A Summary of the ZIPI Network
Author(s): Keith McMillen, David Simon and Matthew Wright
Source: *Computer Music Journal*, Vol. 18, No. 4 (Winter, 1994), pp. 74-80
Published by: The MIT Press
Stable URL: http://www.jstor.org/stable/3681359
Accessed: 09-04-2017 21:35 UTC

A Summary of the ZIPI Network

Keith McMillen,* David Simon,† and Matthew Wright*††
*Zeta Music/Gibson Western Innovation Zone
2560 Ninth St., Suite 212
Berkeley, California 94710 USA
†Probitas Corporation
2570 El Camino Real, Suite 310
Mountain View, CA 94040
††Center for New Music and Audio Technologies (CNMAT)
Department of Music, University of California, Berkeley
1750 Arch St.
Berkeley, California 94720 USA
McMillen, Matt@CNMAT.Berkeley.edu

The ZIPI network specification defines a collection of communication protocols intended for musical instruments and similar devices. The network is laid out as a star-shaped token-passing ring with a hub in the center. One device, typically the hub, monitors the general health of the network. The protocols constitute a stack that conforms to the open systems interconnection (OSI) model.

This is a technical introduction to the ZIPI network that summarizes the capabilities of the protocols and the full specification. We explain briefly what the ZIPI network does, how it works, how a network of ZIPI devices is connected, and what is involved in building instruments to run in conjunction with ZIPI networks.

Body and Soul

In our opinion, the key feature of ZIPI is our definition of the Music Parameter Description Language (MPDL), an application-level protocol for sending musical information. A complete network standard must, of course, define a physical layer as well as the application layer, so this article describes an appropriate physical layer that can handle the bandwidth and determinacy required for a real-time interface.

Computer Music Journal, 18:4, pp. 74–80, Winter 1994

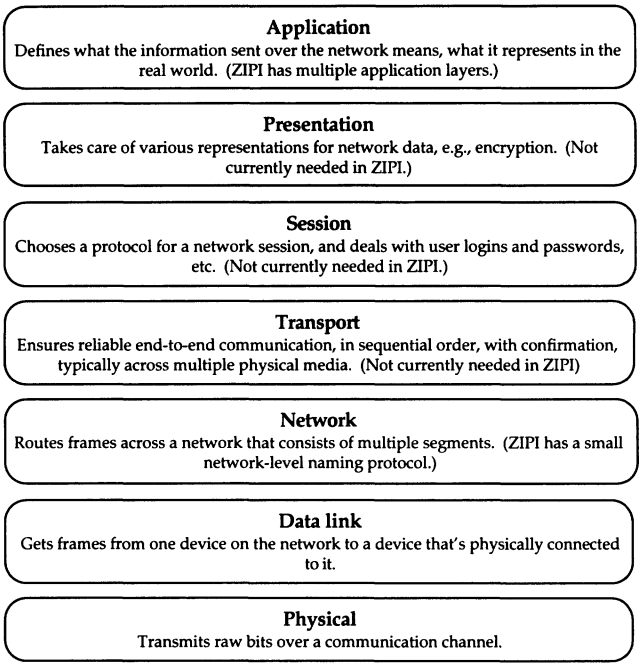
Great strides are being made in networking technology with regard to performance, cost, and features. Ethernet (Tanenbaum 1989, section 3.4.1) is now common on many personal computers, and there are fast, deterministic versions of it. In fact, our first prototype MPDL implementation is based on IP/UDP (Tanenbaum 1989) and Ethernet. We are carefully keeping the nature of the physical implementation separate from our specification of the MPDL to allow the MPDL to run on other networks, such as Apple's announced 150-MBaud serial protocol Firewire (Teener 1994) and Lone Wolf's Medialink (Lacas, Moses, and Warman 1993).

Unfortunately, all of these physical layers are currently much more expensive than the US$ 5 that ZIPI interface hardware costs. We cannot imagine that ZIPI will gain widespread acceptance if manufacturers must add hundreds of dollars to the price of an instrument to make it ZIPI-compatible. Therefore, we have chosen a physical layer that has acceptable performance at a very low cost.

Open Systems Interconnection (OSI)

OSI was developed by the International Standards Organization as a model of how to divide computer networks into various conceptual layers (Tanenbaum 1989). ZIPI's organization conforms

Figure 1. The seven layers of OSI.



to this model. OSI's seven layers and their functionality are shown in Figure 1.

