

Using DDS to Unlock the Power of TSN



This paper explains how the Data Distribution Service (DDS®) for Real-Time Systems standard by the Object Management Group® (OMG) can be combined with the emerging IEEE Time Sensitive Networking (TSN) set of standards. The result is a powerful distributed data-centric integration framework with high reliability and availability properties.

Lead Author

Rajive Joshi, Ph.D.

Principal Solution Architect, Real-Time Innovations (RTI)

Chair, Connectivity Task Group, Industrial Internet Consortium (IIC)

Chair, Automotive Task Group, Industrial Internet Consortium (IIC)

Introduction to DDS

DDS is a standard managed by the Object Management Group (OMG), the systems software standards body responsible for technologies such as middleware and modeling languages (e.g. UML). The OMG DDS standard is the first open international standard directly addressing data-centric publish-subscribe middleware for real-time systems. DDS features extensive fine control of real-time Quality of Service (QoS) parameters, including reliability, bandwidth control, delivery deadlines, and resource limits.

DDS is fundamentally designed to address the challenges of real-time communications. The DDS data-centric publish-subscribe (DCPS) model connects anonymous information producers (publishers) with information consumers (subscribers). The overall distributed application is composed of participants, possibly in separate address spaces, possibly running on different computers. A participant may simultaneously publish and subscribe to typed data-streams identified by names called “Topics.” The interface allows publishers and subscribers to present type-safe API interfaces to the application.

DDS defines a communications relationship between publishers and subscribers. The communications are decoupled in space (nodes can be anywhere), time (nodes and topics can start, join, or leave in any order at any time), and flow (delivery may be “best effort,” reliable, or at controlled bandwidth). It explicitly manages the communications data model, or types used to communicate between endpoints. It is thus a data-centric technology. Like a database, which provides data-centric storage, DDS understands

the contents of the information it manages. DDS is all about the data. In the industry, DDS is often referenced as an open standard for a “databus.”

At its core, DDS implements a connectionless data model with the ability to publish and subscribe to data with the desired QoS.

A DDS-based system has no hard-coded interactions between applications. The databus automatically discovers and connects publishing and subscribing applications. No configuration changes are required to add a new smart machine to the network. The databus matches and enforces QoS.

DDS overcomes problems associated with point-to-point system integration, such as lack of scalability, interoperability and the ability to evolve the architecture. It enables plug-and-play simplicity, scalability and exceptionally high performance.

DDS is fundamentally designed to work over unreliable transports, such as UDP or wireless networks. No facilities require central servers or special nodes. Efficient, direct, peer-to-peer communications, or even multicasting, can implement every part of the model.

Introduction to IEEE TSN

TSN is a set of standards by the Time-Sensitive Networking task group of the IEEE 802.1 working group. The standards define mechanisms for the time-sensitive transmission of data over Ethernet networks. Most of the projects define extensions to the IEEE 802.1Q – Virtual LANs (VLANs). These extensions in particular address very low transmission latency and determinism. Applications for TSN include converged networks with real time audio/video streaming and real-time control streams used in automotive or industrial control applications.

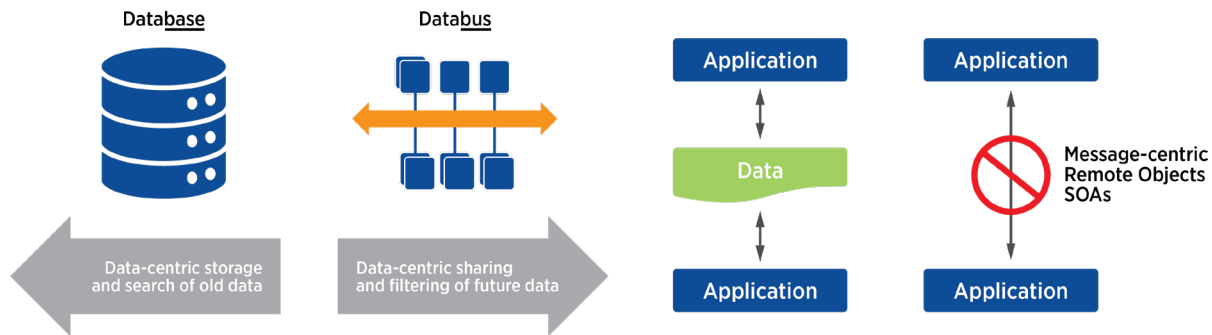


Figure 1: Database vs. Databus. In a data-centric architecture, applications communicate only with the data infrastructure, not with each other. A database implements data-centric storage that finds the right old data through search. A databus implements data-centric sharing that finds the right future data through filtering. Both technologies make system integration much easier, supporting larger scale, better reliability, and application interoperability

