Automating connectivity in defence assets

Military assets, which often last in excess of 20 years, cause significant problems when new technologies are brought in, each utilising different communication network types. Nick Pridham, Managing Director of Hamersham Limited, discusses the efficient automation of data exchange in modern defence systems.

Legacy, current and future defence assets have many different communication network types, such as CANBUS, Serial and RS422. If networks of different types remain separated, then a problem called 'stove piping can arise.' The stove pipe description evokes the idea of control system stacks in vertical columns that are not connected. If control system connectivity is not considered from the beginning, then getting different stove pipe columns to communicate retrospectively can be expensive and technically difficult.

The efficient automation of data exchange is imperative for modern defence systems, whether they be on land, in the air or on water. Any single or group of systems or devices must be able to send and receive data from any other system or device. The nature of the data exchange must be able to be Unicast, Multicast or Broadcast. This means simple point-to-point or many-to-many communication links are required to give systems and devices complete flexibility in how they want to exchange data.

Put simply, it means that any device type must have a seamless, reliable way of joining a communications network, reading and writing the data it needs to, and then disconnecting. Device types can be anything from deeply embedded sensors with limited memory and processing, to high-powered system control computers. A low cost, reliable and secure way for defence systems to exchange data is one of the key problems that has to be addressed by electronic system designers.

Another key networking consideration is the elimination of the need for a messaging server. This means that devices can communicate directly with each other on the same network without having to go through a central message server. This is referred to as peer-to-peer (P2P) communications.

The efficient automation of the data exchange process described is imperative. System designers have complex data exchange requirements as well as complex application development requirements. The ideal scenario therefore is for system designers to be able to put most of their development effort into the application and delegate the responsibility of networking to a piece of interfacing software. This software is often referred to as middleware. One of the tried and tested middleware solutions deployed in defence applications is called DDS – Data Distribution Service.

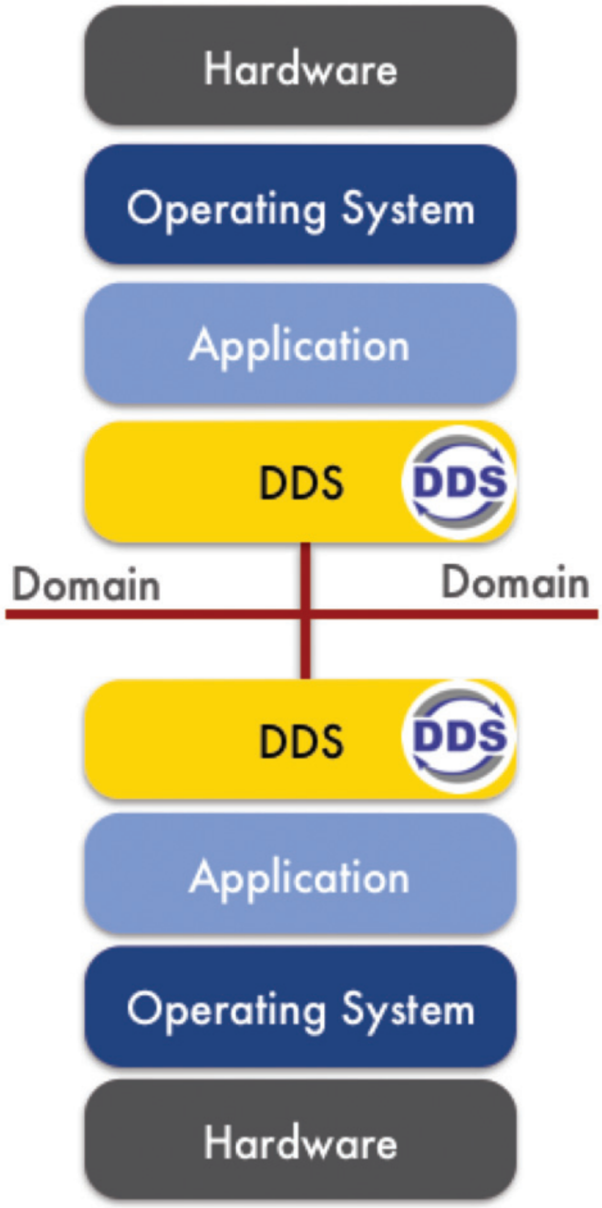
DDS middleware

DDS communications middleware is computer software that enables two separate software components, processes, or applications to exchange information, from either within one device, or between multiple devices. DDS is a specific kind of middleware. It is located in the layer that lies between the operating system (Linux, MAC OS, Unix, Windows, FreeRTOS, Greenhills, Integrity, etc) and system applications (vehicle control, weapons command, display information, etc), that allows for communications.

