Hardware Assets

Military-Specific I/O: Going Through Channels

Fabrics and Publish-Subscribe Schemes: A Net-Centric Blend

Traditional systems architectures have run out of stream. To meet net-centric warfare demands, switch fabrics and publish-subscribe middleware are required.

Brett Murphy, VP, Distributed Systems,
Real-Time Innovations
Emmanuel Eriksson, Director, Strategic Alliances
Dy 4 Systems

Net-centric warfare and the larger concept of net-centricity enabled by the Global Information Grid (GIG) are the driving paradigm in today's defense systems development. The net-centric architecture flattens information or data distribution patterns, making data sources directly available to any authorized node on the network that wishes to use the data. Applications, once centralized on high-performance servers, can now be "dis-aggregated"—meaning applications will span multiple, networked computers. Dis-aggregated or distributed systems are much easier to scale, make fault-tolerant and upgrade. This dis-aggregation calls for new application infrastructures.

Publish-subscribe communications is a key enabler of the new distributed architecture. Data sources publish their data to the network; data users subscribe to the data to receive real-time updates. The DoD feels that the net-centric systems philosophy will provide the real-time data distribution needed to create superior situational awareness for their warfighters. This net-centric transformation is revolutionizing embedded and real-time system design as well.

To dis-aggregate or distribute real-time embedded systems, new technologies are needed. On the hardware side, switched-fabric networks serve as high-performance communications conduits. Feeding that need, the new Data Distribution Service for Real-Time Systems (DDS) publish-subscribe standard recently adopted by the OMG, maps very naturally to the topologies and capabilities of switched fabrics.

Yesterday's Architectures Fall Short

The DoD and other customers are demanding increasing capability, supportability and availability (Table 1). Yesterday's single-CPU, stove-piped system architectures are bending and breaking under the load. Today's architectures are typically based on a multi-CPU, VME-backplane solution with hard-wired input/output interfaces to whatever sensors and effectors the system needs to do its job.

The capability, supportability and availability of the larger platform are not really a consideration for the subsystem developer. Interfaces between the systems have to be defined early by the lead system integrator. Changes during development, let alone after deployment, are expensive. Therefore the interfaces are typically kept to a minimum. Each system has to rely on its own stove-piped solutions to meet its needs.

An Ideal System Platform

In an ideal solution the multi-CPU, backplane shared-memory architecture can scale up infinitely. The backplane would have zero-latency and infinite bandwidth. The quality-of-service (QoS) would be perfectly matched to the latency and bandwidth requirements of each data stream. Each application would be a separate operating system process, and communication with other processes would be through a common, well-specified messaging interface.

Moreover, in an ideal solution software components would be dynamically distributed anywhere in the system to the location where its execution is optimized by the underlying hardware. Adding new functionality, system upgrades and fault-tolerance would be straightforward with an ideal architecture like this.

Good Step Forward

While not ideal, publish-subscribe and switched fabrics get much closer to the ideal. Publish-subscribe excels at real-time data distribution. Publish-subscribe is characterized by a set of data producers and data consumers. Whereas client-server has a request-reply model, publish-subscribe is more a "push" model. That is, after the publishers and subscribers have identified themselves on the network, the data is pushed onto the network by the publishers. Subscribers can then pull the data off the network anonymously—no requests or polling are required.

The OMG—which manages the CORBA standard—recognized the need for publish-subscribe communications. In June 2003, the OMG adopted the new Data

