Real-Time Innovations’ NDDS networking middleware is fundamentally changing the way complex distributed systems are being architected. Traditional client-server solutions offer standard protocols, but often don’t fit real-time systems well. Publish-subscribe networking is an elegant, powerful addition. It allows many distributed systems to efficiently share data in a network. With the recent addition of Object Management Group’s Data-Distribution Service (DDS) standard, publish-subscribe networking is now ready for the most demanding applications.

Real-Time Innovations created the first commercial publish-subscribe middleware for real-time systems—Network Data Distribution Service (NDDS®). It works on multiple architectures, operating systems, compilers, and physical networking transports. NDDS was designed from the ground up for high-performance and embedded applications.

NDDS is field proven and successfully used in many mission-critical applications around the world, including optical transport interconnect messaging, Navy shipboard communications, air-traffic control systems, and distributed industrial control.

---

**NDDS 4.0**

Real-Time Publish-Subscribe Network Middleware

“NDDS provides a scaleable real-time COTS middleware that is easy to integrate and has proven to be very reliable in the field.”

Sea SLICE Lead Software Engineer

<table>
<thead>
<tr>
<th>Requirement</th>
<th>NDDS Delivers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct transfer from node-to-node</td>
<td>Faster than other middleware; performance is there when you need it</td>
</tr>
<tr>
<td>Support for many architectures, OSES, languages and network transports</td>
<td>Connects complex systems and expands to cover future needs</td>
</tr>
<tr>
<td>Dynamic reconfiguration</td>
<td>Nodes can enter and leave system in any order at any time</td>
</tr>
<tr>
<td>Many-to-many communications</td>
<td>Send the data everywhere you want it, now</td>
</tr>
<tr>
<td>Scalability</td>
<td>Build large systems with thousands of topics</td>
</tr>
<tr>
<td>Multicast support</td>
<td>Distribute data to many nodes efficiently</td>
</tr>
<tr>
<td>Configurable QoS</td>
<td>Tune to meet complex delivery requirements</td>
</tr>
<tr>
<td>No central server</td>
<td>Prevents single point-of-failure and bottlenecks</td>
</tr>
</tbody>
</table>
Introduction to Publish-Subscribe

There are two basic networking paradigms in common use: client-server and publish-subscribe.

Client-Server

Client-server networks connect multiple clients through a central server. Most enterprise networking is client-server based, including HTTP, CORBA, and DCOM. Client-server is best for:

- Naturally-centralized information
- No single-point failure or data bottleneck problems
- Little data between clients

Publish-Subscribe

Publish-subscribe networks push data out, in the same way that magazines and newspapers deliver content. Publish-subscribe has gained rapid growth and acceptance due to its simplicity, versatility, and low overhead. Publish-subscribe is best for:

- Complex networks and data flow
- Fault-tolerant networks
- Time-critical node-to-node transfer
- Dynamic “plug and play” operation

OMG Networking Standards

The Object Management Group (OMG) developed both the CORBA and the DDS standards. CORBA addresses client-server object distribution, DDS addresses publish-subscribe data distribution.

DDS extends the publish-subscribe model for real-time systems. It supports deadlines, publisher arbitration and failover, reliability tuning, and more.

<table>
<thead>
<tr>
<th>Property</th>
<th>CORBA</th>
<th>DDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Model</td>
<td>Client-Server</td>
<td>Publish-Subscribe</td>
</tr>
<tr>
<td>Primary Use Case</td>
<td>Call Remote Methods</td>
<td>Send Data to Many Nodes</td>
</tr>
<tr>
<td>Implementation</td>
<td>Complex, General</td>
<td>Simple, Lightweight</td>
</tr>
<tr>
<td>Target Application</td>
<td>Many</td>
<td>Real-Time Systems ranging from Embedded to Enterprise</td>
</tr>
<tr>
<td>QoS Configurability</td>
<td>Limited Real-Time QoS</td>
<td>Determinism/Reliability Levels, Deadline, Resource Usage, Bandwidth Usage, More</td>
</tr>
<tr>
<td>Underlying Network</td>
<td>Connection-Oriented</td>
<td>Connectionless</td>
</tr>
</tbody>
</table>

“CORBA covers the client-server communication requirements for distributed real-time systems and DDS covers the data-distribution requirements. The DDS specification is a significant addition to OMG’s real-time networking standards.”