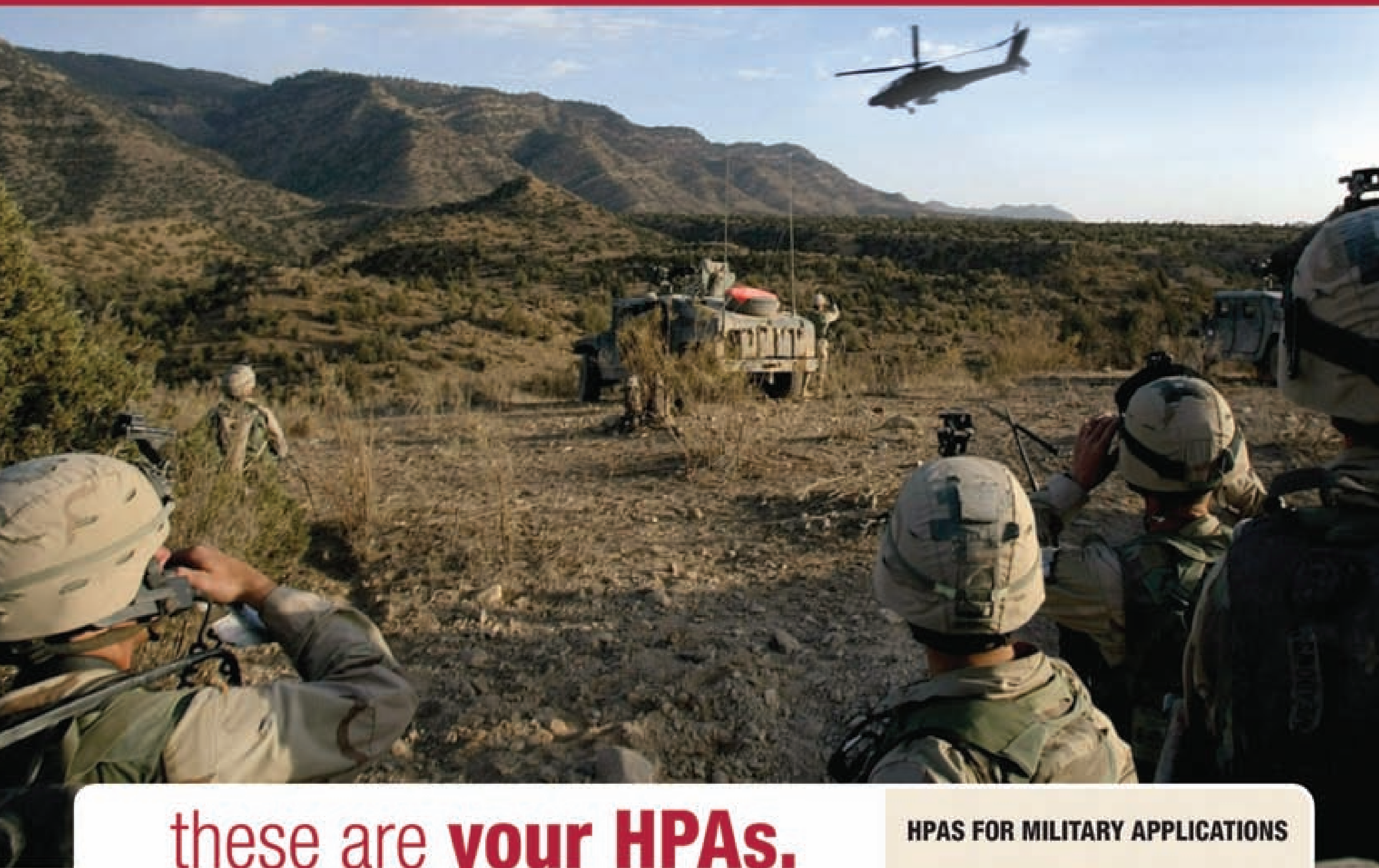
DDS communications middleware may be built into or added to one or both of the applications. Sometimes DDS is referred to as plumbing because it is the piece that connects multiple applications and allows data to pass through. The purpose of DDS Middleware is to simplify the designing, programming, and managing of software applications by streamlining the way these applications receive and process data.

DDS middleware can be deployed in a wide variety of software systems, from mobile devices (navigation, system displays, sensors) to static control and database systems. The equipment in these systems varies in screen and visual display capabilities, bandwidth capacities, and processing power. DDS middleware facilitates communications between these differing devices and can understand and support multiple programming languages (C, C++, Modern C++, C#, Java, PHP, Ruby on Rails, etc.).

