

## SatCat5: A low-power, mixed-media Ethernet network for smallsats

Alexander Utter, Mark Zakrzewski, Andrew Keene, Samuel Dietrich, Sammy Lin, Eric McDonald, Nathan Whitehair, and Jason Zheng  
 The Aerospace Corporation  
 2310 E. El Segundo Blvd., El Segundo, CA 90245; 310-336-0753  
 alexander.c.utter@aero.org

### ABSTRACT

In any satellite, internal bus and payload systems must exchange a variety of command, control, telemetry, and mission-data. In too many cases, the resulting network is an ad-hoc proliferation of complex, dissimilar protocols with incomplete system-to-system connectivity. While standards like CAN, MIL-STD-1553, and SpaceWire mitigate this problem, none can simultaneously solve the need for high throughput and low power consumption.

We present a new solution that uses Ethernet framing and addressing to unify a mixed-media network. Low-speed nodes (0.1-10 Mbps) use simple interfaces such as SPI and UART to communicate with extremely low power and minimal complexity. High-speed nodes use so-called “media-independent” interfaces such as RMII, RGMII, and SGMII to communicate at rates up to 1000 Mbps and enable connection to traditional COTS network equipment. All are interconnected into a single smallsat-area-network using a Layer-2 network switch, with mixed-media support for all these interfaces on a single network. The result is fast, easy, and flexible communication between any two subsystems.

*SatCat5* is presented as a free and open-source reference implementation of this mixed-media network switch, with power consumption of 0.2-0.7W depending on network activity. Further discussion includes example protocols that can be used on such networks, leveraging IPv4 when suitable but also enabling full-featured communication without the need for a complex protocol stack.

### INTRODUCTION

Unique interfaces are tempting but costly. While bespoke designs can yield lower size, weight, and power (SWaP) for a given function, the complete system and its developers may suffer in aggregate. Excessive customization leads to growing numbers of mutually incompatible interfaces. Each additional unique interface adds monetary and schedule costs for design, test, documentation, problem-solving, and ground-support equipment. In addition, this proliferation of interfaces adds mission risk due to the growing complexity of new subsystems with multiple interfaces. High complexity leads to additional edge cases that are difficult or impossible to test thoroughly.

Adoption of standards is the best way to mitigate these costs—both on a project-by-project basis and industry-wide.

This paper evaluates standards for passing messages between systems within a spacecraft. Such local-area networks are sometimes referred to as a “bus” regardless of whether the implementation is truly a shared-access parallel electrical bus. Notable prior efforts in this area include the Jet Propulsion Laboratory’s “X2000”

program<sup>1</sup>, which successfully adopted COTS standards for high-reliability deep-space applications.

However, most industry standards for computer networks are ill-suited to the stringent size, weight, and power constraints of cubesats and smallsats. As a result, there is no widespread consensus or adoption of such technologies among the smallsat community.

### SURVEY OF EXISTING STANDARDS

The authors recognized a growing need to adopt a useful local-network standard for their own smallsat designs. The first steps were to identify needs for near-future missions, survey existing industry standards, then choose one or more to meet as many objectives as possible.

A poll of stakeholders within The Aerospace Corporation identified the following near-term needs:

- Allow direct communication from any device to any other device.
- Isolate faults caused by a single malfunctioning device.

- Improve modularity (i.e., ability to easily integrate new avionics and payloads).
- Facilitate high-speed data transfer (at least 622 Mbps for laser downlink<sup>2</sup>).
- Maintain compatibility with low-SWaP microcontrollers (e.g., Microchip PIC family, Vorago VA10820, etc.).
- Maintain very low power consumption (i.e., much less than 1W total in standby mode).

The last requirement was particularly stressing. Many cubesats have a total orbit-average power budget of only a few watts. It is unreasonable to allocate a significant fraction of this budget simply to keep its internal network in standby mode. However, most high-speed interfaces require hundreds of milliwatts for each connected node. This power cost is paid whenever the link is ready, even when no data is sent.

To resolve this dilemma, one solution is to divide the network into separate service tiers. For example, a low-power command and control network that is always ready, and a separate high-speed data network that is activated only on-demand to save power. The survey included a variety of options suitable for this type of two-tiered network. When necessary for comparison purposes, the high-speed network was assumed to have four endpoints.

Survey results are summarized in Table 1. The “MMBD” (multi-master by design) column indicates whether that standard includes built-in provisions to allow concurrent or interleaved communications without a central coordinating authority.

The following sections include a brief description of each option, along with any notable caveats.

### *Aurora / Reflex*

Aurora is a simple, low-overhead protocol for data transfer in point-to-point links, using built-in SERDES in Xilinx FPGAs. Reflex is a similar protocol for Intel FPGAs. Various cross-platform solutions exist with similar properties, typically with an operating power of around 130 mW per transceiver (i.e., one Tx+Rx pair).<sup>3,4</sup> Data rates are limited by the underlying SERDES transceivers, with a single LVDS lane transmitting in each direction.

These protocols are intended as point-to-point links. They have no addressing and no off-the-shelf ability to route traffic. However, given the small number of participating nodes in the high-speed network, even a fully-connected network of point-to-point links is not necessarily prohibitive. A four node-network requires each node to include three outgoing links—for a typical total operating power of 400 mW and 12 pins (i.e., 6 differential pairs) connected to each endpoint.

### *Bluetooth Low Energy*

Bluetooth Low Energy (BLE) is a standard for short-range wireless networking, operating in the 2.4 GHz ISM band. It can operate at 125, 1000, or 2000 kbaud. However, maximum usable application throughput is only 508 kbps.<sup>5</sup>

For a local smallsat network, BLE should be operated in a wired mode, replacing antennas with direct wired connections to a common electrical bus. Resistive

**Table 1: Industry survey summary. Asterisk indicates significant caveats.**

<i>Name</i>	<i>Rate (Mbps)</i>	<i>Wires (Per node)</i>	<i>Topology</i>	<i>MMBD</i>	<i>Power (Per node)</i>
Aurora / Reflex	10,000+	4 (Diff×2)	Pt-pt only	No	400 mW*
Bluetooth LE	2*	Wireless	Peer-to-peer	Yes	25 mW
Bluetooth 4/5	3*	Wireless	Peer-to-peer	Yes	50 mW
CAN	1	2 (Diff)	Linear bus	Yes	50 mW
FlexRay	10	2 (Diff)	Linear/star	Yes	300 mW
Ethernet (100Base-T)	100	8 (Diff×4)	Switched	Yes	260 mW
Ethernet (1000Base-T)	1000	8 (Diff×4)	Switched	Yes	960 mW
Ethernet (RGMII)	1000	12	Switched	Yes	40 mW
Ethernet (SGMII)	1000	4 (Diff×2)	Switched	Yes	145 mW
I <sup>2</sup> C	0.4	2	Linear bus	Yes	< 10 mW
IEEE-1394a	400	4 (Diff×2)	Chained	Yes	400 mW
LIN	0.02	1	Linear bus	No	< 10 mW
MIL-STD-1553	1	2 (Diff)	Linear bus	Yes	Unknown
RapidIO (Serial)	5000	4 (Diff×2)	Switched	Yes	300 mW
RS-485	~10	2 (Diff)	Linear bus	Varies*	< 25 mW
SpaceWire	400*	8 (Diff×4)	Switched	Yes	750 mW
SPI	~10	3+N	Linear bus	No	< 10 mW
TTP/C	25	4 (Diff×2)*	Linear bus	Yes	300 mW
UART-TTL	~1	2 (Tx/Rx)	Linear bus	Varies*	< 10 mW
USB 2.0	480	2 (Diff)	Pt-pt tree	No	150 mW
USB 3.x	5000+	6 (Diff×3)	Pt-pt tree	No	700 mW