The TSN standards specified by IEEE 802.1 can be grouped into three categories that are required for a complete real-time communication solution. Each standard specification can be used on its own and is mostly self-sufficient. However, when used together in a coordinated manner, TSN as a communication system can achieve its full potential. The three basic components are:

1. Time synchronization: All devices that are participating in real-time communication have a common, synchronized understanding of time
2. Scheduling and traffic shaping: All devices that are participating in real-time communication adhere to the same rules in processing and forwarding communication packets
3. Selection of communication paths, path reservations and fault-tolerance: All devices that are participating in real-time communication adhere to the same rules in selecting communication paths and in reserving bandwidth and time slots, possibly utilizing more than one simultaneous path to achieve fault-tolerance.

This document explores how these categories can be leveraged when combining DDS and TSN.

DDS and TSN, a Natural Fit

There are several reasons why DDS and TSN play well together. Most fundamentally, both technologies provide one-to-many communications that support different levels of reliability for different streams of data. This section explores in more detail the implications of this similarity.

DDS dataflows and TSN streams

DDS revolves around a strongly typed Publish/Subscribe model where DataWriters are responsible for updating particular types of data and any number of matching DataReaders observe those updates. This granularity of interest is called a Topic and typical DDS systems consist of one to hundreds of Topics. The combination of a DataWriter with its Topic name can be seen as an identifier of the source of a DDS dataflow. All matching DataReaders, one to many, are sinks to that same flow.

Similarly, TSN has Talkers that update one or more streams delivering data to the connected Listeners. Such TSN streams are identified by their VLAN tag ID and the combination of a Talker with a VLAN tag ID can be seen as an identifier of the source of a TSN stream.

From this description, it is clear how DDS dataflows can be mapped directly onto TSN streams. Most DDS systems nowadays already leverage IP multicasting and replacing that mechanism with communications over TSN streams is a logical step. At the application level, the impact would be minimal, because the details of the underlying transport are invisible to DDS users. An example of this is given in the section called [An example mapping](#).

DDS Quality of Service and TSN Traffic Classes

In addition to the one-to-many nature of DDS, explained in the previous section, it also has quite an extensive set of different Quality of Service settings. The purpose of those is to instruct the middleware about, among others, the importance and urgency of the information in the different DDS dataflows. In line with the data centric philosophy, these QoS settings are applied per dataflow.

Some of those QoS settings have close similarities to the mechanisms that define TSN Traffic Classes. Examples of these are the Deadline, LatencyBudget and TransportPriority QoS settings. Other DDS-level QoS settings can be used to make smart decisions to select the proper underlying TSN stream. In other words, by selecting the correct QoS settings at the DDS level, system designers can rely on data distribution behavior from end-to-end: from the producing application all the way down to the network stack, over the network, and back up to the consuming application.

Predictable, low latency traffic

The full name of the DDS standard is “Data Distribution Service for Real-Time Systems” and DDS products have been designed to and used to implement such systems from their inception. Best practices associated with real-time behavior as well as high performance mechanisms have been followed. Thousands of mission-critical systems rely on, among others, DDS infrastructures.

This means that existing DDS ecosystems are demonstrably suitable for application domains that are targeted by TSN technologies, such as automotive and industrial automation.

Using DDS to Unlock the Power of TSN

So far, we have discussed the fundamental similarities between DDS and TSN. But what does DDS add to TSN to make it worthwhile combining the two? And why would DDS users be interested in exploring TSN as the underlying network? The subsections below explore how DDS can help to unlock the power of TSN.

A higher level of abstraction

While the interaction models for DDS and TSN are quite similar — both are fundamentally one-to-many publish/subscribe technologies — applying DDS on top of TSN results in a programming model at a higher level of abstraction. DDS users develop their systems by producing or consuming strongly typed, predefined elements of data. The data modeling constructs supported for this are rich and modern. Application developers see their data in native programming language constructs whereas the middleware takes care of all lower level mechanisms such as de/serialization or network interactions.