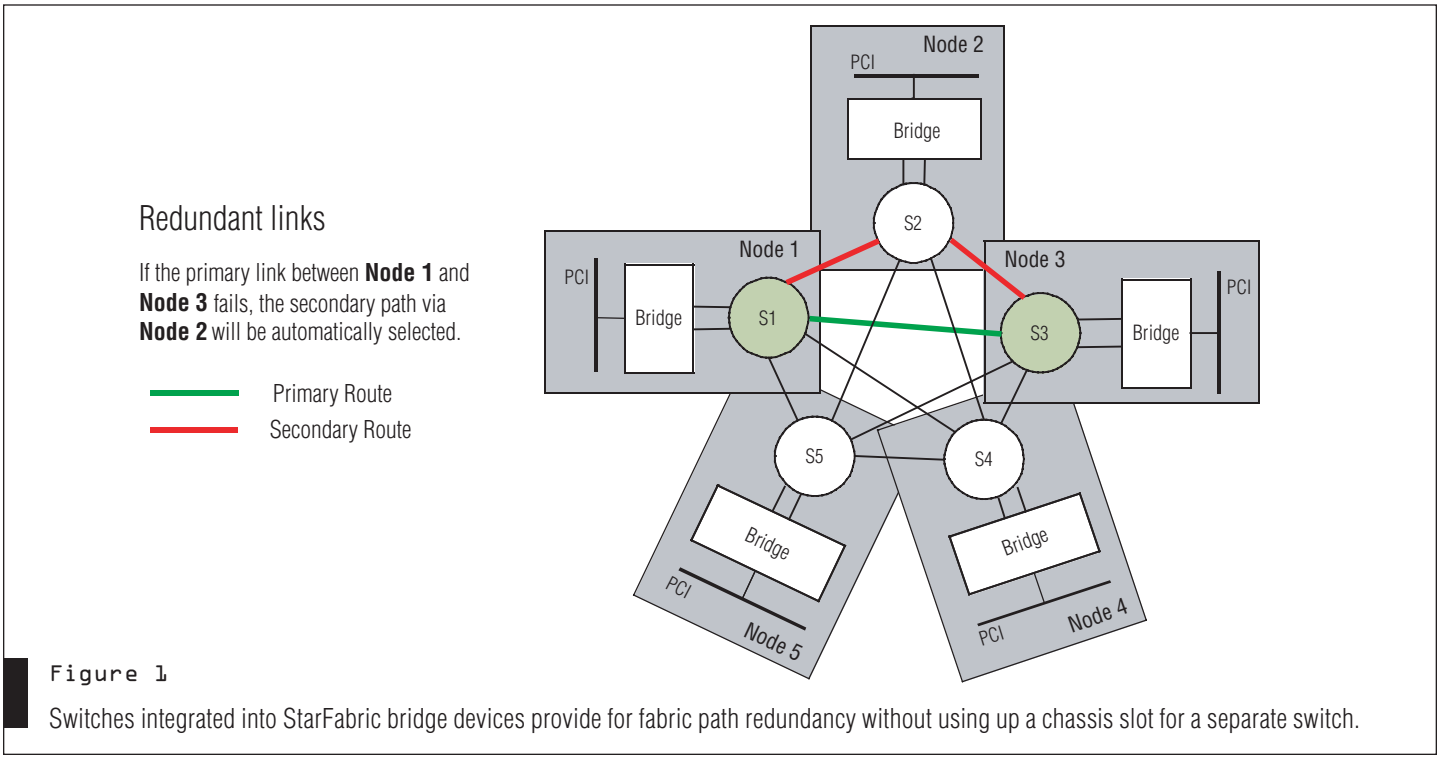


Figure 1

Switches integrated into StarFabric bridge devices provide for fabric path redundancy without using up a chassis slot for a separate switch.

Distribution Service for Real-Time Systems (DDS) standard and issued the specification to the public this year. Now there is a publish-subscribe standard for developers to use tailored specifically for real-time distributed systems.

Designers of complex, distributed systems stand to gain from the emergence of switched fabrics. Table 1 shows some of the common requirements that these designers are faced with and how those requirements are typically met with existing bus backplane technologies. The complexity of software needed to implement current solutions to these common requirements increases system development and post-deployment maintenance costs. In some cases, the required complexity results in projects that fail to complete development.

Switched fabrics such as StarFabric, PCI Express Advanced Switching (AS) and Serial RapidIO give new freedoms to designers to implement their systems. In the case of StarFabric, the solutions are available today. For others, solutions are just around the corner.

Real-Time Distributed Architecture

Together, publish-subscribe middleware and switched-fabric interconnects promise to greatly simplify system integrators' need to add capability, supportability and availability to their systems.