power-dividers (simple but lossy) would allow flexible interconnection while maintaining signal strength superior to a free-space link. This allows predictable connections that don't depend on wireless propagation inside a dense cubesat body, keeping undesired RF/EMI emissions to a minimum.

Transceiver modules are rated from Class 1 (100 mW, ~100m free-space range) to Class 4 (0.5 mW, ~0.5m free-space range). Often the power is software-adjustable, to improve battery life. A wired-Bluetooth cubesat is unlikely to need higher than Class 3. Surprisingly, Class 3 or 4 transceivers typically draw less than 25 mW even in continuous transmit/receive operation,<sup>6</sup> which is competitive with many of the wired standards under consideration.

BLE networking is peer-to-peer, though one node is promoted to act as a "master" timing reference. Notably, commercial Bluetooth system-on-modules often include UART, I<sup>2</sup>C, SPI, and other simple serial protocols that would make integration with smallsat microprocessors straightforward. Example drop-in modules include the Rigado BMD-330 or the Insight ISP1507. While superficially simple, the internals of these black boxes are very complex, containing microcontrollers, flash memory, and software that are not designed or tested for the space environment. This raises significant risks that may not be acceptable for a mission-critical network backbone.

#### **Bluetooth 4/5**

Bluetooth Low Energy is a subset of the Bluetooth 4.x and Bluetooth 5 standards. They are intended for safe interoperability (i.e., non-interference), but the BLE and Bluetooth 4/5 networks cannot directly interconnect. Bluetooth 5 supports line rates up to 3 Mbps, allowing user-data transfer rates of 1400 kbps.<sup>5</sup>

Many Bluetooth system-on-chip modules support either BLE or Bluetooth 4/5, depending on the loaded software. (For example, the Nordic nRF52810.) As with BLE, however, these modules contain built-in flash and CPUs that may not survive in the space environment. Power draw for such modules, when operated in Bluetooth 4/5 mode, is typically quoted at around 50 mW.<sup>7</sup>

#### **CAN**

The Controller Area Network (CAN) bus is ubiquitous in automotive applications, supporting robust communications up to 1 Mbaud. Wiring consists of a single differential pair. Low-speed or "fault tolerant" CAN (ISO 11898-3) allows mixed or star topologies, but higher-speed CAN (ISO 11898-2) networks mandate a linear bus with termination at each end.

Natively, CAN is a multi-master network where priority is set by the node's address. Individual frames are short, which ensures prompt access but also leads to high overhead; 64 useful bits in every 108-bit frame gives a maximum useful throughput is less than 600 kbps for bulk data transfer, even before accounting for bit-stuffing.<sup>8</sup>

Many extensions, such as AGATE and TTCAN, exist to add desirable reliability and safety guarantees, including time-triggered protocols that allow strict real-time bounds on transmission times.

Some microcontrollers have direct CAN interfaces; others can take advantage of the many adapter ASICs, which often include queueing, filtering, and other niceties to minimize CPU burden. A typical example of the latter is the Microchip MCP25625, which draws less than 30 mW in receive mode or 230 mW in transmit mode.<sup>9</sup> Since transmission duty cycle is typically low, average power is usually less than 50 mW.

#### **FlexRay**

FlexRay is a newer automotive networking standard, led by BMW, Volkswagen, Daimler, and GM. It is designed to be faster and more reliable than CAN, for use in drive-by-wire and other safety-critical systems. Line rate is fixed at 10 Mbps.

FlexRay is a multi-master time-triggered protocol where each node is assigned a specific timeslot for transmission. An independent bus guardian on each node enforces these constraints even if the device malfunctions.

Each device node typically requires a "communications controller" (e.g., Freescale MFR4310, ~170 mW) and a "bus driver" (e.g., ON-Semi NCV7381, ~130 mW).<sup>10,11</sup> A few microcontrollers have built-in FlexRay interfaces, but these are extremely rare.

#### **Ethernet (100 Base-T)**

Ethernet (IEEE 802.3) is a ubiquitous standard in personal computer networking. Physical layer connectivity uses a star-topology with point-to-point links over a variety of media. 100 Base-T is the variant using 8-wire UTP or STP cable, operating at 100 Mbps.

At the center of this star, older Ethernet networks were connected by a hub that blindly repeats signals on all ports, with bus arbitration handled by collision detection. However, this method has poor congestion performance and has largely been superseded by intelligent central switches. In switch-based networks, each point-to-point link can transmit and receive simultaneously; congestion can still occur but there is no chance of collisions.

While the switch itself is a potential single-point-of-failure, Ethernet networks can have any number of switches. For example, critical systems can be segregated on a lower-speed subnet using a high-reliability rad-hard switch, connected to a separate high-speed switch for other payloads. In this model, communication between the two subnets is easy in the nominal case, but a failure of the high-speed subnet has no impact on the high-reliability subnet.

The use of a switch has the added benefit of isolating many failures. Since every Ethernet frame ends in a checksum, the switch will simply reject any malformed frames. While excess congestion due to so-called “babbling idiot” failures remain a risk, this can be mitigated with extensions such as TTEthernet.<sup>12</sup>

IEEE 802.3 requires that each device on the network include small signal transformers (a.k.a. “magnetics”) for safety and EMI/EMC mitigation. These transformers add nontrivial size, even when they are built into the 8P8C / RJ45 jack. In practice, however, applications with short cables (e.g., smallsats) may omit these in most cases, instead using simple capacitive coupling.<sup>13</sup>

Because every connection is made from an endpoint to a centralized switch, total power includes the transceivers at both ends of that link plus the switch itself. (A switch-based network with N endpoints has 2N transceivers, one each for the endpoint and the switch itself.) The NXP SJA1105 is a low-power automotive Ethernet switch with five ports, drawing about 100 mW total in RMII mode (i.e., 20 mW per port).<sup>14</sup> A typical RMII-to-100Base-T transceiver, such as the Texas Instruments DP83822, draws about 120 mW.<sup>15</sup> Since each end of the link requires a transceiver, total power per link is 260 mW.

