Tools to Help Plan, Test, and Configure Your Federation for Performance

Integrated Training Program

Defense Modeling & Simulation Office
(703) 998-0660 Fax (703) 998-0667
hla@msis.dmso.mil
http://www.dmso.mil/

Current as of 10 September 1999

Outline

- Understanding performance
- Describing a federation execution
- Measuring performance
- Configuring your federation

Outline

- **Understanding performance**
- **This section defines performance and the performance framework. It is a general overview for the rest of the presentation:**
  - **Performance definition**
  - **Performance framework**
  - **Tools that support performance framework**

What is Performance?

Integrated Training Program

- **Merriam-Webster definition:**
  - **1 a : the execution of an action b : something accomplished : DEED, FEAT**
  - **2 : the fulfillment of a claim, promise, or request : IMPLEMENTATION**
  - **3 a : the action of representing a character in a play b : a public presentation or exhibition <a benefit performance>**
  - **4 a : the ability to perform : EFFICIENCY b : the manner in which a mechanism performs <engine performance>**
  - **5 : the manner of reacting to stimuli : BEHAVIOR**
- **The word performance in and of itself does not lend to quantifiable metrics for assessment. Performance is a measure of how well an action occurs compared to how it was intended.**

Different Perspectives

Integrated Training Program

- Each domain in M&S has different connotations for what performance means
  - Analysis: how much faster than wall-clock time can the system run without compromising validity, repeatability, and extractability of results
  - Test and Evaluation: can data be transmitted/received within time constraints of algorithms & hardware in the loop
  - Training: how many objects can be simulated with a given number of federates within latency & time scale constraints

- Performance of a system must be defined
  - What aspect of functionality/behavior is of importance?
  - What quantifiable metrics exist and how can they be measured and analyzed?
  - How does each component affect performance?
  - Understand that performance is affected by the entire system!
  - Understand dependencies between components!

What is the Federation Performance Framework?

- **A common framework for defining the aspects of HLA <u>federations</u> which bear on their runtime performance**
- **Foundation for planning federation executions and defining performance of HLA federation components**
  - **Run-time Infrastructure (RTI)**
  - **Federate Capabilities**
  - **Hardware Requirements**
  - **Network Requirements**
- **Comprised of two basic components**
  - **Federation Execution Planners Workbook (FEPW)**
  - **Performance Benchmark Programs**

Purpose of the FEPW

Integrated Training Program

- **Initially to understand RTI performance requirements and capabilities**
  - **How HLA is being used**
  - **Performance needs of actual federations**
  - **Input for benchmark definitions for RTIs**
- **Now supporting broader uses**
  - **Planning of federation executions**
    - **Insight into the entire planning process**
    - **Understanding relationships among federates**
    - **Understanding timing and coordination requirements**
  - **Documenting prior federation executions**
  - **Functional tools that support FEPW & performance**
    - **FEPW Editor, Federation Verification Tool, and RIDeditor**

What is the FEPW?

- **Federation Planner's Workbook (FEPW)**
  - **Assists federation developers in the planning of High Level Architecture (HLA) federation executions**
  - **Template for capturing the run-time federate and federation characteristics**
    - **Summary Tables**
    - **Host Table**
    - **LAN Tables**
    - **RTI Services Tables**
    - **Object/Interaction Tables**
    - **Cross-Reference Tables**
    - **Data Representation Tables**
  - **Provides a mechanism for describing the federation physical structure and performance and resource requirements**
- **FEPW Data Interchange Format (DIF) support**
  - **DIF files are standard ASCII file formats for exchanging HLA information**
  - **FEPW DIF enables sharing of federation execution characteristics among automated tools and across federations**

Performance Benchmarks

- **Goals of the Benchmark Programs**
  - **Performance indicators for each of the major categories of inter-federate exchange through the RTI.**
  - **Simple and unambiguous tools that can be applied by general users of the RTI.**
  - **Easy to understand metrics that facilitate comparison and investigation of factors influencing federation performance.**
  - **Source code that can be easily distributed and compiled on all RTI supported platforms.**
  - **Benchmark programs that are parameterizable using simple command-line arguments and FED file modifications.**
- **Benchmarks are evolving based on experiences in HLA and input from the user community.**
  - **DMSO is interested in feedback on the dimensions of performance that are important to you!**
  - **Benchmarks are being elevated to a formal product under configuration management and a regular review cycle.**