Char Wales,
Co-chair of the MARS Platform Task Force at OMG

The OMG Middleware and Related Services Platform Task Force (MARS PTF) produced the DDS specification to address the need for a data-centric publish-subscribe standard. The DDS and CORBA standards share a common communication definition, the OMG’s Interface Definition Language (IDL). NDDS 4.x is implemented to coexist with any CORBA implementation.
NDDS Real-Time Publish-Subscribe Network Middleware

NDDS is network middleware that sits between your application and the operating system. It is a layer of software that sits on top of a network stack. It simplifies the underlying low-level network code with a common, standards-based, application programmer interface (API).

NDDS alleviates the need to manage complicated initialization procedures, network addresses, failover, and a host of other tricky networking chores.

NDDS takes care of these tasks. The programmer simply uses a few NDDS function calls, replacing hundreds of socket calls.

The publish-subscribe model defines:
- Publishers, which simply create a publication and give it a topic name. To send an issue (data), the application just calls a single NDDS function.
- Subscribers, which simply create a subscription to a topic name and tell NDDS what to do when a new issue arrives.

NDDS handles the network I/O, transparently sending each published issue to all interested subscribers.

The publish-subscribe model takes care of channel configuration and data distribution for the application.

Availability

**Multi-Language:**
- C
- C++
- Java
- More

Extensive training and consulting available

**Multi-OS:**
- Windows
- Linux
- Solaris
- VxWorks
- Integrity
- LynxOS
- More

Additional Information

See [www.rti.com](http://www.rti.com) for:
- Publish-Subscribe Overview
- Performance Paper
- Build Your Own Middleware Guide
- DDS Standard
- Load Calculation Spreadsheet
- Ethernet Can Be Real Time Paper
- RTPS Wire Protocol Specification
- Application Examples
NDDS Application Examples

**Land**

* DARPA Grand Challenge Unmanned Autonomous Vehicle
  Flying Fox, a 2005 DARPA Grand Challenge Race entrant built by Autonomous Vehicle Systems, relies on NDDS for synchronized, distributed sensor integration for their unmanned autonomous vehicle.

* Schneider PLC Devices for Factory Automation
  Schneider Automation uses NDDS to provide global data access in its new line of programmable logic controllers.

**Air**

* CAE SimXXI Flight Simulator
  CAE powers their next generation flight simulators with NDDS for real-time communications between simulator subsystems.

**Sea**

* US Navy LPD-17
  NDDS forms the backbone for the entire Ship-Wide Area Network (SWAN) on the Navy’s newest ship.

* Schilling Electric Work-Class ROV
  ALSTOM Schilling Robotics built the communications system for the Quest remotely operated undersea vehicle using NDDS.

**Space**

* NASA Robonaut
  Johnson Space Center uses NDDS for simulation and communications of their EVA robot, Robonaut.

---

About RTI

Real-Time Innovations, Inc. the expert in real-time information networking, leads the industry with high performance standards-based software solutions for data-critical applications. Its products and consulting services provide the infrastructure for national railways, air traffic control, traffic monitoring, mission-critical combat systems, financial transaction processing, and industrial automation. RTI’s flagship product, NDDS, is middleware based on the Object Management Group’s (OMG) Data Distribution Service (DDS). NDDS provides the essential foundation for real-time communication in a networked system and enables a new class of embedded to enterprise (e2E) applications. Companies such as Raytheon, Nikon, Omron, Harmonic, Applied Materials, Schneider Automation, Boeing, Lockheed Martin and the US Military rely on RTI technology for their real-time, data-centric, distributed applications. Headquartered in the heart of Silicon Valley since 1991, RTI is a privately held company.