### Ethernet (1000 Base-T)

1000 Base-T is the most common standard for gigabit Ethernet over UTP or STP cables. It uses cabling and connectors that are backwards-compatible with 100 Base-T. However, it cannot be used with a hub; a switch is always required. For backwards compatibility, most interface devices can automatically negotiate the use of 10/100/1000 Base-T.

Due to the higher speed, power requirements are somewhat higher. Compared to the 100 Base-T example above, power for the NXP SJA1105 switch increases to about 200 mW, or 40 mW per port.<sup>14</sup> A typical RGMII-to-1000Base-T transceiver, such as the MicroSemi VSC8541, draws about 460 mW.<sup>16</sup> As a result, total power per link is about 960 mW.

### Ethernet (RGMII)

Often, microcontrollers and FPGAs include a MAC controller but use a separate chip to provide the PHY interfaces for the long-distance connection. An Ethernet PHY chip communicates with the host via a “media independent interface” such as MII, GMII, RGMII, SGMII, or XAUI, depending on required data rate and available pin count. This allows interoperability with copper, fiber, and other transmission media.

However, most MAC-to-PHY interfaces are nearly symmetric. As a result, many MAC devices can be connected back-to-back, directly through the media-independent interface signals. So-called “MAC-to-MAC” configurations are commonly used within devices to save cost and power by eliminating redundant conversion to and from a longer-range physical interface.<sup>17</sup>

The two most relevant formats are RGMII and SGMII, both widely adopted de-facto industry standards for gigabit Ethernet MAC-to-PHY interfaces.

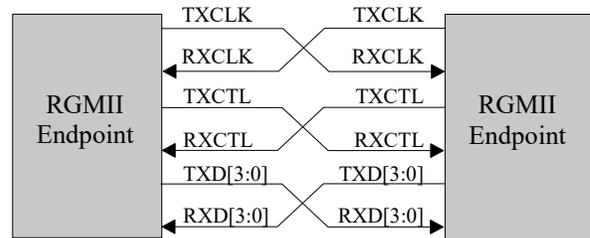


Figure 1: MAC-to-MAC RGMII

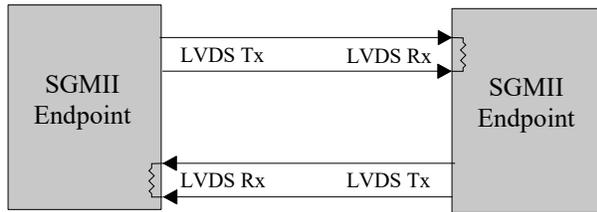
RGMII (Reduced pin-count Gigabit Media Independent Interface) uses 12 pins: 6 Tx and 6 Rx as shown in Figure 1. Data is transmitted at 250 Mbps/pin alongside a 125 MHz clock.<sup>18</sup> Board-layout guidelines from Texas Instruments recommend that RGMII traces be kept to a maximum length of 6 inches.<sup>19</sup> With care, such a limit is achievable for a cubesat backplane.

Power for a MAC-to-MAC RGMII link is very low. Many large microcontrollers support the interface natively, as does the NXP SJA1105. In such a network, marginal power increase is dominated by the switch itself, approximately 200 mW total.<sup>14</sup> Amortized over five ports, that’s just 40 mW per link, i.e., 1/20<sup>th</sup> the power of a 1000-Base-T Ethernet link.

### Ethernet (SGMII)

SGMII (Serialized Gigabit Media Independent Interface) is another de-facto standard that can be used for MAC-to-MAC interfaces.<sup>20</sup> A common example is the use of SFP-to-SFP cables in short network runs. Baseline

SGMII provides gigabit user rates, but faster variants allow user rates up to 2,500 Mbps.



**Figure 2: MAC-to-MAC SGMII**

SGMII uses just 4 pins: one differential pair for Tx and another for Rx as shown in Figure 2. The main advantages of SGMII compared to RGMII are a further reduction in pin count (i.e., 4 vs. 12), the availability of 2,500 Mbps transceivers, relaxed trace-length restrictions (up to several meters over twinax cables), and support for galvanic isolation. Each SGMII differential pair operates at 1,250 Mbaud due to the use of an additional 8b/10b encoding layer, so that the useful rate remains 1,000 Mbps.

Notably, at this line rate some FPGAs (including all Xilinx 7-series parts) can drive such signals directly from regular I/O pins, and do not need dedicated SERDES transceivers.<sup>22</sup>

Power for SGMII was estimated using the Microsemi VSC7418, an Ethernet switch with 11 SGMII ports. Typical power consumption is 1.6W, or about 145 mW per port.<sup>21</sup>

### ***I<sup>2</sup>C***

The Inter-Integrated-Circuit (I<sup>2</sup>C) bus is ubiquitous in microcontroller communications, usually used to control multiple peripherals using a single host interface. It uses a two-wire open-drain interface to convey clock and data. This wired-AND functionality limits top speed to 400 kHz, but also enables multi-master arbitration, clock-stretching, and other functionality.

Notably, I<sup>2</sup>C was selected as the low-throughput complement to IEEE-1394a for JPL's "X2000" initiative.<sup>1</sup> In addition to control and housekeeping functions, the I<sup>2</sup>C network offered a redundant, system-wide communications channel for use in diagnosing faults in the higher-speed bus, and then enabling reconfiguration to route around those faults.

Power overhead for this protocol is minimal; it is built into nearly every microcontroller and requires no power when idle. When transmitting, as much as 10 mW may be required to overcome strong pullup resistors.

### ***IEEE-1394a***

IEEE-1394a, popularly known under Apple's trademark "FireWire", is a consumer electronics standard that operates up to 400 Mbps. Physical layer is via point-to-point links with two differential pairs. Networking is accomplished by daisy-chaining, so each device typically needs two ports and must always be ready to relay messages to downstream nodes. The resulting peer-to-peer network may have up to 63 nodes.

Notably, IEEE-1394a was selected as the high-throughput bus for JPL's "X2000" initiative.<sup>1</sup> The bus includes many desirable properties, including multi-master support and scheduled timeslots to guarantee low-latency transmission of high priority data. The JPL network added redundancy by using three physical ports per device; by disabling or enabling specific ports they could adjust the network topology to reroute around damaged nodes without creating loops or requiring redundant systems.

However, future support for both IEEE-1394a and -1394b may be limited. In the consumer electronics domain, it has largely been supplanted by USB 2.0 and USB 3.0. Texas Instruments is the only identified COTS vendor still selling 1394a transceiver chips. Estimated power is around 200 mW per port.<sup>23</sup> Due to daisy-chaining, most devices will require two ports, for a total of 400 mW per node.