Benchmark Programs Defined

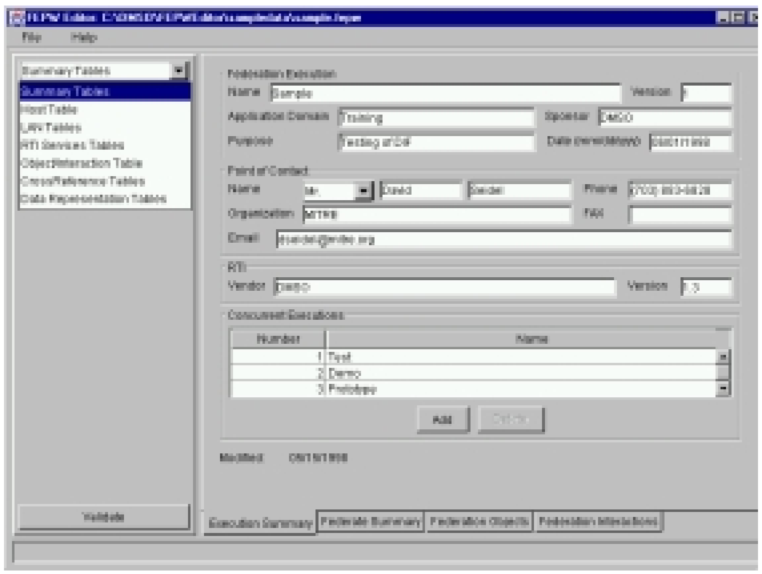
- There a four benchmarks currently defined:

- Update Latency Benchmark
  - Measures the round trip time (update latency) for an Update Attribute Values (UAV) service call
  - Key arguments: size of attribute, number of federates, number of objects per federate
- Update Throughput Benchmark
  - Measures the number of update attribute values per second that are possible for the given system configuration
  - Key arguments: size of attribute, number of federates, number of objects per federate

Benchmark Programs Defined (Cont'd)

- **Time Synchronization Benchmark**
  - **Measures the number of RTI time step cycles that can be processed by the RTI per second**
  - **Key arguments: number of federates, lookahead for the federates**
- **Ownership Management Benchmark**
  - **Measures the number of round trip ownership transfers per second per federation execution**
  - **Key arguments: number of federates, number of objects per federate**

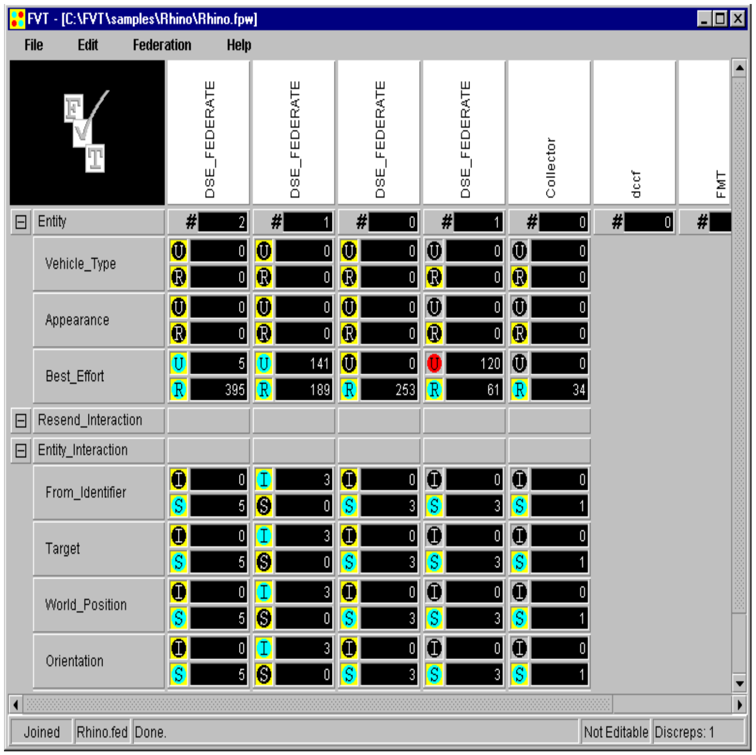
FEPW Editor



- **The Federation Execution Planner's Workbook (FEPW) Editor assists federation developers in the Federation Integration and Test phase of the FEDEP**
  - **By providing federation developers the ability to document the characteristics of the federation execution other tools such as Federation Verification Tool (FVT) can verify that a federation executed as planned**
- **FEPW Editor can be used to**
  - **Read in a Federation Object Model (FOM)**
  - **Add run-time characteristics for the execution**
    - **Host & LAN topology**
    - **Federate publish & subscription responsibilities**
    - **Expected data rates per federate objects & interactions**
- **How does FEPW Editor operate?**
  - **FEPW Editor is a pre-execution tool that allows you to document FEPW information and save that information in a standard Data Interchange Format (FEPW-DIF)**
- **What does it do?**
  - **Allows federation developers to document the characteristics of the federation execution which can be verified by the Federation Verification Tool (FVT)**