A possible physical layer and data link layer have been fully defined. In addition, a network-layer naming service is specified, and various application-layer protocols have been defined. Routing, transport, session, and presentation protocols are not specified, as we see no immediate need for them, but hooks have been left to allow the inclusion of such protocols.

Features of the ZIPI Protocols

The following features of the ZIPI protocols are the most notable.

Peer-to-Peer Architecture

Any device can send frames directly to any other device on the ring or to all devices at once. Up to 253 devices can be on the ring.

Low Cost per Node

The hardware required to implement the network on a node consists of a serial controller chip, a small PAL, and a dual opto-isolator. The total cost is under US$ 5.

Low Development Cost

We will provide sample schematics for the hardware, the PAL equations or already-programmed PALs, and most of the software needed to implement the lower network layers and much of the MPDL. A manufacturer need only provide software to deal with the hardware-dependent aspects of its node.

Serial Communications Chip

Most of ZIPI's lower levels are taken care of by the 8530 serial communications chip (SCC), an inexpensive, plentiful chip found in Apple Macintosh and Silicon Graphics computers. It is a dual device and can support both MIDI and ZIPI at the same time.

Compatibility

The ZIPI protocol works with other protocols. Specific provision has been made to carry MIDI data over the ZIPI data-link protocol. Hooks have been left to run other protocols as the market requires.

Open Architecture

We have published the draft specifications for the protocols and will continue to publish revised specifications.

Efficiency

The protocols allow the most common information to be passed in very small frames. The token-

Figure 2. ZIPI hub configuration: physical interconnect.

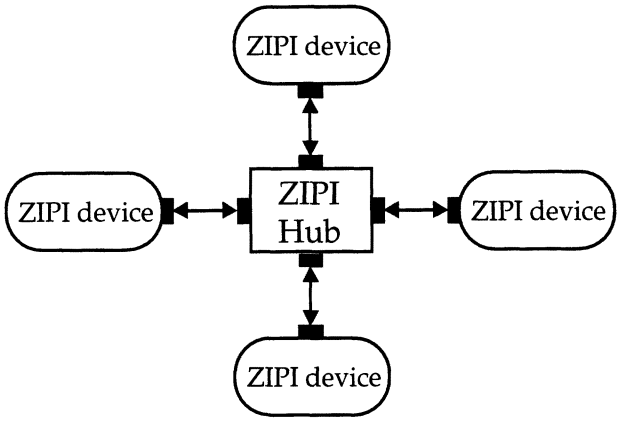
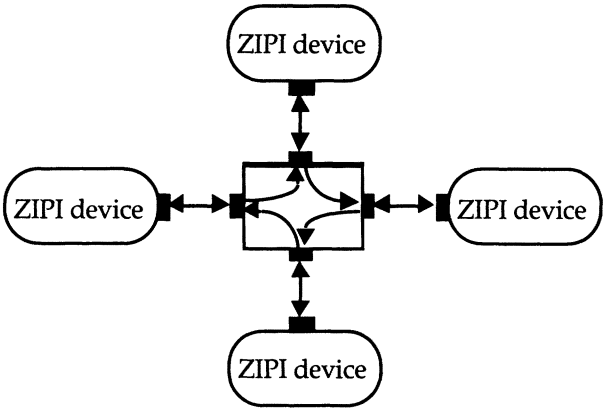


Figure 3. ZIPI hub configuration: logical interconnect.



passing network scheme uses most of the network bandwidth for transmitting data. Typical applications would be on the order of 95 percent efficient.

Determinism

The token-passing ring guarantees that each node will get a chance to send data whenever the token comes around the ring. Once a device has the token, the data that it transmits is guaranteed to arrive before a certain time because no device has permission to talk while another holds the token.

Fault Tolerance

The network does not fail if one of the nodes fails; protection in the hubs will simply remove a failed node from the ring.

OSI

The protocols conform to the OSI model, allowing ZIPI to interface with other networks, such as Ethernet or FDDI.

Expandability

ZIPI has been designed to allow the addition of new features to existing protocols or the addition of entirely new protocols.