### ***LIN***

Local Interconnect Network (LIN) is a single-wire networking protocol developed for ultra-low-cost automotive applications. Though its maximum speed is only 19.2 kbps, the use of a single wire for power and data makes it attractive for sensors and actuators when data rates are of secondary importance to reduced wire-harness complexity and weight.

The network is single-master event-driven, so no bus arbitration is required. Power overhead is miniscule, with standalone slave devices often drawing less than 1 mW.<sup>24</sup>

### ***MIL-STD-1553***

MIL-STD-1553 is a multi-master serial data bus used in many aircraft. Data rate is fixed at 1 Mbps, using a multi-drop bus over a single differential pair, typically 78-ohm twinax. Though quite robust, 1553 is not compatible with cubesat-scale SWaP due to high signal voltages (typically 18-27 V<sub>pp</sub>) and the requirement for bulky isolation transformers.

Many interface controllers support connections to 2-4 separate bus pairs, for physical redundancy in case of a

wiring fault. There is also an optical variant of the standard, MIL-STD-1773, which can operate up to 20 Mbps.

### **RapidIO (Serial)**

RapidIO (formerly “Serial RapidIO” or “sRIO” until the parallel version was declared obsolete) is a networking standard used in high-performance computing. The protocol is designed to support low-latency communications, including flow-control and error-handling, with a minimum of required transceiver complexity. Networks use point-to-point links with switches to form a star topology.

Physically, each “lane” consists of two differential pairs, one in each direction, operating at 1.25/2.50/3.125 Gbaud (RapidIO 1.2), 6.25 Gbaud (RapidIO 2.0), or even 25 Gbaud (RapidIO 4.0). Multiple lanes can be combined in parallel to form a single logical “channel”. Alternate physical layers exist, including 10GBASE-KR and optical links.

FPGA IP-cores for RapidIO endpoints are available, including at least one open-source core. Incremental power is dominated by the SERDES, estimated around 130 mW (see also: Aurora/Reflex).<sup>3,4</sup> A typical switch, such as the IDT Tsi577, does not require an external transceiver and can shut down power to unused ports. This switch draws around 2.7W when all sixteen lanes are operating at full speed (i.e., 170 mW/port).<sup>25</sup> This gives a total estimated power of around 300 mW per active network node.

Notably, RapidIO was selected as the standard interconnect for AFRL’s Next Generation Spacecraft Interconnect Standard (NGSIS).<sup>26</sup>

### **RS-485**

RS-485 (a.k.a. TIA-485 or EIA-485) is a multi-drop, multi-master bus that uses a single differential pair, with a specific termination/biasing network. Maximum data rate depends on cable length; for a cubesat-scale bus, rates up to 10 Mbps are readily achievable.

The standard specifies the physical layer only; all further properties are application-specific. As such, there is no built-in arbitration of bus access, etc. A wide variety of higher-layer protocols are in common use for industrial control, such as Modbus, Profibus, DCC, and SSIP. However, all these networks assume a single centralized controller, where each peripheral is permitted to transmit only in response to a controller command.

True peer-to-peer networking using carrier-sense collision detection is theoretically possible (using methods similar to older-style Ethernet networks), but

such designs are not widely recommended or supported. At minimum, such systems tend to suffer unpredictable latency problems even when the network is lightly congested.

For a typical RS-485 transceiver (e.g., Maxim MAX3140), required power is 10 mW in receive mode, but transmissions with typical termination may require up to 125 mW.<sup>27</sup> Assuming 10% transmit duty cycle, this gives an average power no more than 25 mW.

### **SpaceWire**

SpaceWire is a routable protocol intended for robust space applications, derived from IEEE 1355 and widely adopted by the European Space Agency. As such, almost all available parts are radiation-hardened to some extent.

Individual links are point-to-point, using two LVDS pairs for each simplex connection (i.e., four pairs for a full-duplex connection). Baud rates vary from part to part. Many parts support a maximum of 200 Mbps, but a few are rated up to 400 Mbps.

Estimated power consumption is derived from the Atmel AT7910E 8-port router (200 Mbps max per port). At maximum activity, this chip draws up to 4 W; or 500 mW per port.<sup>28</sup> For lack of a better reference, an additional 50% is assumed for the endpoint PHY, giving a total estimate of 750 mW per link.

### **SPI**

The Serial Peripheral Interface (SPI) is a single-master serial protocol. It is ubiquitous in embedded systems. It may use four wires (clock, Tx, Rx, chip-select), or three wires (clock, shared Tx/Rx, chip-select). In most cases a single bus is shared between N devices, each with its own chip select, so a total of 3+N wires are required for the four-wire variant.

Maximum baud rates for SPI vary widely. As a representative example, the Vorago VA10820 can send/receive SPI at up to 12.5 Mbps if it is the bus master, but only 4.1 Mbps as a slave.<sup>29</sup>

Power required for SPI is minimal.

### **TTP/C**

TTP/C (“Time Triggered Protocol”) is a networking protocol for safety-critical control systems. It is a multi-master bus with assigned timeslots for all messages. In most cases, dual physical channels are used for redundancy, with the option to use both for increased speed under nominal conditions.

The protocol supports multiple physical layer options. The most common are RS-485 (up to 4 Mbps) and Ethernet/MII (up to 25 Mbps).

Node architecture consists of a host (the payload or application processor), a Communications Controller (implements media-agnostic TTP/C including guardian functions), and Controller Network Interface (connects the controller to the network physical layer). Companies such as TTEch sell chipsets for the latter two functions, with power consumption of around 300 mW.<sup>30</sup>

### **UART-TTL**

Universal Asynchronous Receiver-Transmitters (UARTs) refer to a broad class of serial communication formats, in which idle time is used to eliminate the need for high-precision clock recovery. For byte-oriented transfers, clock differences up to 5% have no impact on receiver performance. The TTL variant uses LVCMOS-level single-ended signaling, unlike RS-232, RS-422, RS-485, and other formats that use various differential voltage ranges.

Individual UART links may be point-to-point or on a shared linear bus using tri-state drivers. As a PHY-only standard, there is no built-in arbitration protocol.

Power overhead of UART-TTL interfaces is minimal.

### **USB 2.0**

USB is a ubiquitous standard in consumer electronics. USB 1.x operates at up to 11 Mbps (“Full-Speed”) using a four-pin connector (GND, +5V, D+, D-) for each point-to-point link. USB 2.x operates at up to 480 Mbps (“High-speed”) using the same connectors.<sup>31</sup>

Originally intended for use with PCs, a USB network consists of a single root (typically the host PC), a tree of hubs, and up to 127 peripheral devices. All transfers must be to or from the root; direct peripheral-to-peripheral transfers are not possible.

Per-node power is estimated at approximately 150 mW per device: 50 mW for the device PHY (e.g., Cypress CY7C68003) and additional 100 mW for the HUB.<sup>32,33</sup> (e.g., Texas Instruments TUSB4041), amortized over four devices.