Federation Verification Tool

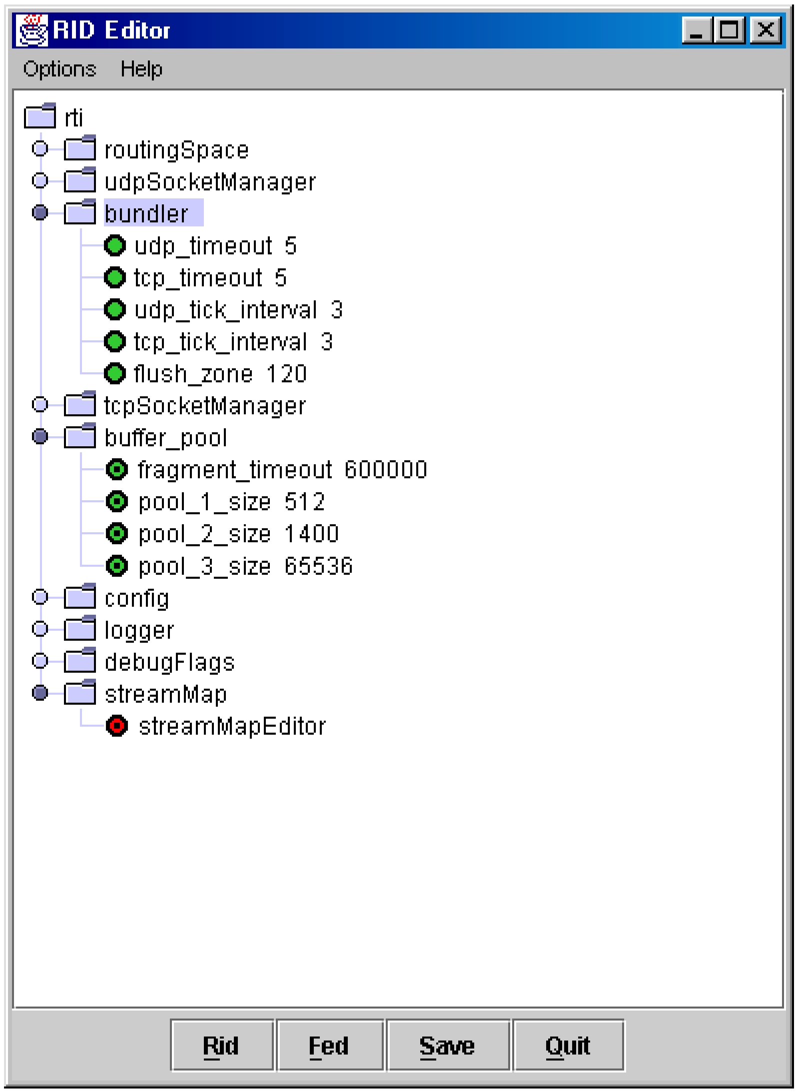
Integrated Training Program



- **The Federation Verification Tool (FVT) assists federation developers in the Federation Integration and Test phase of the FEDEP**
  - **Verifies that each federate is meeting their data exchange responsibilities as specified in FEPW**
- **FVT can be used to**
  - **Assess consistency across federation plans (FOM, FED, FEPW)**
  - **Assess individual federate conformance to federation requirements**
  - **Assess federation conformance to federation requirements during integration**
- **How does FVT operate?**
  - **FVT joins the federation as a federate, collects data from federates through standard RTI interface, and then checks federate behavior against the federation requirements**
- **What does it do?**
  - **The FVT is verifies that each federate updates and reflects the objects and sends and receives the interactions it is responsible for in the federation**

RID Editor

Integrated Training Program



- **Each RTI implementation provides tunable parameters**
- **The RID Editor provides access to the RTI 1.3 parameters**
- **The parameters include installation configuration, for example**
  - **Network interface**
  - **Performance tuning**
  - **Consistency checking**

Conclusion

- Processes, metrics and tools are needed to support user design of HLA federations to meet specific performance requirements of applications
  - The performance framework is the first step
  - Represents input from variety of users in developing HLA federations
  - It provides a structured and consistent way to assists the Federation execution planner
  - Provides a common frame of reference for the future development of tools, metrics, and federations
- In the following sections we will look at the practical application of the performance framework within select user's context

Outline

- **Describing a federation execution**

- **This section covers the Federation Execution Planner's Workbook (FEPW) and the FEPW Editor tool**
  - **FEPW information**
  - **FEPW Editor**

