefore parallel buses are employed at various stages on switching system line cards and through switch fabrics. To leverage commercial parts, standards are developed for these parallel buses.

Since protocol analyzers cannot probe the packet signals as they travel across these parallel buses, new tools are required to enable visibility of packet traffic on them.

## Bandwidth Drivers



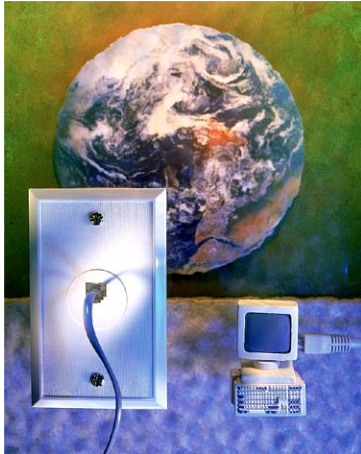
- **Number of Internet Users Doubling Every Year**
- **Amount of Data is Tripling Every Year**
- **Number of Domain Names: 3 Million**
- **Number of Web Sites: 1.5 Million**
- **Number of Web Pages: 350 Million**

Looking at the growth in the internet in another way, the implications for our industry are truly astounding:

- Number of Internet Users Doubling Every Year
- Amount of Data is Tripling Every Year
- Number of Domain Names: 3 Million
- Number of Web Sites: 1.5 Million
- Number of Web Pages: 350 Million

Switch designers must do all they can to build faster switches, to meet the incredible demand for bandwidth.

## Trends



### Higher Throughput/Higher speed

- To increase data throughput, the data path is separated from the processor path
- Lower voltages & differential signals (+ fiber optics)

### Move from proprietary technologies to standards

- Buses : PCI, Utopia, POS-PHY
- SW : OS, Libraries, Protocols

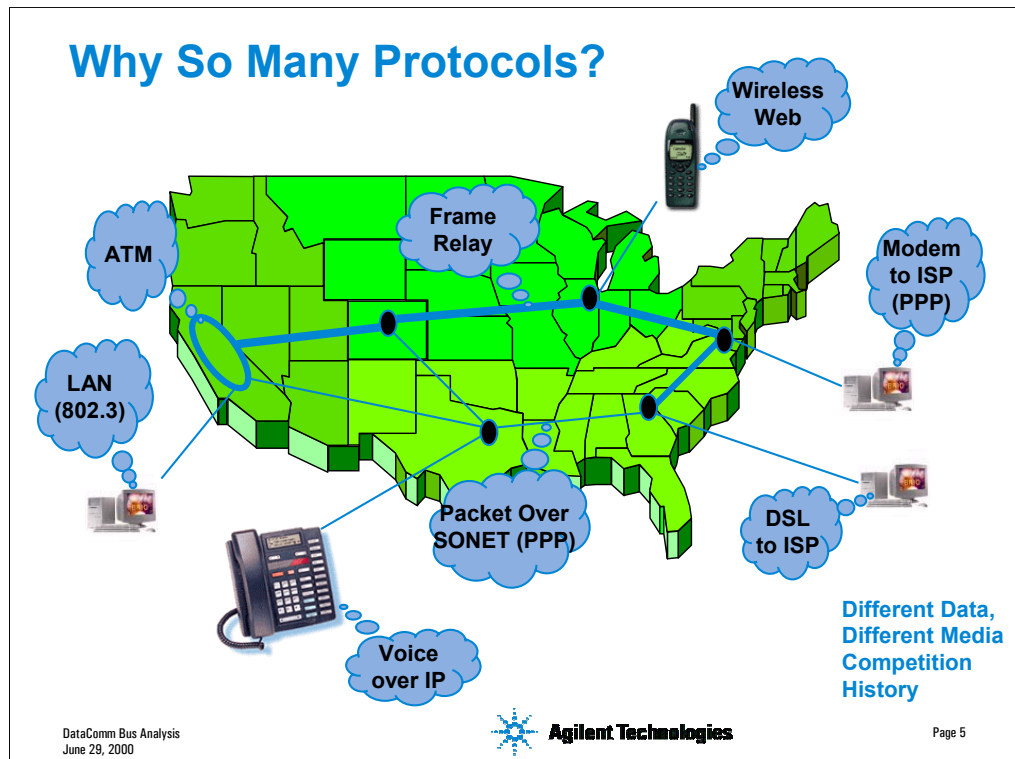
The changes taking place within the telecom industry are driving corresponding changes in the way that we design, debug and test new products.