Unfortunately, the single root of a USB network presents significant problems for use in smallsats. The root is a single-point-of-failure with no potential for redundancy. Furthermore, robust low-speed devices cannot act as the host because all traffic must route through this node.

### **USB 3.x**

USB 3.0, 3.1, and 3.2 are expansions of the USB standard that allow much higher transfer rates. Each uses a 9-pin connector that adds two “SuperSpeed” differential pairs and an additional ground pin.

USB 3.0 (“SuperSpeed”) operates at up to 5.0 Gbps. USB 3.1 (“SuperSpeed+”) operates at up to 10.0 Gbps. USB 3.2 adds a flat and reversible “Type C” connector that allows additional current draw (up to 3000 mA!) and allows multi-lane ganging for transfers up to 20.0 Gbps.<sup>31</sup>

Per-node power is estimated at approximately 700 mW per device: 400 mW for the device PHY (e.g., Texas Instruments TUSB1310) and an additional 300 mW for the HUB (e.g., Texas Instruments TUSB8041), amortized over four devices.<sup>34,35</sup>

Note that hub power scales well with device class; a USB 3.0 capable hub operating in USB 2.0 mode reduces power substantially, almost to the level of a USB 2.0 hub.

Concerns about single-point failure are largely the same as those discussed under USB 2.0.

### **MIXED-MEDIA ETHERNET**

As previously discussed, the authors’ initial assumption was that a two-tiered network would be necessary to meet all design objectives. Of surveyed options for the high-speed tier, only two met the MMBD and throughput requirements: Ethernet and RapidIO. Of these, Ethernet was selected due to its relative simplicity, the reduced power consumption of RGMII or SGMII interfaces, and its ubiquitous adoption in both consumer and commercial electronics.

However, RGMII and SGMII could not meet the requirement for compatibility with small microcontrollers, nor could a network with dozens of small devices meet the 1W power requirement.

In the original two-tier solution, all payloads must connect to the separate low-speed network. Specific payloads also connect to the high-speed Ethernet network, which is then used only to carry bulk data.

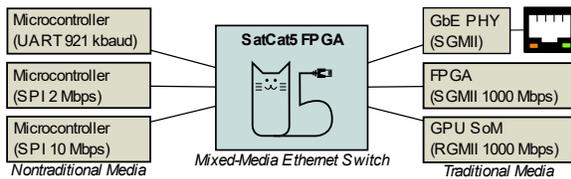
The leading candidate for the low-speed network was CAN, chosen for its simplicity, flexible topology, robust signaling, and the wide availability of automotive-grade parts. However, several factors made this two-tier design undesirable. First, the low-speed tier would have significantly degraded access to the main radio link, since downlink rate for many cubesat radios already exceeds CAN’s maximum user rate. Second, the need

for any-to-any connectivity meant that high-speed payloads would have to include separate interfaces for both networks. This duplication meant increased size, weight, power, and complexity.

The superior solution was to extend the Ethernet network to service low-speed devices while avoiding the complexities of RGMII and SGMII. This was accomplished by carrying Ethernet frames over a selection of simpler interfaces.

The IEEE 802.3 standard already supports dozens of physical-layer options. The universal underlying factor is the Ethernet frame: a header with 14-bytes (octets) and a 4-byte CRC encapsulating an arbitrary packet of user data.<sup>36</sup> The only barrier to carrying Ethernet frames over a simpler interface is the creation of a suitable network switch.

### SATCAT5



**Figure 3: Mixed-media Ethernet switch**

*SatCat5* is our implementation of the mixed-media Ethernet switch concept. The core element is a field programmable gate array (FPGA) that has been configured to act as an Ethernet switch, where each port can be set to use a different physical interface. A typical network with six ports is shown in Figure 3.

*SatCat5* has been released as free and open-source software under the LGPLv3 license:

<https://github.com/the-aerospace-corporation/satcat5>

### Switch Architecture

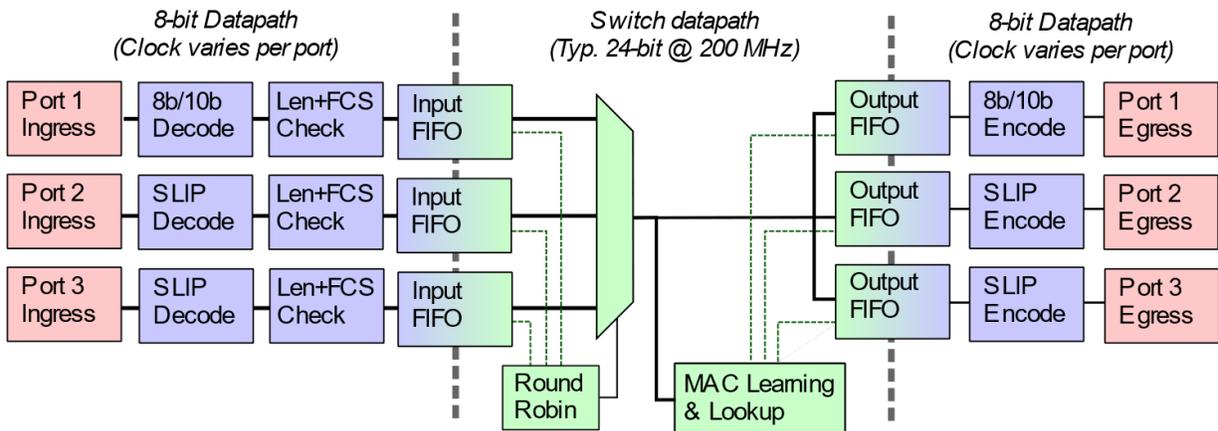
The core datapath is a typical output-queued, shared-medium switch.<sup>37,38</sup> as shown in Figure 4. This design pattern is relatively simple, is immune to head-of-line blocking, maintains first-in/first-out ordering, and maximizes throughput (i.e., only drops packets if an output is oversubscribed).

Parameters such as port count, buffer size, and the width of the shared pipeline are all adjustable to match system requirements and available FPGA resources. Aggregate throughput across all ports is limited by the shared-medium segment, with a maximum practical total of approximately 9.6 Gbps (i.e., 48 bits at 200 MHz). This is small compared to an enterprise-grade switch but more than adequate to avoid bottlenecks in most smallsat networks. (i.e., Enough throughput for all traffic from eight gigabit ports and a few dozen low-speed ports.)

At present, the switch follows round-robin prioritization of incoming packets, queuing at each output on a first-come, first-served basis. Packets that overflow the fixed-size output queue are simply dropped. We plan to add improved packet scheduling in future updates, including VLAN- or QoS-based packet prioritization. Simple broadcast packets are supported, but IGMP is not.