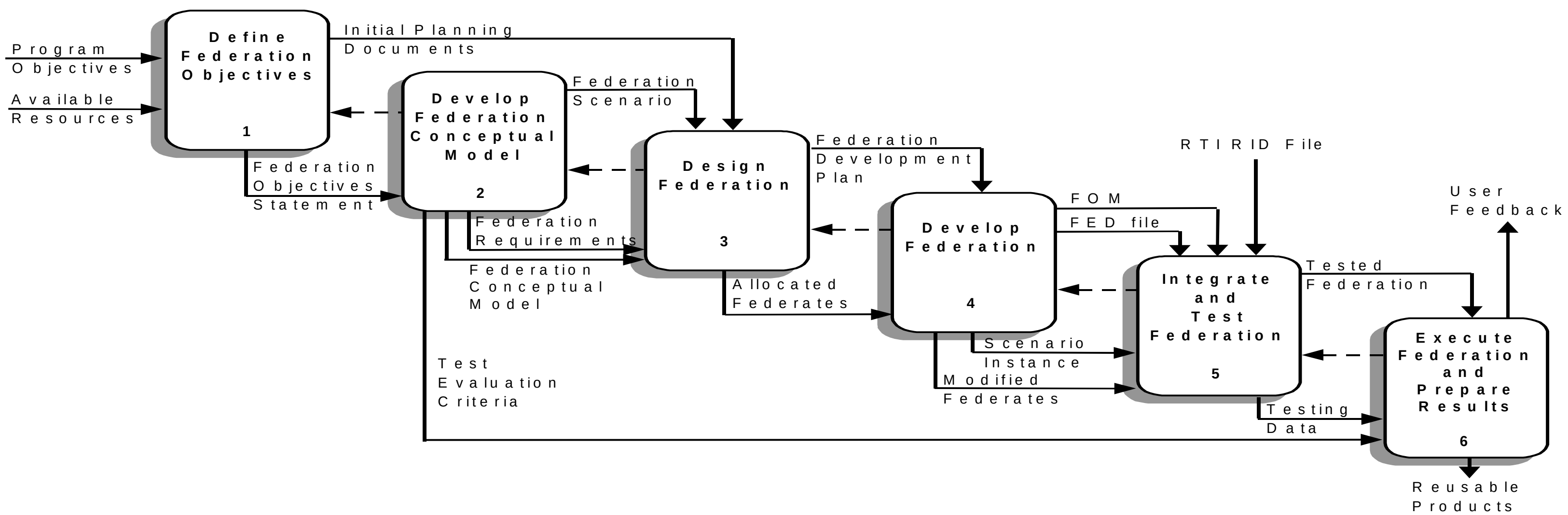
What Is It?

- Describing a federation execution is the process of understanding and documenting the unique characteristics of a specific federation that will affect overall federation performance requirements, network and computing requirements, and RTI requirements
- Why is describing a federation execution important?
  - Each Federation has unique characteristics that affect the planning, design, development, configuration, and execution of the federation and the overall performance of the system
  - Lessons Learned from similar Federations can be shared across the community

When Is It Done?

Integrated Training Program

- **First we must have a clear understanding of the Federation Objectives and the Conceptual Model, then we can begin to assess each of the federation constituents and general characteristics of the Federation Execution**
  - **This Assessment is accomplished in Steps 3&4 of the FEDEP**



What to Do? And How to Do It?

Integrated Training Program

- **The following activities will provide the information for Federation Developers to assess the federation and to analyze the performance requirements**
  - **Characterize the federation**
  - **Identify RTI services required**
  - **Define the computing & network environment**
  - **Allocate federates to computing environment resources**
- **The Federation Execution Planner's Workbook (FEPW) provides a framework for assisting the planning of HLA federation executions and mechanism for documenting the results of the Federation Assessment activities**

Federation Execution Planners Workbook

- The original purpose of the FEPW was to act as a means of gathering data on the performance needs of federation executions; it was intended to provide benchmark data to assess how well an HLA RTI needed to perform
- The FEPW now provides a framework for characterizing and assessing a Federation Execution
- There is a FEPW Tool available for free on the DMSO Software Distribution Center (SDC) at http:\\hla.dmso.mil

FEPW (Cont'd)

- **The FEPW can be used by Federation Developers for the following purposes:**
  - **Federation planning**
    - **Identification of the federates and their capabilities**
    - **Identification of computing environment**
    - **Define how federation objects, attributes, and interactions are distributed around the federation**
  - **Federation testing and execution**
    - **Validate expected federate performance in a federation execution**
    - **Facilitate gathering and logging of simulation data**
  - **Document characteristics of past federation executions**

Characterize the Federation

- **The 1st step in defining the performance requirements for a specific federation is to understand and properly characterize the Federation**
  - **Type of Federation (Analysis, Training, Test and Evaluation)**
  - **Federation Size (Number of Federates, Objects, Interactions, etc.)**
  - **Time synchronization requirements**
  - **Latency/Throughput requirements**
  - **etc.**

Types of Federations

Integrated Training Program

Summary Tables

NOTE: One table per Federation

Federation Execution Summary Table

Federation Execution Name		Sample	**Date of execution**	Jun-98
Application Domain		Training	**Version of Fed Ex**	1
Purpose		JTF and MSC	**Sponsor**	DMSO
Concurrent Federation Executions	**Number**	4	**Point of contact**	Mr. David Seidel
	Names	Test		MITRE
		Demo		703-883-6828
		Prototype		dseidel@mitre.org
RTI Software Used	**Vendor**	DMSO		
	Version	1.3	**FEPW last mod date**	15-May-98