Early routers were basically built from microprocessors, which would store incoming packet data in memory while determining where to route the packet. As performance demands increase, the microprocessor and memory have become the bottleneck. Current system designs now move packet data through ASIC's. While a microprocessor is still common in these systems, they are used mostly for "housekeeping" chores like powerup configuration and system management.

Standards are increasingly employed so that components are interchangeable, and competing products can work together.

Also, increasing reuse of core components (microprocessors, bus interfaces, DSP's) allows designers to turn on new products in shorter times.

## Why So Many Protocols?

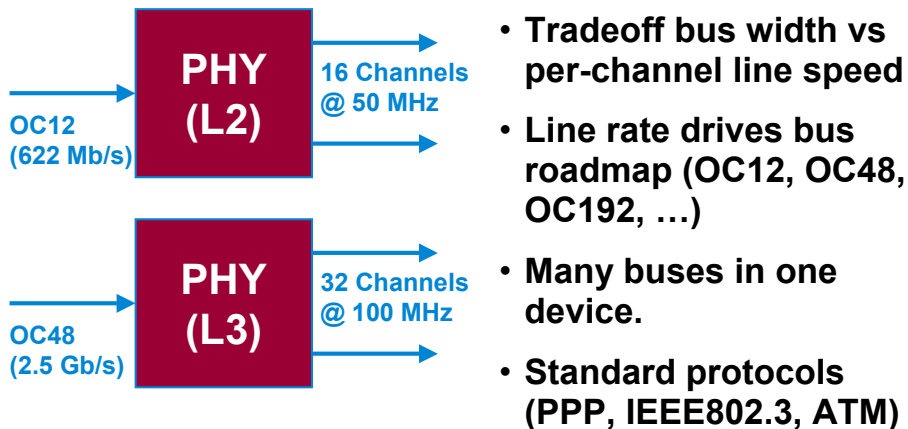


The proliferation of different protocols comes from numerous sources:

- Different data types (such as voice, video, web traffic) have different needs
- Communications technology has been evolving for many years
- Companies have differentiated their products through innovation
- Data travels across very different media (wireless, fiber optic, copper, etc)

So as communication services converge onto a single integrated network, these protocols must co-exist and the network must be able to handle very different needs effectively.

## Communication Buses



Communication buses have evolved over recent years. As the line speed has quadrupled with each generation (from OC-3, to OC-12, to OC-48, etc) communication buses have kept up by doubling the parallel bus width and doubling the bus clock speed.

As line speeds continue to increase, doubling the bus width becomes very difficult, as circuit board routing resources become quickly exhausted.

Also, as bus speeds increase, problems resulting from cross-talk, EMI and heat dissipation make it more difficult to simply increase the clock speed.

Future buses will turn to LVDS (low-voltage differential signaling) to increase the speeds, while reducing power consumption and interference.

## System Integration Issues



- **Team communication.**
- **System-level timing problems.**
- **Component-level timing problems.**
- **Intermittent operation.**
- **Noise and glitches.**
- **Questions about the source of problems**

System integration is when all the parts of a product design come together. The parts can be custom or commercial chips, circuit boards, system boot code, and application software.

The types of problems that the design team encounters during the system integration phase vary with the speed and complexity of the target and with the level of progress through the debugging cycle. Initial problems include those caused by miscommunication and invalid assumptions. These are usually discovered and quickly eliminated. From this point the problems become subtler and much harder to isolate and resolve. Race conditions or timing problems can exist under certain operating conditions or when components operate outside their specifications. Noise and glitches may introduce false signals and/or data. Even the probing of the target with instrumentation can introduce problems at high speeds by introducing reflections and improper terminations. Complicating the search for problems are questions about whether their source stems from hardware or software. The addition of buses such as Ethernet and PCI that connect to external devices complicates the debugging process even more.