MAC-address learning is handled automatically, by inspecting the source address for each Ethernet frame as it traverses the switch. Because this function is resource-intensive, multiple algorithms are provided to better match specific use cases. The preferred general-purpose implementation is a LUTRAM-based content addressable memory (CAM) that readily scales to networks with 64 unique MAC addresses.<sup>39</sup>

Packet validation is handled as each packet is received. A small FIFO on each input, just big enough for a single maximum-size frame, handles clock-domain conversion and allows easy deletion of partial or invalid frames.



**Figure 4: SatCat5 Switch Datapath**

This process checks the length and checksum per IEEE 802.3 recommendations, with optional support for jumbo frames if enough FIFO memory is available.

The one exception from standard IEEE 802.3 validation is the minimum frame size. SPI and UART interfaces have no need for ambles, minimum inter-packet gaps, etc., and these are omitted. Furthermore, padding small frames to the minimum of 64 bytes wastes valuable time on low-bitrate interfaces, so this requirement is waived as well. Several methods are provided to zero-pad runt frames and recalculate the frame check sequence before transmission on fully-compliant RMII, RGMII, and SGMII interfaces.

### Supported Interfaces

Low-speed interface options include:

- SPI (up to ~10 Mbps)
- UART (fixed rate, default 921,600 baud)
- Auto-sensing SPI/UART (Port-type determined at runtime)

In all three cases, byte streams are SLIP-encoded<sup>40</sup> to mark frame boundaries. All three options use four pins.

Since many low-speed ports will be connected to single-tasking microcontrollers that are not always ready to accept data, flow control is essential. For SPI ports, flow control is handled by making the remote node the master; it should only drive the SPI clock when it is ready to exchange data. For UART ports, flow control uses a discrete CTSb signal to indicate the endpoint is ready to accept data.

High-speed interface options include RMII (up to 100 Mbps), RGMII (fixed at 1,000 Mbps), and SGMII (fixed at 1,000 Mbps). Any of these can be used directly in MAC-to-MAC mode, or in conjunction with an off-the-shelf Ethernet PHY ASIC to connect to regular 10/100/1000 Base-T networks.

Core logic is written to be platform-agnostic. Apart from SGMII, all required I/O is implemented using extensible wrappers that can be ported to primitives on multiple platforms. Supported FPGA platforms and toolchains include the Xilinx 7-series (Vivado), Microsemi PolarFire (LiberO), and Lattice iCE40 (iCEcube). Additional platforms may be added in a future update.

### Prototypes

The first *SatCat5* prototype (Figure 5, Figure 6) was a pathfinder to allow initial development for most of the supported interfaces (SPI, UART, RGMII, and SGMII) and to evaluate the performance of an alternate design in which most of the gigabit Ethernet switching would be

provided by an off-the-shelf ASIC. (Namely, the NXP Semiconductor SJA1105.) The design consisted of a custom I/O card coupled to one of several off-the-shelf FPGA development boards (e.g., Xilinx AC701 or Microsemi Polarfire Splash). The I/O card contains multiple Ethernet PHYs, several PMOD ports, the switch ASIC, and an FMC port. The Ethernet-over-PMOD pinouts are chosen to be compatible with off-the-shelf USB-UART and USB-SPI adapters.

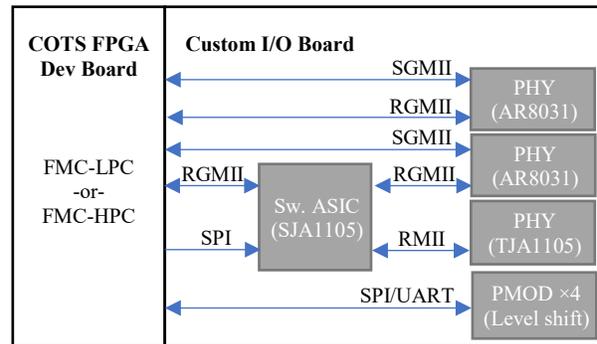


Figure 5: Block diagram of 1<sup>st</sup>-generation prototype

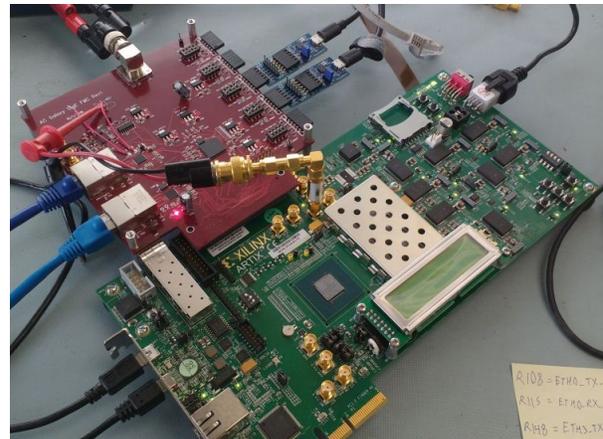


Figure 6: First-generation prototype with AC701

The prototype was successful and allowed the first direct measurements of FPGA power consumption under different conditions. Total power draw for the AC701's Artix7-200T FPGA ranged from 150 to 680 mW, depending on the number of active gigabit ports. (Not including power-supply conversion overhead and unused accessories on the AC701 board.) Of this total, nearly 144 mW is due to the static power of the large FPGA. The simple optimization of moving to a smaller FPGA, such as the Artix7-75T—still adequate for five gigabit ports and twenty low-speed ports—would reduce the static power by an estimated 45 mW.

The second-generation prototype (Figure 7, Figure 8), was intended to showcase efforts to miniaturize *SatCat5*. The goal was a self-contained system that could provide data and power to multiple payloads, in a 9×9 cm form-

factor that could fit in a small cubesat. In addition to the *SatCat5* Ethernet switch FPGA, the second prototype included a 60W 5VDC switched-mode power supply (SMPS) for connected payloads, full voltage and current telemetry for each payload port, eight low-speed ports, four SGMII ports, and a 1000 Base-T Ethernet PHY.

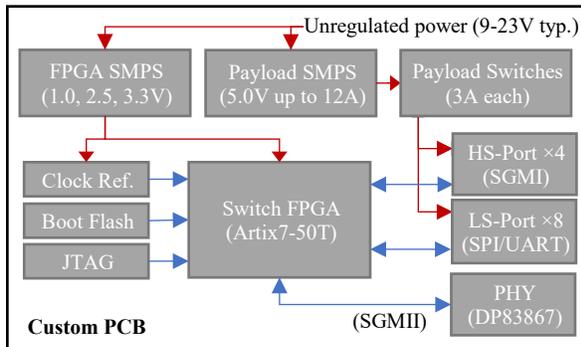


Figure 7: Block diagram of 2<sup>nd</sup>-generation prototype



Figure 8: Second-generation standalone prototype

PCB design files for the first-generation prototype are available as part of the initial *SatCat5* release. The authors hope to release design files for the second-generation prototype in a future update.

### Off-the-shelf Adapters

Gaining widespread adoption is a significant challenge for any new would-be standard. Requiring custom hardware is a significant barrier to entry for any prospective adopter who wishes to try out a given design. To reduce this barrier for *SatCat5*, we've included two different off-the-shelf solutions in the open-source release, both targeting hobbyist-friendly hardware.