Speed

The network operates at speeds of 250 kHz and up. (Presently available hardware will support up to 20 MHz.) The data rate is variable, depending on the capability and requirements of the devices on the network. Initial turn-on Baud rate is 250 kBaud, but there is a software protocol for bringing the Baud rate up to any speed that all devices can handle. Another protocol allows two high-speed nodes to increase the network speed temporarily beyond the limits of slower devices.

Small Processor Load

Each host processor is interrupted only when it receives a frame addressed to it. Only the monitor device must see every frame.

Minimal Cabling Required

Each ZIPI device needs only one cable to connect it to the network. ZIPI cables will be inexpensive.

Network Functionality Overview

The ZIPI protocols are divided into layers according to the OSI model. The following sections summarize network operations at the various layers.

Figure 4. ZIPI standard plug pin-out.

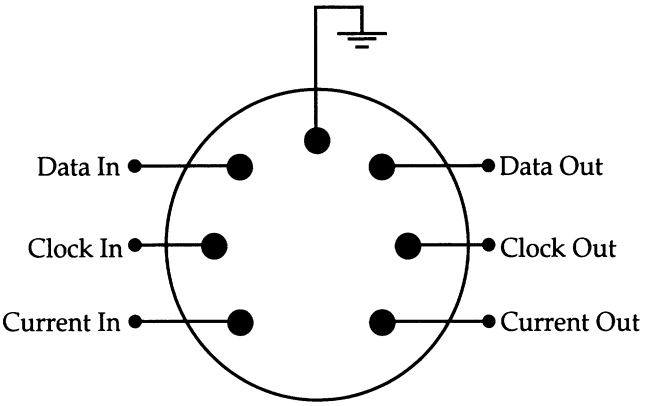


Figure 5. ZIPI mini-DIN pin-out.

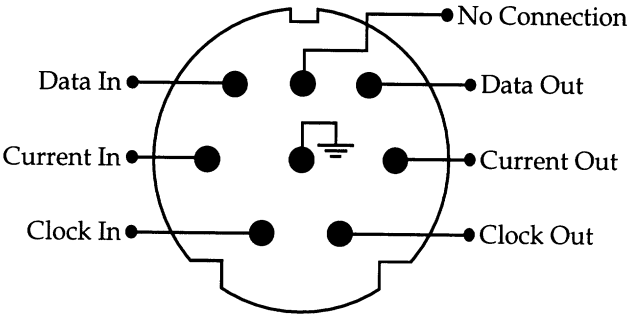


Figure 5

Figure 6. ZIPI hub detailed diagram.

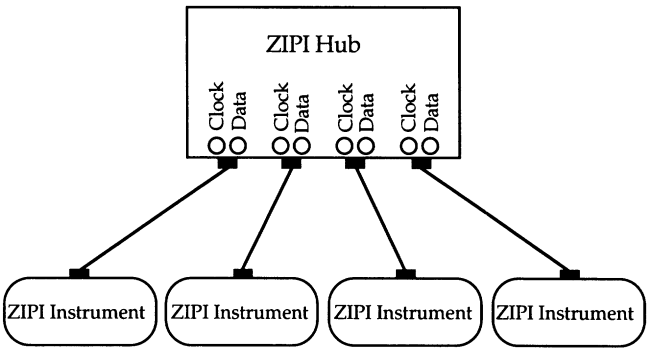


Figure 6

Physical Layer

Logically, ZIPI devices are connected in a ring, and each device passes data to the next one around the ring. Physically, the devices are connected in a star-shaped configuration in which each device is connected to an active "hub" at the center of the star, as shown in Figure 2. The hub maintains the logic of a ring by sending the data coming from each device out to the next device in the star, as shown in Figure 3. Each connector on each hub has a set of relays to bypass that connector if no instrument is attached or if the attached instrument has failed or has been turned off.

Devices are connected to the hub by a seven-wire cable with two directions of ZIPI data flowing through it. Each direction has a clock, data, and current line; the seventh wire is for shielding the entire cable. Each cable end is terminated with a seven-pin DIN plug; the pin-out for seven-pin ZIPI plugs is shown in Figure 4. ZIPI cables can also be terminated with an eight-pin mini-DIN connector to allow ZIPI interfaces to be built into laptop computers. In that case, the pin-out is as shown in Figure 5.