## Comprehensive Time Correlated Measurements

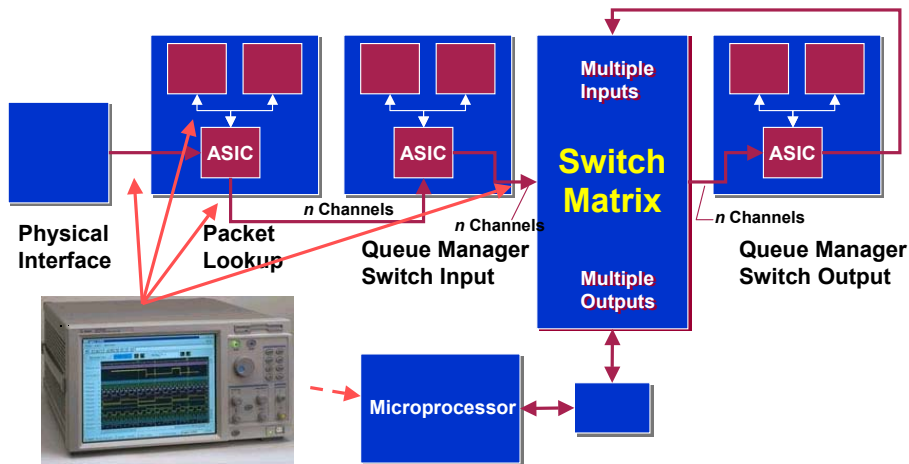


- These measurements are time correlated for
- establishing cause/effect relationships.

To make real-time measurements on a complex system that may include many high speed buses, we need an analysis system that can make simultaneous measurements on multiple buses and correlate the results. Not only do these logic analysis systems have the capability to capture and display data from your target in many different ways, but the various measurement displays can be time-correlated to help establish cause/effect relationships.



## Example Switch Architecture

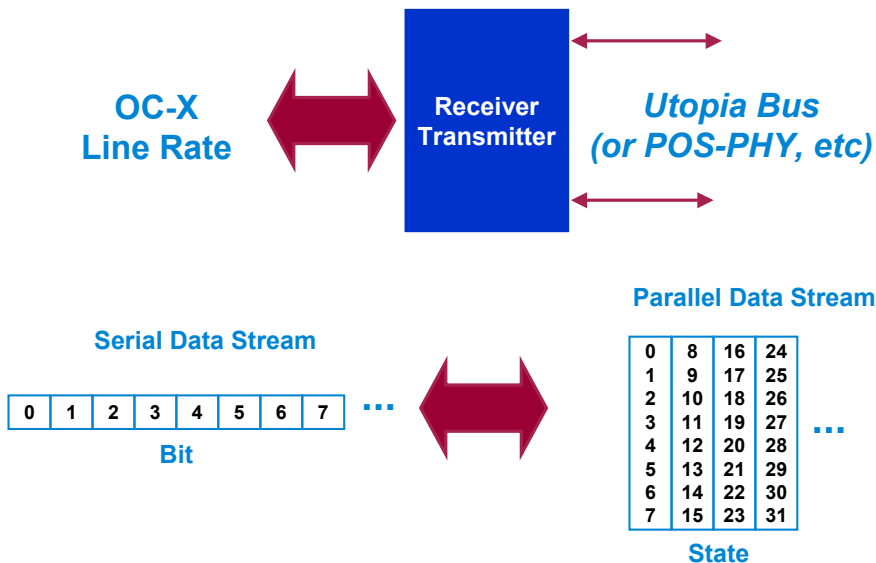


As an example consider a typical switching or routing system. The logic analyzer probes the parallel data path buses in the system. All these buses may be time-correlated in the logic analyzer. The information provided helps the design team to evaluate what the system is doing and to evaluate how the various components are interacting. System timing can be evaluated and correlated to the specification.

All packets or cells are time-stamped in the logic analyzer for time-correlation measurements with other system buses such as a microprocessor, memory interface, PCI bus, or other UTOPIA bus. All state listing and waveform displays in the logic analyzer are time-correlated with global markers for a complete view of the system. With this tool, it is possible to trigger the logic analyzer with a microprocessor event and see what is happening on a parallel data bus with protocol information.

By monitoring multiple time-correlated data buses, you can monitor a packet entering one ASIC and see how long it takes for the packet to reach another part of the system. The powerful trigger can also monitor a packet entering one port and trigger if the packet has not reached another port by a designated time.

