DDS in Low-Bandwidth Environments

Workshop on Real-time, Embedded and Enterprise-Scale Time-Critical Systems
April 17-19, 2012, Concorde La Fayette, Paris, France

Jaime Martin Losa
CTO eProsima
JaimeMartin@eProsima.com
+34 607 91 37 45
www.eProsima.com
Agenda

- DDS in Low-Bandwidth environments
  - Motivations
  - DDS Behaviour out-of-the-box
    - Poor performance
    - Reasons
    - Optimizations Required
    - Performance after optimizations
    - Available Market Products/ Success Cases
  - Optimization Details
    - Discovery
    - Data Compression
    - Protocol
Motivations

DDS in Low Bandwidth Environments
Motivations

- Currently, DDS main market is Defense
  - Defense applications use intensively radio & satellite links
- Features of Tactical Radio & Satellite links:
  - Low Bandwidth: Even 2400 bps or less.
  - Shared Bandwidth: Even 32 nodes or more.
  - Disconnections and Packet Loss
  - High latency.
Motivations (II)

- **Radio link typical Capabilities:**
  - VHF Radio (Range 20 km): < 64 kbps shared
  - UHF Radio (Range 1 km): < 1 Mbps shared
  - 4-32 nodes sharing bandwidth in the same Radio net.
  - High latency, Packet loss, disconnections

- **Satellite link typical capabilities**
  - Channel bandwidth: from 64 kbps to several Mbps
  - High latency, Packet loss, disconnections
Motivations (III)

- Real Example: Spanish Army
  - Uses VHF, UFH and Satellite links intensively for Data Transmissions.
  - Propietary comms solutions for their different C2 systems
    - Poor Performance
    - Lack of Interoperability
DDS Behaviour
out-of-the-box

DDS in Low Bandwidth Environments
DDS Behaviour out-of-the-box

- Very Long discovery times
- Very Low effective throughput
- Example: 6 nodes, VHF Radios (4800 bps-shared), RTI DDS
  - Discovery: >45 Min! (unusable)
  - Effective throughput: <100 bps! (unusable)
Poor performance: Reasons

- Chatty discovery protocol
  - Requires dozens of messages for a single system
  - Number of messages = $K \times (\text{Number of Nodes})^2$
- Large Protocol Headers
  - RTPS typical header is 56 bytes long
- DDS does not compress data.
- QoS default values are not the best suited for this scenario.
Optimizations required

- **Discovery**: Reduce the number of messages
  - Should be of the order of number of nodes
  - The payload of the discovery messages should be small

- **Protocol**:
  - Reduce header length

- **Compress data and metadata**

- **Use Multicast for data, metadata & heartbeats**

- **Qos**: Set up according bandwidth and latency

- **Nack Based Reliability, Use of flow controllers, Type optimization…**
Performance after optimizations

- Discovery:
  - Number of messages = \( O(\text{number of nodes}) \)
  - Very small message payload, 100-150 bytes.
- RTPS Headers
  - Reduced from 56 to 26 bytes
- Data Compression:
  - 50%-80% of compression for typical C2 data
- Multicast for Data, Discovery metadata and heartbeats
- Nack Based Reliability, Use of flow controllers, Type optimization…
Performance Example

- **Example**: 6 nodes, VHF Radios (4800 bps-shared), RTI DDS

<table>
<thead>
<tr>
<th></th>
<th>Out of the Box</th>
<th>Optimized</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovery Time</td>
<td>&gt;45 min</td>
<td>&lt;20 seg</td>
</tr>
<tr>
<td>Effective data throughput</td>
<td>&lt;100 bps</td>
<td>&gt;2000 bps (*1)</td>
</tr>
</tbody>
</table>

(*1) Radio Effective bandwidth decreases with number of nodes.

- **Example app**: C2 system: squadrons of 6 tanks
  - Quick System Startup
  - Update position and status every 5 seconds
  - Bandwidth Room for alarms, tactical chat, enemy positions, etc.
Available Market Products

- eProsima Low Bandwidth Plugins for RTI DDS
  - eProsima LB Discovery Plugins
  - eProsima LB Compression Transport
  - eProsima LB Optimized RTPS Transport
  - eProsima LB Simulation Transport
    - Allow Radio/Satellite link simulation
    - All plugins can be used together.