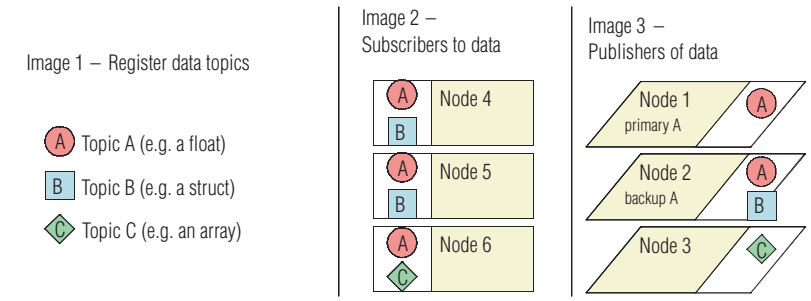
Scalability: Buses scale poorly—they are restricted by physical size and bandwidth. Switched fabrics on the other hand are highly scalable both in the number of nodes and in the bandwidth between nodes without the determinism and reliability constraints. Meanwhile, systems designed using publish-subscribe protocols are naturally scalable. With anonymous messaging, designers can change the number of subscribers to published data without affecting the publishing application code by simply duplicating the subscribing code on the added nodes.

Multicasting: Multicasting capabilities are used by publish-subscribe to efficiently publish data to any node that may potentially be subscribed to the data. Unlike IP, which relies on the stack to perform multicast function, switched fabrics implement multicasting in the switches. The result is that protocol stack overhead is minimized.

Quality of Service: Quality of Service (QoS) features are essentially lacking in Ethernet protocols. Switched fabrics, however, offer rich QoS features that help designers develop reliable, hard real-time systems. For example, the credit-based flow control mechanisms used by StarFabric and PCI Express AS permit bandwidth-reserved isochronous transactions across the fabric. Just as switched fabrics offer deterministic latencies at the transport level, DDS publish-subscribe middleware makes determinism at application level possible. For example, the DDS specification allows application developers to specify a Latency_Budget QoS policy. The middleware can use this Latency_Budget policy to better manage how it aggregates data for sending from multiple applications running on one node to multiple applications on another node.

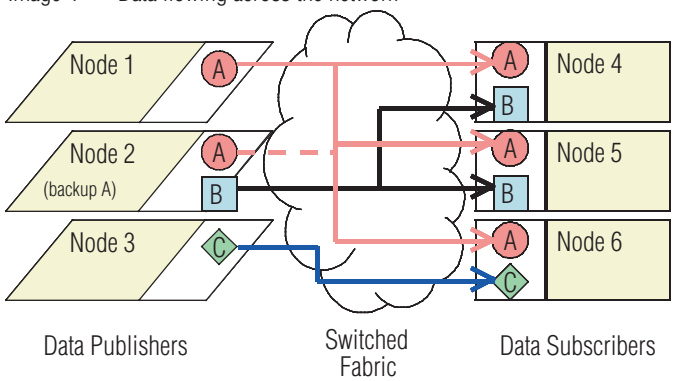
Hot Plug / Hot Swap: The ability to replace processor blades in a powered and running system is becoming a common requirement. Switched fabrics specifications provide for physical layer hot plug capability. DDS publish-subscribe also provides "virtual" hot plug capability at the application level that complements switched fabrics' support. Because publish-subscribe messaging is anonymous, the unannounced removal of data-reading nodes from the network will not cause the errors that would occur under a connection-based client-server model.

Error Management: Switched fabrics provide rich error management features designed to support high availability requirements. For instance, failures in PCI Express AS fabric paths are reported to



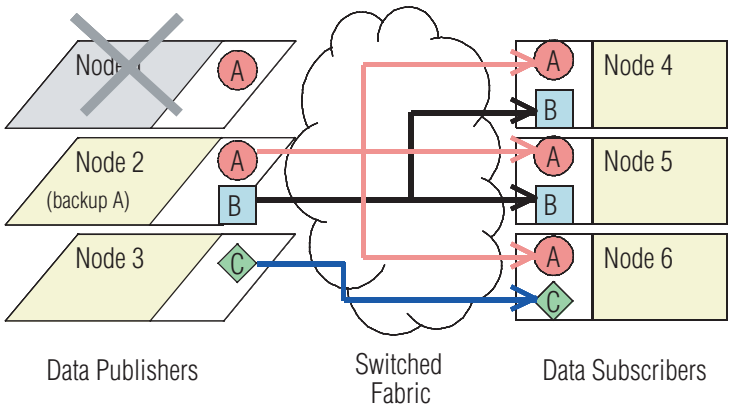
Topics identify data objects that will be sent and received.
Subscribers register Topics with the middleware they wish to receive.
Publishers register Topics with the middleware they will send or publish.
The order does not matter.