- **Federations can typically be characterized by the general M&S domain they fall under. Each domain has different operating characteristics where different aspects of performance are most important**
  - **Analysis**
    - **Advance time as fast as possible**
    - **Repeatability of results**
    - **Tractability and correlation of results**
    - **Validity of results**

Types of Federations (Cont'd)

Integrated Training Program

- Training
  - Advance time with wall clock
  - Causality of events
  - Man In The Loop Simulators and C4I systems
  - Scalable number of federates and objects
- Test and Evaluation
  - Real-time requirements
  - Minimum latency
  - Hardware In The Loop

Characterize the Federates

- **Identify federates and their roles & responsibilities**
  - **Number of and types of federates**
  - **Object attributes and interactions each federate will publish/subscribe**
  - **Hardware and software requirements for each federate**
- **The FEPW provides a series of Tables to document the characteristics of each Federate in the Federation**
  - **Federate Summary Table**
  - **Object/Interaction Table**

Federate Summary Table

Integrated Training Program

- **This data identifies the federates and begins to add detail regarding how they operate in the federation execution**
  - **API used**
  - **Memory requirement**
  - **Tick and timing data**
  - **Host & LAN connected to**

Federate Summary Table

Fed	Name	API	Size (MB)	Tick			Time Management			Host	LAN
				Rate (/sec)	Min Value	Max Value	Regulating (y or n)	Constrained (y or n)	Lookahead		
1	Land_Simulation	Ada	10	10	0.001	0.005	Y	Y	1.00	1	1
2	Land_Simulation	Ada	15	10	0.001	0.005	Y	Y	1.00	2	1
3	Air_Sim_model	C++	8	50	0.001	0.005	Y	Y	1.00	3	2
3	Air_Sim_orders		10							7	2
4	Sea_Simulation	Java	7	20	0.001	0.005	Y	Y	1.00	4	2
5	Data_Logger	Java	1	20	0.001	0.005	N	N	1.00	5	2
6	Viewer	C++	1	10	0.001	0.005	N	Y	1.00	6	2
7	Intel bridge	C++	8	10	0.001	0.005	Y	Y	1.00	8	2
7	Intel bridge	C++	8	10	0.001	0.005	Y	Y	1.00	8	3
8	Intel sim	Ada	30	10	0.001	0.005	Y	Y	1.00	9	3

Federate Object Table

Integrated Training Program

- **This table describes the object attributes that each federate updates, reflects, transfers or accepts ownership of**
  - **One per federate**
  - **Attributes updated by the federate: How often & In what groupings**
  - **Attributes to which the federate subscribes: Latency constraints on updates**

Object/Interaction Tables

Federate	1	Land_Simulation

NOTE: One table per Federate

Object Table

Object Class	Attribute	Size (bytes)	Update									Reflect		Ownership		
			y/n	Count	Nominal Rate	Maximum Rate	Condi-tions	Group-ing	Transport (reliable or best effort)	Ordering (TSO or RO)	Routing Space	y/n	Max latency (msec)	T/A	Transfer Rate	Group-ing
Ground_unit				10												
	Attribute1	4	y		0.10	0.50		A	R	TSO	GRD	y	200	T/A	1/fedex	C
	Attribute2	4	y		0.01	0.10		B	R	TSO	GRD	n				
	Attribute3	4	y		0.10	0.50		A	R	TSO	GRD	y	150			
	Attribute4	4	y		0.01	0.10		B	R	TSO	GRD	n				
	Attribute5	4	y		0.10	0.50		A	R	TSO	GRD	y	300	T/A	1/fedex	C
	Attribute6	4	y		0.10	0.50		A	R	TSO	GRD	y				

Federate Interaction Table

Integrated Training Program

- **This table describes the interaction parameters that each federate initiates, senses, or reacts to**
  - **One per federate**
  - **Interactions generated by the federate: How often**
  - **Interactions to which the federate subscribes: Latency constraints**

Interaction Table

Interaction Class	Parameter	Size (bytes)	Initiate									Sense/React	
			y/n		Nominal Rate	Maximum Rate	Condi-tions	Group-ing	Transport (reliable or best effort)	Ordering (TSO or RO)	Routing Space	S/R	Max latency (msec)
Interaction1									R	TSO	GRD		
	Parameter1	4	y		0.03	0.03		D				S	150
	Parameter2	4	y		0.01	0.10		E					
	Parameter3	4	y		0.03	0.03		D				S	150
	Parameter4	4	y		0.01	0.10		E					
	Parameter5	4	y		0.03	0.03		D				S	150
	Parameter6	4	y		0.03	0.10		D				S	150