Standardized interactions

DDS consists of a group of open standards with the OMG standardization body. These are freely available for download at OMG's website, with the main specification found at <https://www.omg.org/spec/category/data-distribution-service/>. In addition to the functional description of DDS, this specification includes a standardized API to be used by application developers. With a DDS-TSN integration available, this API can be leveraged for a simpler learning curve, lower maintenance cost and off the shelf ecosystem support.

Scalable for mission critical systems, by design

The DDS pedigree is in large-scale, mission critical systems. This means that the addition of DDS to a TSN network will preserve those properties necessary for reliable, dependable systems.

A unified approach

Many systems consist of a combination of subsystems with different requirements and capabilities. Not all of those subsystems require the stringent real-time behavior provided by TSN. With DDS, it is possible to reuse the same infrastructure in different subsystem, seamlessly connecting them with gateways while preserving a unified data model. DDS is agnostic to operating system and programming language and as such can be leveraged in different types of environments. In such situations,

the common theme between the subsystems remains the data model as well as the rich ecosystem provided by DDS. This simplifies the task of introducing TSN into an existing system or design, or of extending a TSN-based system with additional subsystems with different non-functional requirements.

An Example Mapping

Management of a robotic device via a DDS infrastructure is an application that could leverage TSN at the lower level. Integration of the two technologies can happen at several stages of the system development process, each of them explained below.

Overview

This example introduces a system for controlling a robotic arm in a distributed setup. This means that the controlling application runs on a different machine than the robotic device itself. For analysis and predictive maintenance purpose, another machine is observing and virtually duplicating the robot's functioning (a "digital twin"). All of these applications are capable of issuing alarms whenever irregularities are discovered. These are displayed in a dedicated user interface application running on a separate machine. All applications involved communications over DDS. In table format, the flows and participants are identified in Figure 2.

In DDS-speak, the flow name is equivalent to the Topic name. The layout of the example system is illustrated in Figure 3.

System layout definition and schedule calculation

OMG DDS-XML defines a schema for defining DDS Entities in an XML format, allowing for a mechanism to capture system data flows in a descriptive way. With additional tags — yet to be defined by a new OMG DDS-TSN standard, as an extension to the existing DDS-XML schema — an indication of critical versus non-critical data flows as well as their frequencies and bandwidth can be provided. In combination with physical endpoint information, indicating which application runs on which node, sufficient information is available to feed to a TSN network scheduler for calculation of the different TSN flows with their tags as well as the timings and durations of the associated network time slices. Using TSN terminology: a DDS-aware Central User Configurator (CUC) tool will ingest the DDS-specific information and translate that into an interaction with the Centralized Network Configurator (CNC) tool with the goal to obtain a calculated network schedule, expressed in a TSN-standardized format.

Flow Name	Publisher	Subscribers	Purpose	Critical
DesiredPos	Controller	Arm, Twin	Objective position for robot arm	Yes
ActualPos	Arm	Controller, Twin	Actual position of robot arm	Yes
Alarm	Controller, Arm, Twin	AlarmUI	Indication of situation requiring attention	Yes
Heartbeat	Controller, Arm	SysMon	Overall health of the system	No

Figure 2: Flows and participants for distributed control of a robotic arm.

For those DDS data flows that are not marked as critical, no information is provided to the TSN network scheduler. This basically means that the associated data is transported in the remaining TSN best effort time slices.