By having one cable carry ZIPI information in two directions, each device will need only one female seven-pin DIN connector. ZIPI hubs will have multiple female seven-pin DIN connectors, all equivalent. A device can be attached to a hub using a male-to-male seven-pin DIN cable. Hubs can be attached to one another simply by cabling any connector on one to any connector on the other. ZIPI hubs will have two LEDs per ZIPI plug, with the following meanings: (1) A device is connected and is sending a clock. (2) The connected device is sending data. These LEDs should make it very easy for musicians to debug problems in their ZIPI setup, such as malfunctioning devices, bad cables, or loose connections. Figure 6 shows a typical hub configuration.

Like MIDI, ZIPI uses an opto-isolated current loop. Up to 253 devices can be connected on a single ring. The total distance from one device to the hub can be up to 300 meters.

The preferred implementation of a ZIPI network device is based on the 8530, an inexpensive serial communications controller chip available from Zilog and from AMD, or on one of its close relatives (Zilog 1992). The 8530 running in "SDLC Loop Mode" implements most of the ZIPI physical layer protocol as well as some of the ZIPI data link layer protocol. The 8530 automatically handles most of the Synchronous Data Link Control (SDLC) protocol (Tanenbaum 1989, section 4.7.1),

including device addressing, host processor interrupts only for matched addresses, data framing, CRC error checking, and hardware arbitration of who gets to talk when. It has a 4-byte FIFO buffer and supports DMA (direct memory access).

Data is sent packaged in SDLC frames. A token circulates around the ring; devices are allowed to transmit only when they receive the token. When a device has completed its transmission, it must pass the token on to the next device on the ring. Because each device gives up the token as soon as possible, the token goes all the way around the ring many times in a second, ensuring that it is never very long before any particular device gets a chance to talk.

Each ZIPI network must also have one device that implements a monitoring function to keep the network running smoothly. (Initially, only ZIPI hubs will be capable of this monitoring function, not other devices, because each network must have a hub anyway, just for connectivity. Putting monitor functionality in the hub keeps the cost down for other devices.) First, the monitor provides a clock for all ring communications; devices on the ring can negotiate the speed of the clock. Second, the hub monitors the ring to ensure that the token is circulating and that no garbage circulates on the ring.

It is possible to have multiple monitor-capable devices; the software protocols elect a single monitor from among the qualified devices. It is also possible to have passive hubs that provide connectivity with no monitor capabilities.

When a ring of ZIPI devices is formed, an automatic start-up sequence begins. First, the ring monitor is elected. The clock rate is determined, and all the other network setup occurs automatically within a second of connecting the ring. From the user's point of view, just turn everything on and plug everything in, and ZIPI will work fine.

Data Link Layer

The data link layer provides the following services in addition to sending and receiving frames:

1. It ensures that data has been received correctly by checking the CRC included with each frame and discarding frames in which the CRC is bad.
2. It sends acknowledgments of received frames (at the option of the sender).
3. It establishes an address for its device that is unique among devices on the network.
4. It negotiates with the other devices on the network to determine the clock speed at which the network runs. The data link layer in the device monitoring the ring also ensures that there is a token on the ring and that no garbage is on the ring, and it notifies other devices when the clock speed is changing.
5. It synchronizes the real-time clocks of all devices on the ring to within 50 μsec.

Network Layer

ZIPI includes a network-layer naming protocol to allow devices on the network to find one another either by name or by any of a large and expandable list of characteristics. For example, the naming protocol allows a device to search the network for synthesizers that can accept data at 1.0 Mb per second or faster and that run ZIPI protocol version 1.1 or later.

Application Layer

ZIPI will contain multiple application layer protocols. They include the following.