We can use a vehicle fuel level sensor and a logistics control platform here as an example. They both function in vastly different capacities, but with DDS middleware, they are able to talk and work with each other. The two devices have completely different operating systems. The sensor maybe a deeply embedded RTOS whereas the logistics platform is a powerful database system running on Windows. DDS is the communication element that allows these diverse device types to communicate.



If this is **your office...**



these are **your HPAs.**

Superlinear® TWTAs
LEFT: 2.5 kW X-band
CENTER: 750 W Ku-band TWTA

Solid State BUCs
RIGHT: 160 W peak Ka-band GaN BUC

Warfighters shouldn't have to fight against their equipment.

That's why CPI amplifiers feature better efficiency and unmatched reliability even in the harshest of climates, keeping you connected anywhere at any time.

CPI offers:

- Better efficiency
- High reliability
- Rugged designs
- TWT, solid state and klystron technology
- Linearizer, BUC, and integral switch control options
- More than 20 service centers worldwide
- 24/7 technical support

For more information, please visit **www.cpii.com/satcom**, contact your local sales representative, or call us at **+1 (650) 846-3803.**

HPAS FOR MILITARY APPLICATIONS

- Up to 700 W Ka-band TWTAs
- SuperLinear® TWTAs in Ku- and Ka-band, delivering 25 to 540 watts of linear power
- C- and Ku-band HPAs for troposcatter applications
- 400 to 750 W C-, X-, and Ku-band indoor and outdoor TWTAs
- 2500 W SuperLinear® and CW X-band TWTAs
- X-, Ka- and Ku-band GaN BUCs

Download our new app!
Search: CPI Satcom

Download on the App Store
GET IT ON Google Play

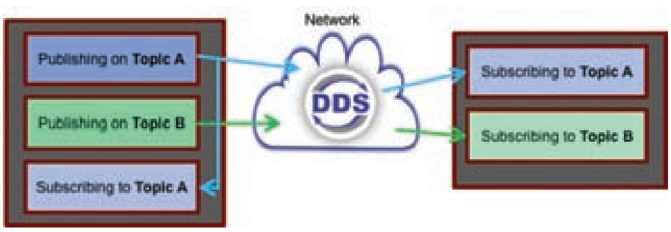
CPI
Communications & Power Industries
satcom products

DDS - Publish Subscribe Communication

Elimination of the requirement for a message broker is achieved by deploying a publish/subscribe methodology. Many communication middleware technologies are available. Most are based on a functional model. For example, RPC (Remote Procedure Call) and CORBA (Object Request Broker) are two examples of middleware that allow function calls to be distributed across the network between a client and a server. However, these architectures lead to tight coupling between the client and the server; this makes these systems difficult to extend.

The client-server architecture is appropriate for centralised data processing and works well in some systems and some use cases. In some client-server technologies, the drawbacks are increased integration costs for new capabilities and potential single point of failure. An alternative to this approach is the Publish-Subscribe architecture embodied in DDS. This architecture promotes a loose coupling between data producers and data consumers. The architecture is flexible and dynamic; it is easy to adapt and extend systems to changing environments and requirements.

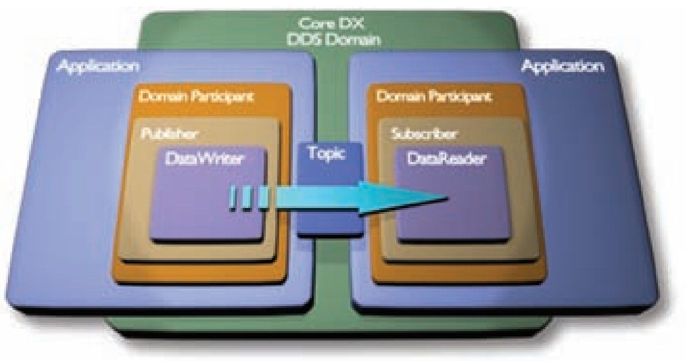
The figure below illustrates the DDS Publish Subscribe architecture where multiple Publishers and Subscribers exchange strongly typed data through a common Topic. The communications are controlled by a Quality of Service model.



Administration costs and DDS middleware

As system networks grow, keeping control of the costs to manage the networks is a huge challenge. As networks scale up in terms of traffic and device numbers, then automating this process becomes necessary. New devices must be able to join and leave the network without configuring IP addresses and port numbers. Fortunately, DDS allows networking complexity to be kept under control by using keyed TOPICS and a Discovery process.