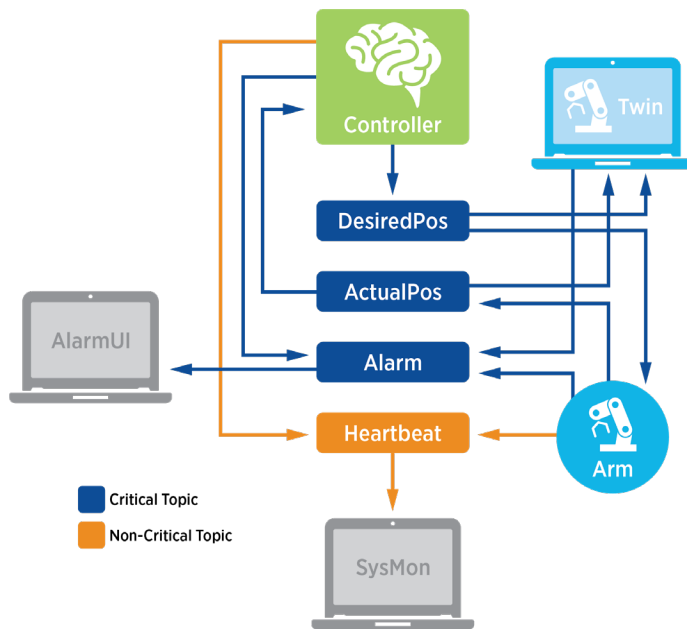


Figure 3: Data flows in a typical “smart” system.

In our example system, this means that the critical Topics, DesiredPos and ActualPos, need to be accompanied by their update frequencies — for example 1KHz — for the network scheduler to take into account. The Alarm Topic is critical as well, but aperiodic. In order to make that work with the concept of pre-scheduled time slots, an estimate of the required or desired allocated bandwidth as well as the maximum acceptable latency needs to be made upfront. This could happen with the help of tooling. Finally, the Heartbeat Topic is not critical and no additional information is required.

Runtime selection of TSN flows

With the TSN flows and timings calculated in the previous step, each DDS DataWriter is considered a TSN Talker and each DDS DataReader is considered a TSN Listener for those DDS Topics that contain information flagged as critical. Additionally, the DDS protocol is able to select the correct VLAN tag for each of them, providing the required network-level quality of service as delivered by TSN.

DataWriters and DataReaders for Topics not marked as critical will communicate their data over the TSN best effort time slices.

In our example system, this means that network packets for the critical Topics DesiredPos, ActualPos and Alarm need to have their headers stamped with the correct Ethernet-level tags, as indicated by the TSN flows calculated in the previous step. The DDS infrastructure will be responsible for that.

No such mechanism is needed for the Heartbeat Topic because it is not a critical flow.

Automatic configuration of network equipment

In addition to instrumentation of the DDS middleware itself, the different TSN-aware network equipment elements need to be configured to align with the DDS data flows. This could be the combined task of the CUC and the CNC, with the CNC either directly instructing all network switches about the schedule or the CNC providing the CUC with the information required for this. Additionally, the CUC will instruct the network interfaces.

For our example system, this means that the network interface cards for the machines running the Controller, Arm, Twin and AlarmUI nodes need to be made aware of the calculated time slices for their associated Topics. Additionally, the TSN-aware switch needs to be instructed accordingly.

OMG DDS-TSN Standardizations

OMG has identified the need for a standardized approach to combining DDS and TSN. It is in the early stages of defining different aspects of such an integration, but it is anticipated that aspects of defining standard mechanisms to run a DDS runtime infrastructure over a TSN network will be included. Additionally, it could specify a description of a standardized process to derive TSN network scheduling information from DDS system descriptions.

Having such open standards in place and available to all will allow DDS as well as TSN vendors to develop their integration solutions in a portable and vendor-independent fashion.

See Figure 5.

To learn more about TSN over DDS environments, please contact info@rti.com.