## Parallel Serial Packets

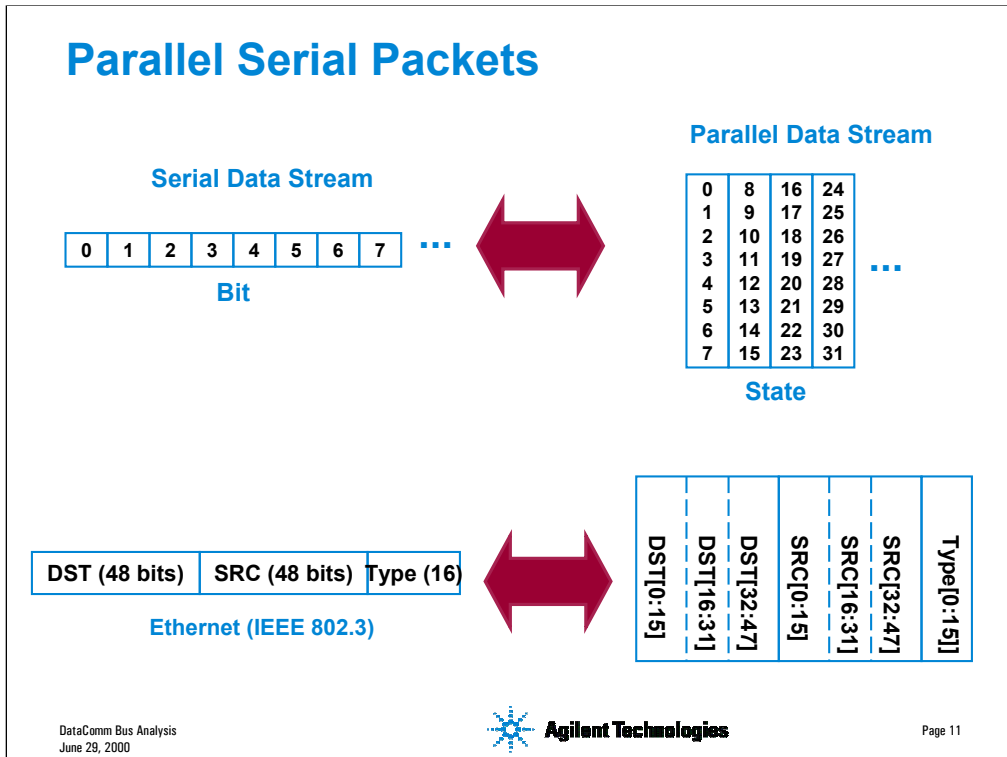


The data buses such as the UTOPIA bus on the communications processors are parallel serial buses. In the example above, the first eight bits on the serial line are demuxed into eight individual channels. The logic analyzer will probe all eight of these channels with other qualifying signals and a synchronous clock.

Your networking hardware uses parallel data buses such as UTOPIA or a proprietary parallel interface to communicate between communications processors, network processors, custom ASICs, or physical interface chip sets. The data communications tool set allows developers to view parallel bus data at a protocol level on the logic analyzer.

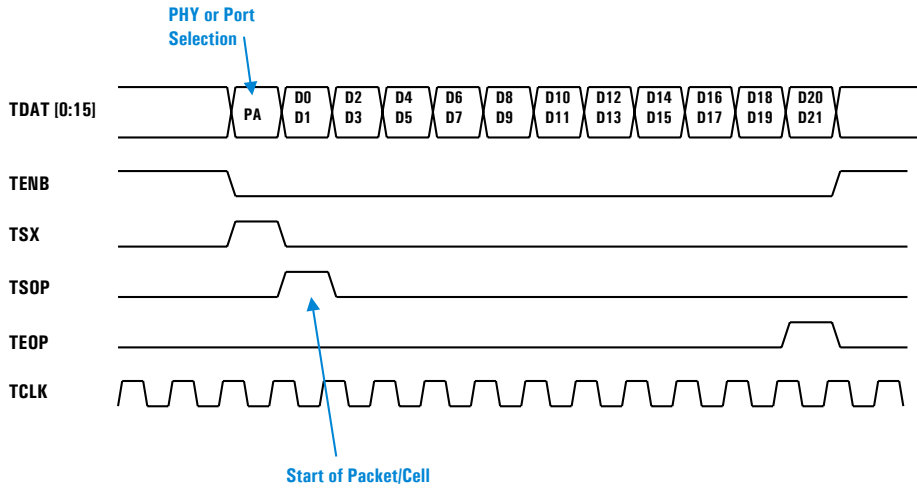
With a higher abstraction view of the data combined with the powerful time-correlation features of the logic analyzer, Agilent Technologies equips designers with a new debugging capability to find complex system-level bus interaction problems combined with protocol information from parallel data paths.

# Parallel Serial Packets



When serial packet data is parallelized onto a bus, the protocol structure of the packet remains the same, only the shape is changed slightly. Now  $n$  bits occur on a single clock strobe (where  $n$  is the width of the bus) instead of just one.

# Typical Utopia or POS-PHY Bus



Buses such as Utopia and POS-PHY typically have  $n$  bits of data, a synchronous clock, a *Start of Packet* (or *Start of Cell*) signal, and a *Transmit Enable* signal.