- Success Case: Spanish Army
  - Spanish Army selected DDS for C2 interoperability.
  - Intensive use of VHF Radios
  - Implemented already in three main C2 systems.
Available Market Products (II)

- OpenSplice Supports ZLIB compression and static discovery
- OpenDDS, CoreDx, others: ?
Discovery Optimization Details

DDS in Low Bandwidth Environments
Overview

- What is discovery?
- Discovery phases
  - Participant discovery phase
  - Endpoint discovery phase
- eProsima LBDP
  - Endpoints Plugin: LBEDP
  - Participant Plugin: LBPDP
- User Traffic Hints.
What is discovery?

- The process by which domain participants find out about each other’s entities
  - Each participant maintains database on other participants in the domain and their entities

- Happens automatically behind the scenes
  - “anonymous publish-subscribe”

- **Does not** cross domain boundaries

- **Dynamic** discovery
  - Participants must refresh their presence in the domain or will be aged out of database
  - QoS changes are propagated to remote participants
Discovery phases

- Two consecutive phases
  - Participant discovery phase
    - Participants discover each other
    - Best-effort communication
  - Endpoint discovery phase
    - Participants exchange information about their datawriter and datareader entities
    - Reliable communication

- Steady state traffic to maintain liveliness of participants
Participant discovery phase

- Participants periodically announce their presence using RTPS DATA message
  - Contains participant GUID, transport locators, QoS
  - Initially sent to all participants in “initial peers” list, then sent periodically to all discovered participants
  - Sent using best-effort
Endpoint discovery phase

- DataWriter/DataReader discovery
  - Send out pub/sub DATA to every new participant
  - NACK for pub/sub info if not received from a known participant
  - Send out changes/additions/deletions to each participant

- Uses reliable communication between participants

- DDS matches up local and remote entities to establish communication paths
Discovery start-up traffic

User creates Data Writer Foo
Send DATA to participants in database

Already know about B
Add B to database of participants
random sleep

Add A to database of participants
random sleep

Participant created on A
Send DATA to peer hosts

Participant created on B
Send DATA to peer hosts

DataWriter DATA Foo
(sent reliably)

Add publication C to database of remote publications
Discovery steady-state traffic

Node A

Update liveliness of participant B
Periodic announcement of participant DATA

lease duration

Participant DATA B is not received
Remove participant and its pubs/subs from database

Node B

DATA participant B
Periodic announcement of participant DATA

Update liveliness of participant A
B goes down

DATA participant A
participant DATA send period

DATA participant A
Discovery Implementation

- Discovery is implemented using DDS entities known as Built-in Data Writers and Built-in Data Readers
  - Uses same infrastructure as user defined Data Writers/Data Readers
  - Participant data is sent best effort
  - Publication/subscription data is sent reliably

- Three Built-in topics (keyed):
  - DCPSParticipant
  - DCPSPublication
  - DCPSSubscription
Discovery phases: Visually

A: Hello!
B: Pleased to meet you!

End of First Phase

A: These are my Pubs
B: ok

A: ok
B: Now These are my Pubs
A: ok

A: These are my Subs
B: ok

B: Now These are my Subs
A: ok

End of Second Phase
Endpoints Discovery Optimization

- **Goals:**
  - Reduce the discovery information transmitted.
  - Reduce net traffic: Less Packets.

- **Scenario:**
  - We now most details of the participant applications in advance.

- **Solution:**
  - Suppress second discovery phase.
  - Information about endpoints stored in XML files.
Endpoints Discovery Optimization

Participant 1
- Participant Built-in Data Writer
- Participant Built-in Data Reader
- Publication Built-in Data Writer
- Publication Built-in Data Reader
- Subscription Built-in Data Writer
- Subscription Built-in Data Reader

Participant 2
- Participant Built-in Data Writer
- Participant Built-in Data Reader
- Publication Built-in Data Writer
- Publication Built-in Data Reader
- Subscription Built-in Data Writer
- Subscription Built-in Data Reader

Participant Data Msg

Best Effort

XML

XML

XML

XML
File Based Discovery

B: Pleased to meet you!
A: Hello!

B: These are my Pubs
A: These are my Pubs
B: ok
A: ok

B: Now These are my Pubs
A: ok

A: These are my Subs
B: ok

B: Now These are my Subs
A: ok

End of First Phase
End of Second Phase

Data of pubs and subs of each participant loaded from a XML File
Participant Discovery Optimization

- **Goals:**
  - Reduce even more the discovery information transmitted.

- **Scenario:**
  - We now most details of the participant applications in advance.