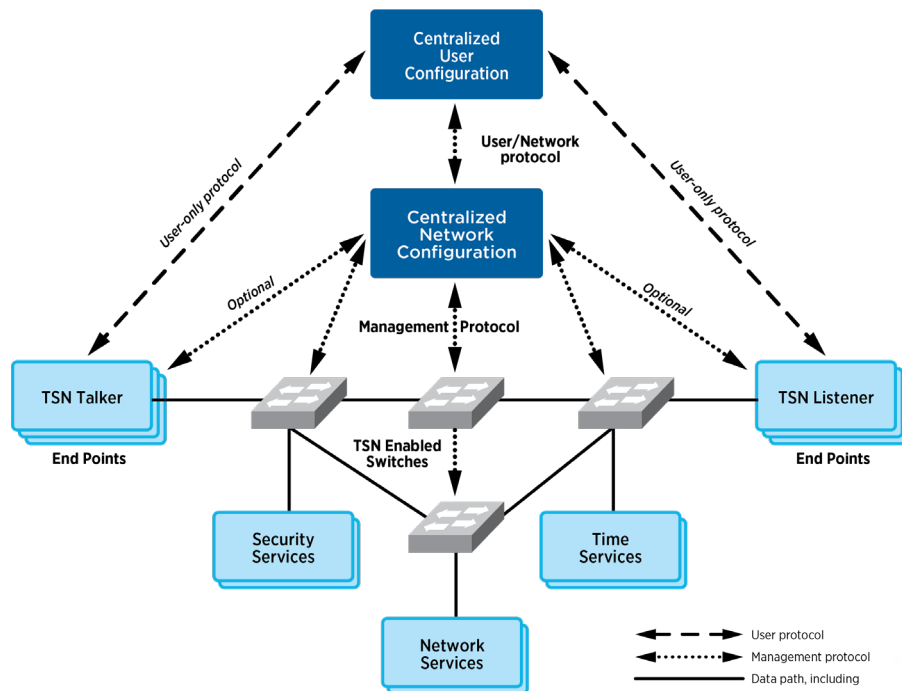


Figure 4: Roles in TSN System Configuration. Key DDS interfaces include the CUC -CNC User/Network Protocol and CUC - TSN End Point interface.

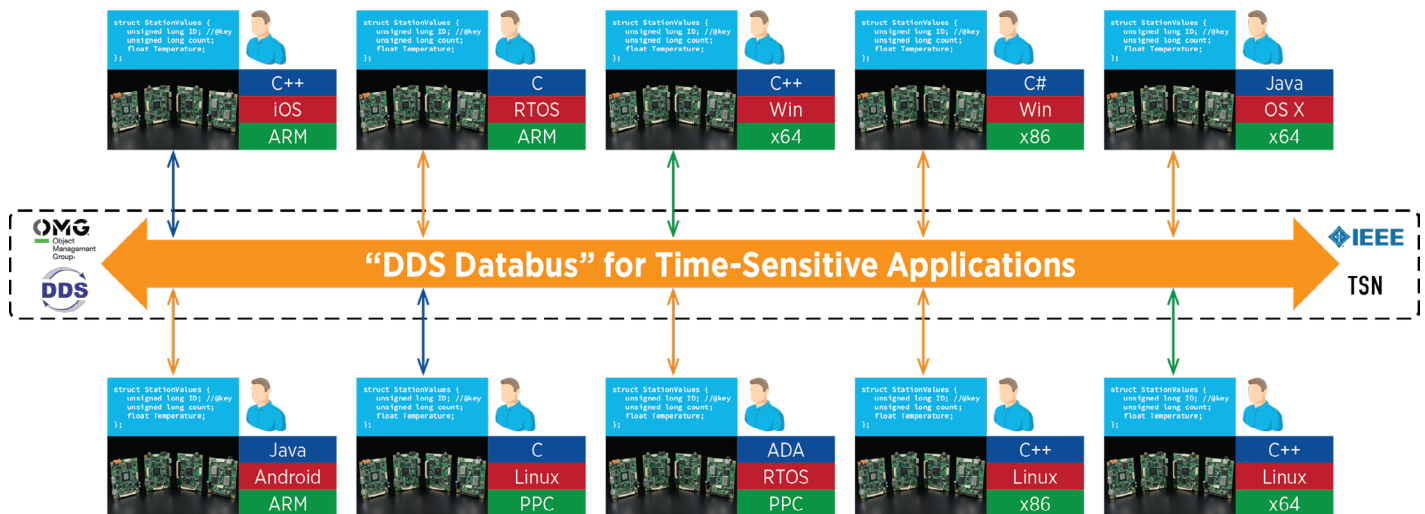


Figure 5: A powerful distributed data-centric integration framework with high reliability and availability properties.

ABOUT RTI

RTI is the real-time data streaming company for intelligent distributed systems. RTI Connext® software is the critical nervous system for over 2,000 designs across Aerospace and Defense, MedTech, Automotive, and Robotics.

RTI, Real-Time Innovations and the phrases “RTI Runs a Smarter World” and “Your systems. Working as one,” are registered trademarks or trademarks of Real-Time Innovations, Inc. All other trademarks used in this document are the property of their respective owners. ©2026 RTI. All rights reserved. TI-001 V2 0126