Devices joining a network can automatically discover each other. IP addresses and port numbers are handled automatically by DDS and the communication process begins. DDS devices with data to offer and those DDS devices wishing to consume data communicate during the discovery process and agree to exchange data on a TOPIC. The data TOPIC could be 'System_KPI' and include varied data such as temperatures, error codes, fuel levels etc.



Life cycle costs and DDS middleware

Device service life of 20 years for defence assets can present a spare parts and maintenance challenge for system designers. Similarly, system networking and device data exchange must be considered for care and maintenance for the working life period of the machine. DDS middleware has a feature called Extensible Data Types allowing devices with more recent software upgrades to interoperate with legacy devices. This means system designers can implement changes for new devices without having to upgrade legacy devices in the field.

“The DDS Security standard contains a complete state-of-the-art security solution”

DDS has a way of ignoring new data types that it does not recognise and using the ones that it does.

Communications profiles

The DDS standards specify the mechanism for moving data – a typical communications middleware technology standard. However, DDS is much more. In addition to communications, DDS provides advanced data management, storage, organisation, filtering, redundancy, extensibility, and security. This rich set of features allows many different communication profiles to be configured and are collectively called ‘Quality Of Services’ (QOS). QOS settings are configured to control things such as:

- **History Depth:** How many samples are held in memory to be read by a new network participant?
- **Delivery Deadline:** How quickly shall a data topic be delivered?
- **Reliability:** How reliable is the data required to be? Are dropped packets allowed?

A total of 25 QOS settings are available to optimise the DDS data traffic in defence networks.

Small footprint - CoreDX DDS

To meet the requirements of working on a wide range of device types, DDS middleware must be small in footprint and very efficient. This is because deeply embedded devices often have very little Flash and RAM available. The CoreDX DDS product is 100 percent designed and developed by Twin Oaks Computing to meet the OMG’s DDS specification. There is no historical code, no code borrowed from the open source community, no code retrofitted to meet the CoreDX DDS requirements. This allows Hamersham, an EMEA distributor of Twin Oaks Computing, to deliver a quality, fully-functional DDS implementation with the smallest footprint. Our entire core library is less than 500KB and runs in environments with as little as 100KB of RAM. The full CoreDX DDS implementation is deployed on FPGAs, DSPs, PLCs, ECUs and other embedded environments. This small library size comes with a proportionally small Line of Code Count making it perfect for safety critical applications requiring DO-178B certification. CoreDX DDS is modular and contains additional run-time memory tuning parameters. Space constrained projects can select components of CoreDX DDS to meet their requirements and tune those components to reduce unnecessary memory utilisation. A small footprint DDS allows system designers the freedom to deploy exactly the same DDS middleware implementation in deeply embedded devices or powerful master computers.

DDS - An open standard

DDS is a standardised middleware according to an open standard and a range of different vendors exist. This means commercial risk is reduced because vendor lock-in is avoided. The vendor community is vibrant and meets quarterly for interoperability testing. DDS is not ITAR restricted and can be freely deployed in defence applications. The DDS standard is administered by the OMG organisation.

DDS is flexible and scaleable

Applications communicating with DDS might be running together on one host, or they might be distributed over multiple hosts, each with different architectures and operating systems. Applications using DDS for communications do not need to know the details of where their other applications are residing, or even if they exist. The Discovery mechanism built into DDS allows applications to come and go from a DDS network without requiring any changes to the applications or the network. This means a new system can be brought into the network, and start sending or receiving data, without any changes to existing applications.

DDS is fast

The Twin Oaks DDS Implementation was built from the ground up with performance in mind. The engineering staff at Twin Oaks Computing have a long history of writing and maintaining real-time and near real-time software, and this expertise was used in creating CoreDX DDS. CoreDX DDS is written in ‘C’ (with additional application language bindings available) for low overhead and memory savings. The CoreDX DDS baseline is tested and enhanced for performance at every step of the development process. The result is a high-quality DDS implementation with extremely low latency and high throughput capacity. CoreDX DDS data aggregation, multi-core data pipeline, and low latency event notification provide for throughput in the +900Mbps range and latencies below 75 usec over a 1Gbps ETHERNET network. The CoreDX DDS release includes source code for example benchmarking applications.

Security

The DDS Security standard contains a complete state-of-the-art security solution that is completely integrated into the DDS protocols (not simply layered on top of SSL). DDS Security includes: Identification, Authentication, Access Control, Integrity, and Confidentiality, allowing the designer full flexibility on a topic-by-topic level.

GMC