The first solution is a *SatCat5* example design targeting the Digilent Arty A7, a low-cost FPGA development board. This board has limited high-speed I/O, but it does have a 10/100 Base-T Ethernet PHY (using RMI) and four PMOD ports. The example design hosts a *SatCat5* Ethernet switch that allows each of the PMOD ports to act as an Ethernet port in auto-sensing SPI/UART mode. The 10/100 Base-T PHY allows the same switch to be connected to regular off-the-shelf Ethernet hardware.

The second solution, called “*PiWire*”, is C++ software that can be hosted on a Raspberry Pi 4, a low-cost single board computer (SBC). The software puts the SBC’s 10/100/1000 Base-T Ethernet port in promiscuous mode, then activates the SPI or UART interface on the SBC’s GPIO pins. Ethernet frames received on either interface are forwarded verbatim to the other, turning the SBC into a low-cost converter connecting 10/100/1000 Base-T to a *SatCat5* SPI or UART port. The main limitation to *PiWire* is that the SPI port cannot be operated as an SPI slave, due to hardware and software driver limitations.

### SLINGSHOT MISSION

The *Slingshot 1* mission, slated for launch in late 2021, will be the first Aerospace Corporation cubesat to make use of *SatCat5*. The primary mission goal is to “build the modular future of space”, with multiple payloads conforming to a specified interface for power, data, temperature control, and other critical functions. *SatCat5* provides the network for exchanging command, control, telemetry, and mission data between different payloads and between each payload and the host vehicle.

With more than a dozen potential *Slingshot 1* payloads vying for limited access to engineering models and other costly test assets, it was important to provide a low-cost alternative with an equivalent host interface. Our solution is the *Slingshot* payload development kit (PDK). Figure 9 shows a PDK connected to a Raspberry Pi and a host PC using an off-the-shelf Ethernet switch. The payload under test is then connected using one of the two empty connectors shown at lower right. Of these, the left connector (green) provides core functions including multiple power rails, temperature monitoring, discrete GPIO, and Ethernet over SPI/UART. The right connector (silver) adds SGMII and additional power for gigabit-capable payloads.

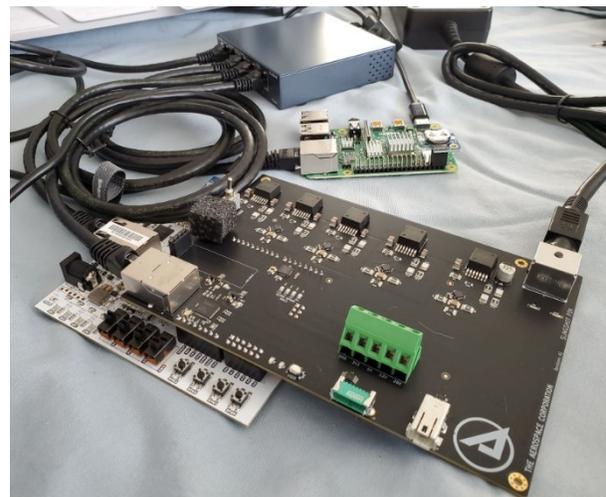


Figure 9: *Slingshot* payload development kit

The PDK uses a custom I/O card with an off-the-shelf Digilent Arty A7 FPGA board to host *SatCat5*. Complete bill-of-materials cost is US\$650. Multiple PDK units can be attached to the same off-the-shelf network switch to support multi-payload tests. Additional software running on the attached Raspberry Pi is used to emulate various host vehicle functions and provide a complete test-like-you-fly environment.

## EXAMPLE NETWORK

As a platform, *SatCat5* acts solely as a mixed-media Ethernet switch. It provides a common ground for sending messages between any two network endpoints, but it is completely agnostic to the contents of those messages. In the OSI model, this corresponds to Layers 1 and 2, i.e., the physical layer and the data link layer.

Higher layers must be agreed upon for the complete system to function, but we intentionally leave this specification separate from *SatCat5*. For now, the task falls to system designers of individual smallsats. Nevertheless, we wish to provide an example that shows how *SatCat5* can be used effectively.

In this example design, a cubesat vehicle has four devices linked together by a *SatCat5* mixed-media Ethernet network:

- Attitude actuators and sensors (UART)
- Attitude control system (SPI)
- Video camera (SGMII)
- Laser downlink (SGMII)

Each attitude sensor module has a simple microcontroller linked to several analog sensors—sun sensors, Earth limb sensors, rate gyros, etc. The sensor module(s) take regularly scheduled readings, then send that data as a series of time-tagged broadcast packets. Taking inspiration from CAN, broadcast packets are preferred for most low-rate telemetry because it's difficult to predict which subsystems might need which data in every possible operational mode. All nodes receive a steady stream of such telemetry, and simply filter out the message(s) of current interest to their needs. The broadcast model increases chatter, but makes device discovery and message distribution considerably simpler than under an explicit subscription model.

Each telemetry packet contains a CBOR (IETF RFC 7049) key-value dictionary. Keys are unique numeric constants, agreed upon by all payloads, that denote the type and scale of the associated value. For example, key #42 might indicate the +X rate gyro, with units of arcseconds per second. The associated value may be a simple numeric value, such as +5.7, or a small array of numbers, etc. Each payload quickly scans over the

available field, parses fields that may be of interest, and skips the rest. The messages use raw Ethernet frames, with an ad-hoc EtherType field indicating that it is CBOR telemetry.

The attitude control system receives this telemetry stream, picks out relevant attitude sensor data, and uses it to update its state-estimation filters. It then broadcasts its own telemetry, such as the estimated spacecraft attitude quaternion, using the same CBOR message format. It also sends commands to specific reaction wheels and other actuators required to maintain the desired attitude.

When the attitude control system determines that the vehicle is ready to initiate an optical communications session, it sends commands to activate the video camera and laser downlink modules.

The video camera has been previously commanded to send an RTP-UDP video stream (IETF RFC 3550) to a specific IP address. To open this stream, it sends an ARP request (IETF RFC 826) to find out the local MAC address associated with that IP address.

The laser downlink module, acting as an IPv4 router (IETF RFC 1027), sees that the ARP request corresponds to the optical ground terminal at the other end of its link. It responds to the ARP with its own MAC address, and the video camera begins streaming RTP-UDP packets.

The laser downlink module receives the UDP datagrams. Without needing to understand the RTP protocol, it sees that the UDP datagrams are addressed to the ground terminal and relays them verbatim. The PC attached to the ground terminal receives the stream and has software to parse the RTP stream and display the live video.