RTI Services Required

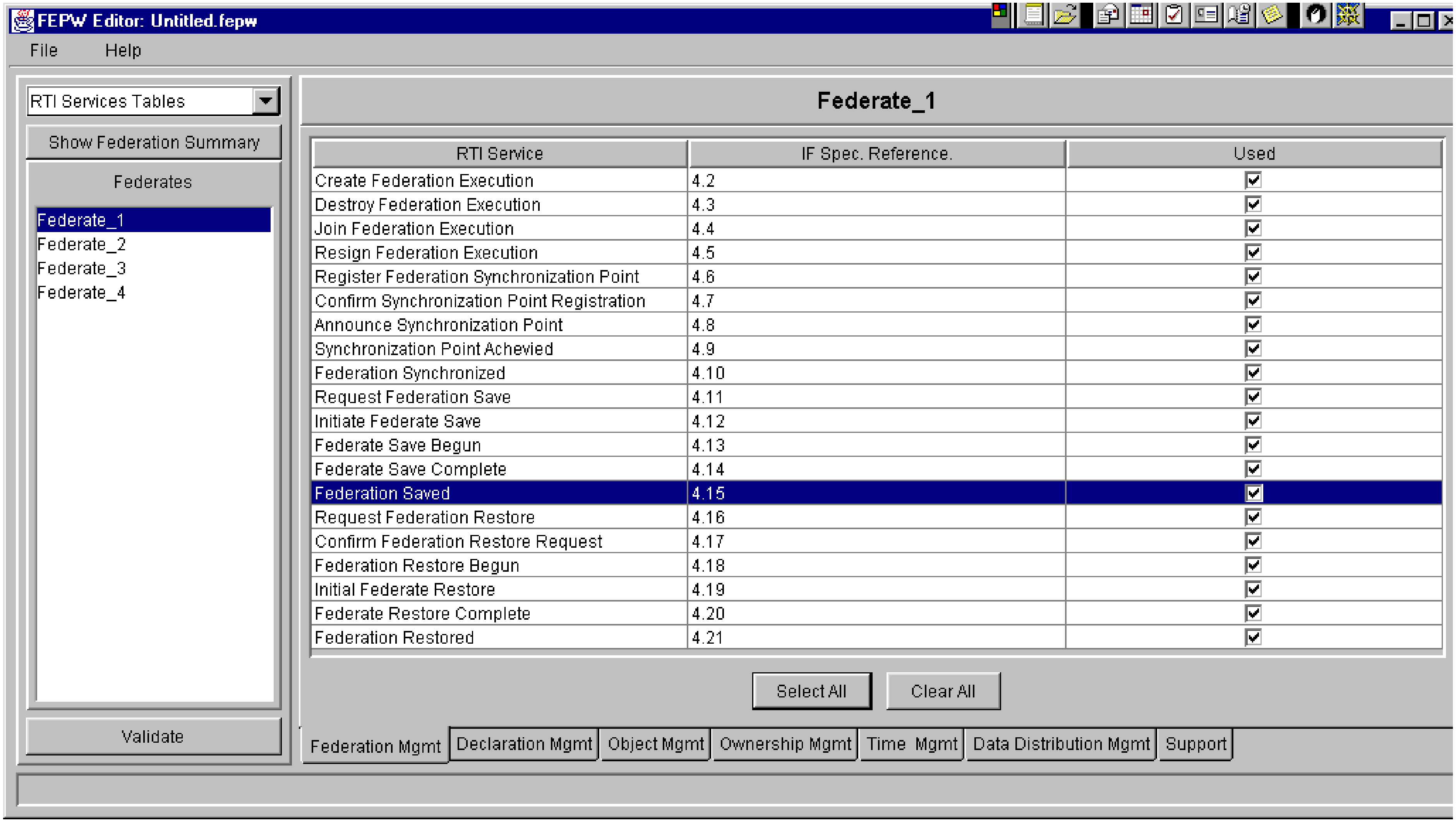
Integrated Training Program

- Identify services for Federation
  - Document services implemented by each federate
  - Agree to use of Time Management services
  - Evaluate additional services required to meet federation requirements
    - Ownership Management
    - Data Distribution Management

- The FEPW provides the RTI Services Table
  - One of these tables should be completed for each federate and a separate table represents the federation execution as a whole. A check next to a service indicates that the federate makes use of the service.

FEPW RTI Services Table

Integrated Training Program



RTI Service	IF Spec. Reference.	Used
Create Federation Execution	4.2	☑
Destroy Federation Execution	4.3	☑
Join Federation Execution	4.4	☑
Resign Federation Execution	4.5	☑
Register Federation Synchronization Point	4.6	☑
Confirm Synchronization Point Registration	4.7	☑
Announce Synchronization Point	4.8	☑
Synchronization Point Achevied	4.9	☑
Federation Synchronized	4.10	☑
Request Federation Save	4.11	☑
Initiate Federate Save	4.12	☑
Federate Save Begun	4.13	☑
Federate Save Complete	4.14	☑
Federation Saved	4.15	☑
Request Federation Restore	4.16	☑
Confirm Federation Restore Request	4.17	☑
Federation Restore Begun	4.18	☑
Initial Federate Restore	4.19	☑
Federate Restore Complete	4.20	☑
Federation Restored	4.21	☑