- **Solution:**
  - Reduce the participant information transmitted.
  - Information about participants stored in XML files.
LBPDP: Discovery Entities

Participant 1
- Participant Built-in Data Writer
- Participant Built-in Data Reader
- Publication Built-in Data Writer
- Publication Built-in Data Reader
- Subscription Built-in Data Reader
- Subscription Built-in Data Writer

Participant 2
- Participant Built-in Data Writer
- Participant Built-in Data Reader
- Publication Built-in Data Writer
- Publication Built-in Data Reader
- Subscription Built-in Data Writer
- Subscription Built-in Data Reader

Reduced Participant Data Msg

XML

Best Effort
Results

- Number of messages = $O(\text{number of nodes})$
- Very small message payload, 100-150 bytes
- Very low discovery times.
Data Compression Optimization details

DDS in Low Bandwidth Environments
Compression details

- Compression at Transport Level
- Several compression libs used
- Several modes of operation
Compression at transport level

- **Compression at Transport Level**
  - Stackable: Use it in any transport: UDP, Serial, Ad hoc...
Several compression libs

- Several compression libs used:
  - ZLIB
  - BZIP2
- Easy to add more by the user.
  - Through Public API.
- Tested:
  - LZO : LZO1X, LZO1B & LZO1F
  - UCL : UCL_NRV2B, UCL_NRV2D & UCL_NRV2E
Several modes of operation:

- Fixed Algorithm
- Algorithm depending on packet size.
- Automatic: when CPU is not the bottleneck, the plugin select the best algorithm for each package.
RTPS Optimization Details

DDS in Low Bandwidth Environments
Optimized RTPS: Overview

- Optimized RTPS for low bandwidth scenarios
- Implemented as a transport.
Optimized RTPS

- RTPS Optimizations:
  - RTPS Header from 20 bytes to 1 byte.
  - RTPS SubmessageHeader from 4 to 3 byte.
  - RTPS extraflags for DATA and DATA_FRAG eliminated (1 byte)
  - ReaderID and WriterID from 4 to 1 byte each (so 2^3 writers or readers per participant)
  - SequenceNumber from 8 to 5 or less bytes (more than enough for these scenarios)
  - ...

- Save more than 30 bytes!
eProsima LB RTPS: Implemented as a transport

- Implemented as a transport
- Stackable:
  - Can be used with any transport and it is stackable, so for example you could use:
    - LB RTPS -> UDP
    - LB RTPS -> Compression Transport -> UDP
About eProsima
About eProsima

- Experts on middleware, focused on DDS.
- OMG Members.
- RTI DDS Distributor for Spain and Portugal.
About eProsima: Products And Services

- **eProsima Products:**
  - DDS based: Plugins, add-ons, adaptors, etc

- **Services:**
  - Communication modules, App development, DDS training, Support.

- **R&D:**
  - R&D Projects with enterprises and universities.

- **Quality: ISO 9001**
  - Design, Development, Marketing and Support of Software.
Customers (I)

- **Amper Programas:**
  - BMS
  - Simacet (Main Spanish C2 System)

- **Cassidian:**
  - UAVs - Neuron, Atlante
    - Ground Station Comm Server

- **INDRA:**
  - Defense (BMS, UAV PASI)
  - Air Traffic Control,
  - SESAR, ATC Interoperability

- **Spanish Army:**
  - IDT :Tactical Data Interface
Customers (II)

- Isdefe
- Spanish Army: JCISAT, DGAM
- CATEC-FADA: R&D Aerospatial
- Santa Barbara: Armoured Vehicles
- RTI
- GMV
Customers (III)

- Tecnobit: COSMOS, Reserved Projects.
- IKERLAN: R&D.
- Navantia: F105 (Aegis)
- Boeing: Atlantida, Swim suit
**eProsima Products.- Index**

- **eProsima Low Bandwidth Tools for DDS:**
  - Set of plugins to enable DDS communications over low bandwidth links, optimizing the protocol and compressing the data.
  - Includes a simulation plugin to simulate different links such as Tactical Radios and Satellites

- **eProsima Client-Server:**
  - RPC over DDS

- **eProsima DDS-Web Services Bridge**
  - Enables DDS Enterprise Integration

- **eProsima DDS Non-Intrusive Recorder.**
  - Stores DDS communication history in a data base.
Thank you!

Jaime Martin Losa
CTO eProsima
JaimeMartin@eProisma.com
+34 607 91 37 45
www.eProsima.com