## CONCLUSION

After a survey of more than twenty widely adopted standards, we selected MAC-to-MAC Ethernet as the best option for distributing high-speed data on our near-future smallsat missions. To allow lower-speed devices to connect to the same network, we send Ethernet frames over simpler, microcontroller-friendly protocols such as SPI and UART. *SatCat5* is an open-source FPGA platform that ties the entire network together using a mixed-media Ethernet switch. This switch requires a power overhead of 0.2-0.7W, depending on network activity. However, we feel this is a small price to pay for flexible and robust communications and hope that other smallsat developers can adopt this tool.

## ACKNOWLEDGEMENTS

The authors would like to thank Jovanna Bunch, Stephen Snow, Stanislav Stanev, Louis Stromeyer, and Emily Wesel for their contributions to *SatCat5*.

## REFERENCES

1. Chau, S.N., L. Alkalai, and A.T. Tai, "The Design of a Fault-Tolerant COTS-Based Bus Architecture," NASA Technical Report #20000053469, April 1999.
2. Welle, R., S. Janson, D. Rowen, and T. Rose, "CubeSat-Scale High-Speed Laser Downlinks," Proceedings of the 13th Reinventing Space Conference, 2018.
3. "Xilinx Power Estimator (Spartan-7, Artix-7, Kintex-7, Virtex-7, Zynq-7000)," Xilinx Inc., Aug 2017.
4. "Early Power Estimator for Intel Stratix 10 Devices," Intel Corporation, Version 19.4, Downloaded April 2020.
5. Afaneh, M., "Bluetooth 5 speed: How to achieve maximum throughput for your BLE application," NovelBits.io blog post, Sep 2017. <https://www.novelbits.io/bluetooth-5-speed-maximum-throughput/>
6. "ISP1507 High Performance Bluetooth 5 Ready, NFC&ANT Low Energy Module," Insight SiP, DS1507 Rev 8, Jan 2018.
7. "nRF52810 Product Specification," Nordic Semiconductor ASA, 4430\_161 v1.1, Nov 2017.
8. "Road vehicles – Controller area network (CAN) – Part 1: Data link layer and physical signaling," ISO Committee on Data Communication, ISO 11898-1:2015, Dec 2015.
9. "MCP25625 CAN Controller with Integrated Transceiver," Microchip Technology Inc., DS20005282B Rev B, Jan 2017.
10. "MFR4310 Reference Manual," Freescale Semiconductor, MFR4310RM Rev 2, Mar 2008.
11. "NCV7381 FlexRay Bus Driver," ON Semiconductor, NCV7381/D Rev 3, Jul 2016.
12. Daniel, O. and O. Roman, "Fault injection framework for assessing fault containment of tternet against babbling idiot failures," IEEE/ACM 26th International Symposium on Quality of Service. 2018.
13. "DP83848 PHYTER Transformerless Ethernet Operation", Texas Instruments Inc., Appnote AN-1519, Oct 2006.
14. "SJA1105 5-port automotive Ethernet switch", NXP Semiconductors product data sheet, Rev 1, Nov 2016.
15. "DP83822 Robust, low power 10/100 Mbps ethernet physical layer transceiver," Texas Instruments Inc., SNLS505E, August 2016.
16. "VSC8541-01 Datasheet, Single Port Gigabit Ethernet Copper PHY with GMII/RGMII/MII/RMII Interfaces," Microsemi Corporation, VMDS-10496, Feb 2017.
17. "Ethernet Switch MAC-to-MAC Connections," Zarlink Semiconductor Application Note ZLAN-30, June 2006.
18. "Reduced Gigabit Media Independent Interface", Broadcom, Hewlett-Packard, and Marvell Technology Group. Version 2.0, April 2002.
19. "DP83867IR/CR Robust, High Immunity 10/100/1000 Ethernet Physical Layer Transceiver," Texas Instruments Inc., SNLS484F, Dec 2019.
20. Chu, Y., "Serial-GMII Specification", Cisco Systems ENG-46158, Revision 1.8, Nov 2005.
21. "VSC7148 11-Port Layer-2 SGMII Gigabit Ethernet Switch with MPLS/MPLS-TP Support and VeriTime", Microsemi Corporation, VPPD-02951, Apr 2016.
22. Defosse, M., "LVDS 4x Asynchronous Oversampling Using 7 Series FPGAs and Zynq-7000 AP SoCs," Xilinx Appnote XAPP523, May 2017.
23. "IEEE 1394a-2000 Consumer Electronics Solution, Data Sheet Extract", Texas Instruments Inc., SLLA275C Rev 1.2, Feb 2010.
24. "UJA1023 LIN-I/O slave," NXP Semiconductors, Rev 5, Aug 2010.
25. "IDT Tsi577 Serial RapidIO Switch, Hardware Manual," Integrated Device Technology Inc., May 2012.
26. Collier, C.P. and J. Marshall, "Next Generation Space Interconnect Standard (NGSIS): A modular open standards approach for high performance interconnects for space," 2015 IEEE Aerospace Conference. 2015.
27. "SPI/MICROWIRE-Compatible UART with Integrated True Fail-Safe RS-485/RS-422 Transceivers," Maxim Integrated Products, 19-1453 Rev 1, Sep 2010.
28. "AT7910E SpW-10X SpaceWire Router Datsheet," Atmel Corporation, 7796G-AERO-02/13, Feb 2013.

29. "VA10820 Datasheet," Vorago Technologies, VA10820 Rev 2.3, Mar 2018.
30. "AS8202B TTP-CSNF Communication Controller," ams AG, AS8202B Rev 1.0, Mar 2013.
31. "Universal Serial Bus 3.2 Specification," USB 3.0 Promoter Group, Sep 2017.
32. "C7C68003 MoBL-USB TX2UL USB 2.0 ULPI Transceiver," Cypress Semiconductor Corporation, 001-15775 Rev N, Jan 2015.
33. "TUSB4041I-Q1 Four-Port USB 2.0 Hub," Texas Instruments Inc., SLLSEK4B, Jul 2015.
34. "TUSB1310 USB 3.0 Transceiver Data Manual," Texas Instruments Inc., SLLSE16E, May 2011.
35. "TUSB8041 Four-Port USB 3.0 Hub," Texas Instruments Inc., SLLSEE4D, Jun 2014.
36. "IEEE Standard for Ethernet," IEEE Computer Society, IEEE Std 802.3-2015, Feb 2015.
37. Demers, A., S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," ACM SIGCOMM Computer Communication Review 19.4, 1989.
38. Chuang S.T., A. Goel, N. McKeown, and B. Prabhakar, "Matching Output Queueing with a Combined Input/Output-Queued Switch," IEEE Journal On Selected Areas In Communications, Vol. 17.6, June 1999
39. Jiang, W., "Scalable ternary content addressable memory implementation using FPGAs," IEEE Architectures for Networking and Communications Systems, 2013.
40. Romkey, J., "A Nonstandard for Transmission of IP Datagrams Over Serial Lines: SLIP", IETF Network Working Group RFC 1055, June 1988.