Image 4 – Data flowing across the network



Notice that Subscribers to Topic A do not receive data from the backup publisher on Node 2

Image 5 – Automatic failover from primary to backup



Example: Application on Node 1 crashes.
Pub-Sub middleware automatically switches to Topic A publisher on Node 2.
The subscribers get uninterrupted data.

Figure 2

Publish-subscribe involves direct, anonymous communication. Publishers send data directly to subscribers efficiently. Since subscribers receive data based on its name (topic) only and not a set address, you can have back-up publishers to effect robust failover scenarios.

	The Need	Current Parallel Bus-Based Solutions	The Problems
Capability	- Increasing functional requirements for new systems - Increasing requirements for deployed systems	- Various parallel bus interconnect schemes. - Serial interconnects such as Ethernet.	- Physical separation of cards is limited to usually less than 3 feet. - Limited bandwidth, high protocol overhead, non-deterministic performance unless using specialized protocols.
Supportability	- To reduce system life-cycle costs - To ease system upgrades	- Design in spare slots for future processors.	- Requires re-architecture of software to distribute functions to new processor cards.
Availability	- Highly available or fault-tolerant systems	- Redundant buses, cards or entirely redundant systems.	- Complex failover logistics and expensive hardware duplication. - If solved by having a redundant "standby" system, in addition to expense, the solution consumes physical space.

Table 1

a Fabric Manager (FM) node that identifies the failed paths and reroutes traffic to avoid the failure. At the application level, DDS-compliant middleware users can set a Deadline QoS policy on their subscribers. If the publisher doesn't publish a new update to the subscriber within the specified Deadline time duration, the subscribing application is notified that new data was not available.

Redundancy: Support for redundant fabric paths is a key feature of switched fabrics. For example, StarFabric's support for a distributed switch topology (Figure 1) results in each node having multiple, redundant paths to other nodes in the fabric. Combined with the error management features, this gives the designer simple-to-implement physical-interconnect redundancy. At the application level, DDS provides for redundant publishers. Redundant applications can be created that publish the exact same data onto the fabric, but with different "strengths". Subscribers to the data topic will receive data from the higher "strength" publisher (the higher strength publication "masks" the lower strength publication). As shown in Figure 2, if the higher strength publisher fails or is removed from the network, the middleware automatically switches the subscribers to the lower strength or back-up publisher without skipping a beat. This provides for "hot failover" redundancy.

Supporting Open Architectures

A major goal of the open approach to computing chosen for open architectures is to enable the development of applications that are portable across multiple brands and generations of COTS computing products. Standards are a cornerstone of the open systems approach.

Switched-fabrics and distribution middleware are COTS, standards-based solutions. COTS, standards-based solutions are key elements of the emerging tactical defense architectures like Navy OA, SOSCOE and others. In fact, the Navy Open Architecture Computing Environment calls out specific publish-subscribe middleware implementations like RTI's NDDS and switched-fabric standards like InfiniBand and Gigabit Ethernet for system developers to meet Navy OA Category 3 compliance.

BAE Systems is using publish-subscribe to do data distribution over StarFabric for the electronic warfare system on the F-35 Joint Strike Fighter. BAE Systems' application is inherently data-centric; they needed many-to-many data distribution with low latency. They selected RTI's NDDS data distribution middleware and Dy 4's StarLink switch fabric PMC and Champ-AV II DSP hardware. NDDS's publish-subscribe communications leverage the capabilities of StarLink's switch fabric interconnect to provide many-to-many data distribution, low-latency communications and seamless failover or switchover between nodes.

These new technologies are finding their way into the emerging tactical architectures like the Navy OA. They provide the basis for new, "net-centric" or distributed systems that are much easier to scale, make fault-tolerant and upgrade. These high-performance, distributed architectures promise to greatly simplify efforts to add better capability, supportability and availability to new defense systems. ▮▮

Dy 4 Systems
Kanata, Ontario
Canada.
(613) 599-9199.
[www.dy4.com].

Real-Time
Innovations
Sunnyvale, CA.
(408) 734-5009.
[www.rti.com].