Define the Computing Environment

- **Now that we have a solid understanding of the Federates participating in the federation, it is important to define the Computing Environment and ensure that the necessary network and computer resources are available to meet the Federation Objectives and Performance Requirements**
- **Often this process is used to help define the required resources, but sometimes we need to document the resources already available and work towards allocating them as efficiently as possible**

Why Define the Computing Environment

- A clearly defined set of available resources combined with a set of federation operating requirements can be used to to proactively manage resources, mitigate risks, and identify performance issues
  - Determine if the required resources are available
  - Put together a set of equipment that must be purchased, leased, or borrowed
  - Determine if additional computers, memory, or disk space is required
  - Plan for backups and replacement equipment
  - Determine if the Network topology will support latency, throughput requirements, and multicast transmission
  - Document a description of the hardware and system software that is used in the federation execution

FEPW Host Table

Integrated Training Program

- **The FEPW provides a Host Table to describe the computers available/required for the federation execution**

Host Table

NOTE: One table per Federation

Host	Identification				Processor			Operating System		Memory (MB)		% CPU
	Name	LAN	Network Address	Location	Vendor/ Model	# CPUs	Speed (MHz)	Name	Version	Total (MB)	Available (MB)	Available to FedEx
1	Bubba	1	128.29.104.119	Reston VA	Sun Ultra 2	1	300	Solaris	2.5	192	150	75%
2	BigGuy	1	128.29.104.120	Reston VA	Sun Ultra 2	1	200	Solaris	2.5	192	150	60%
3	Eagle	2	128.29.102.101	San Diego	SGI Octane	1	300	IRIX	6.2	256	180	60%
4	Hawk	2	128.29.102.103	San Diego	Dell latitude CP	1	233	NT	4.0	64	40	70%
5	Pigeon	2	128.29.102.105	San Diego	Sun Ultra 2	1	250	Solaris	2.5	192	150	80%
6	CanGoose	2	128.29.102.107	San Diego	Compaq	1	233	NT	2.5	128	90	80%
7	Falcon	2	128.29.102.109	San Diego	SGI Octane	1	300	IRIX	6.2	256	180	80%
8	Wolverine	2	128.29.102.111	San Diego	SGI Octane	1	300	IRIX	6.2	256	180	80%
8	Wolverine	3	128.29.100.131	San Diego	SGI Octane	1	300	IRIX	6.2	256	180	80%
9	Badger	3	128.29.100.133	San Diego	Sun Ultra 2	1	300	Solaris	2.5	256	180	80%

FEPW LAN Tables

Integrated Training Program

- **The FEPW provides LAN Tables to describe the networks, including bandwidth and topology**

LAN Tables

LAN Description Table

LAN	Physical Type	Bandwidth	
	(Ethernet, ATM,etc.)	Specified	Available
1	Ethernet		
2	Ethernet		
3	Ethernet		
4			
5			
6			

NOTE: One table per Federation

LAN-to-LAN Connectivity Table

LAN	1		2		3		4		5	
1										
2	Device									
	Bandwidth									
	Latency									
3	Device		Device							
	Bandwidth		Bandwidth							
	Latency		Latency							
4	Device		Device		Device					
	Bandwidth		Bandwidth		Bandwidth					
	Latency		Latency		Latency					
5	Device		Device		Device		Device			
	Bandwidth		Bandwidth		Bandwidth		Bandwidth			
	Latency		Latency		Latency		Latency			
6	Device		Device		Device		Device		Device	
	Bandwidth		Bandwidth		Bandwidth		Bandwidth		Bandwidth	
	Latency		Latency		Latency		Latency		Latency	

Allocate Federates to Resources

Integrated Training Program

- **Now that we have documented the Federates in the Federate Table and documented the computing environment, we can begin the process of allocating resources**

ditor\sampledata\sample.fepw

Federates in Federation

Fed... ID#	Federate Name	RTI API	Memory Usage (MB)	Tick Rate (/sec)	Minimum Tick (sec)	Maximum Tick (sec)	Regulating	Constrained	Lookahead	Host Name	LAN Num...
1	Land Simulation	Ada	10	10	0.001	0.005	☑	☑	1	Bubba	1
2	Land Simulation	Ada	15	10	0.001	0.005	☑	☑	1	Big Guy	1
4	Sea Simulation	Java	7	20	0.001	0.005	☑	☑	1	Hawk	2
4	Air Sim orders		10				☑	☑		Falcon	2
5	Data Logger	Java	1	20	0.001	0.005	☐	☐	1	Pigeon	2
6	Viewer	C++	1	10	0.001	0.005	☐	☑	1	CanG...	2
7	Intel bridge	C++	8	10	0.001	0.005	☑	☑	1	Wolver...	2
7	Intel bridge		8				☑	☑		Big Guy	3
8	Intel sim	IDL	30	10	0.001	0.005	☑	☑	1	Badger	1