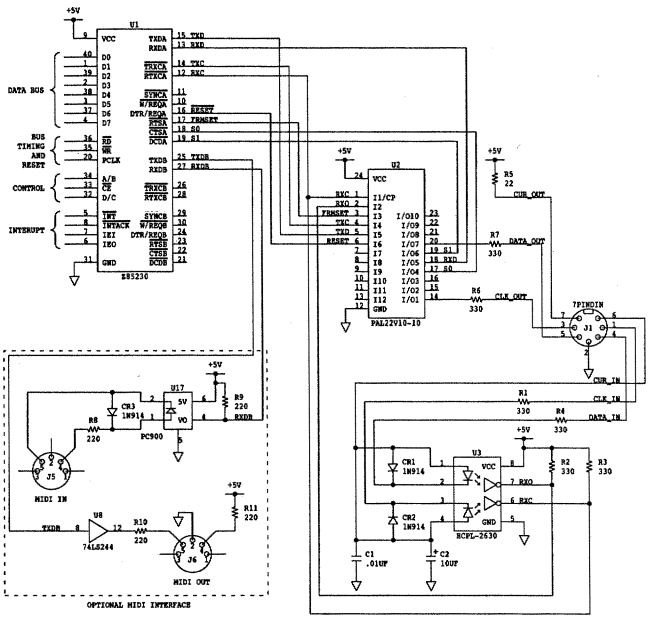
The Music Parameter Description Language (MPDL)

The MPDL is a language for describing music. It delivers musical parameters (such as articulation and brightness) to notes or groups of notes. (This language is fully described in another article in this issue.)

MIDI

The MIDI protocol can be sent through the ZIPI network, and the data can be used at the other end.

Figure 7. Schematic for ZIPI network hardware.



Patch Dumps

Patch dump messages will include the manufacturer ID, instrument ID, patch name, and patch number, in a universal format, followed by any arbitrary amount of manufacturer-specific binary data. This allows any patch librarian to work with any ZIPI synthesizer without knowing the patch data format.

Sample Dumps

There will be a standard format for sending digital audio over ZIPI, with a file format to be determined.

Data Dump Protocol

A format will be defined for transferring unrestricted binary data, such as memory dumps, intercomputer communication, and compiled software, over ZIPI.

Error Messages

ZIPI devices with limited user interfaces can send ASCII-encoded error messages, which will be picked up and displayed to the user by another device, e.g., a computer.

Other Application Layer Protocols

Other application layer protocols may include the following: mixer automation, lighting, and effects control (yet undefined); machine control/synchronization (yet undefined); and images, sound, and gestures for virtual reality (yet undefined). There is room to add potentially thousands of more application layers in the future.

Other Layers

Hooks have been left for other OSI protocol layers, but these protocols are not currently defined.

Building a ZIPI-Capable Device

A ZIPI device needs the capabilities of the 8530 SCC running in SDLC loop mode. In general terms, a ZIPI device must be able to send and receive SDLC frames and to recognize and capture the token as it circulates around the ring. A sample schematic for the ZIPI network hardware is shown in Figure 7. The following parts are required for a network device:

- an 8530 SCC or equivalent;
- an inexpensive 22V10 PAL (the equations for which are available from Zeta Music);
- a fast dual opto-isolator (e.g., the Texas Instruments HCPL2630) for the receive data and receive clock lines;
- and miscellaneous resistors, capacitors, and diodes.

We will provide the source code in C for the data link layer and some useful routines for implementing the upper layer protocols. The following software must be provided by the manufacturer: (1) an interrupt routine for the 8530 that pushes the necessary registers, calls the routines provided by us, and resets any interrupt control hardware other than that in the 8530; and (2) software to set up a timer, handle the timer interrupt, push the registers, and call the pre-defined routine. In addition, our source code must be configured with

such items as the address of the 8530 SCC and the frequency of the timer interrupt.

Building a ZIPI Monitor

In addition to the functionality required of every ZIPI device, the monitor must implement the following: a 16-bit delay in the ring in order to float the token (although in practice this delay may be somewhat longer); resynchronization of the data to deal with the problem that the clock the monitor uses to send data might not be in phase with the clock it uses to receive data; recognition that the token has been lost and the ability to then put a new token on the ring; the protocol for electing a ring monitor when several monitor-capable devices are connected to the ring; and the protocol for picking a clock speed.

For More Information

For a detailed specification of the network described here, please contact the authors.

References

Lacas, M., B. Moses, and D. Warman. 1993. *The Media-Link Real-Time Multimedia Network*. Conference Preprint 3736. New York: Audio Engineering Society.

Tanenbaum, A. S. 1989. *Computer Networks*. Englewood Cliffs, New Jersey: Prentice Hall.

Teener, M., ed. 1994. *High Performance Serial Bus*. P1394/Draft 6.8v1. New York: The Institute of Electrical and Electronic Engineers.

Zilog. 1992. *SCC User's Manual*. Campbell, California: Zilog Corp.