Conclusion

- In order to really understand the performance characteristics of a Federation Execution, we must first understand the operating characteristics and requirements of the specific federates and computing environment
- We now have the basis and foundation to begin experimenting with the computing environment, the RTI, and the federation execution to refine the initial planning data to ensure that we can meet the performance objectives

Outline

- **Measuring performance**

- **This section covers the benchmark programs**
  - **General rationale for benchmarks that are defined (major distributed functions of the RTI)**
  - **Define metrics that benchmarks measure**
  - **Describe how the measurements are made and justify**
  - **Provide an example of how/why you would use each benchmark**

Performance Benchmark Programs

- **Goals of the Benchmark Programs**
  - **Performance indicators for each of the major categories of inter-federate exchange through the RTI**
  - **Simple and unambiguous tools that can be applied by general users of the RTI**
  - **Easy to understand metrics that facilitate comparison and investigation of factors influencing federation performance**
  - **Source code that can be easily distributed and compiled on all RTI supported platforms**
  - **Benchmark programs that are parameterizable using simple command-line arguments and FED file modifications**

Performance Benchmarks

- Standard benchmark applications are delivered with the RTI software that can be configured and utilized for specific measurements
- Benchmark Suite
  - Throughput
  - Latency
  - Time Advancement
  - Ownership

Why Benchmarks?

- **Benchmarks are useful to test the <u>execution environment</u> that a federation will utilize**
  - **Measure performance of computing environment, network and the RTI implementation**
  - **Assess the upper limits of the environment**
  - **Benchmarks can be used as a first test to see if the computing environment and RTI implementation cannot meet the federations performance requirements**
- **Benchmark results should be presented along with any performance test results**
  - **Federation performance results can be viewed with respect to a known baseline for computing environment and RTI implementation**
  - **Can make more educated guesses about applicability of results to other architectures**

Using Benchmark Programs

Integrated Training Program

- Federation Performance is affected by various aspects of the federation/federate design, configuration and specific computing & network environment
  - Overhead of the distributed computing environment
  - Federate processing
  - The way tick() is called
  - etc.
- Benchmarks provide a baseline of data for the computing environment, but not for the federation specific operating conditions
- Performance benchmark tests should be accompanied by an FEPW to understand requirements and results
- Information on how to configure the RTI and optimize specific performance benchmarks will be covered in the next section "Configuring your Federation Execution"

Throughput Benchmark Program

- **Description: Measures maximum rate at which federates can exchange updates with no data loss**
- **Measurement Methodology**
  - **Measure the elapsed time to perform a specified number of Update Attribute Value service calls (calculates throughput from the federate to the RTI)**
  - **Measure the number of updates received by second federate to ensure no data loss**
  - **Results depend upon the average update sizes**

Throughput Benchmark Usage

BmThruput -h gives usage:

usage: BmThruput [OPTIONS]
OPTIONS:
[-f <Federate Name>]
[-F <Federation Name>]
[-d <debug level>]
[-s <Attribute Size>]
[-n <no. of feds>]
[-o <no. of local objs>]
[-c <no. of cycles>]
[-u <no. of updates>]
[-r] {indicates receiving federate}

An example use of BmThruput:
Fed1> BmThruput -n2 -c100 -u1000
Fed2> BmThruput -n2 -r

Latency Benchmark Program

Integrated Training Program

- Description: Measures elapsed time from attribute update to attribute reflect between 2 federates
- Measurement Methodology
  - First sending federate registers an object and updates its timestamp attribute
  - Second receiving federate reflects this update and updates its own object attribute with the received timestamp
  - When first federate receives this second update, it calculates the total latency
  - Average one-way latency is computed by dividing the total by 2
  - Accuracy of time synchronization mechanism and of two clocks on each computer is a key issue

Latency Benchmark Usage

Integrated Training Program

BmLatency -h gives usage:

usage: BmLatency [OPTIONS]
OPTIONS:
  [-f <Federate Name>]
  [-F <Federation Name>]
  [-d <debug level>]
  [-s <Attribute Size>]
  [-n <no. of feds>]
  [-o <no. of local objs>]
  [-u <no. of updates>]
  [-r]   {reflecting federate}

An example use of BmLatency:
  Fed1> BmLatency -n2 -u1000
  Fed2> BmLatency -n2 -r

Time Advancement Benchmark Program

- **Description: Measures rate at which a federation can grant time advancements**
- **Measurement Methodology**
  - **Federates enable time regulating and time constrained**
  - **Each federate calculates the elapsed time between each time request and grant by requesting time advance, then entering a loop to repeatedly tick() until time advance grant is received**
  - **Can be affected by lookahead values**
  - **Can be affected by configuration of reliable distributors**

Time Advancement Benchmark Usage

Integrated Training Program

BmTimeAdv -h gives usage:

usage: BmTimeAdv [OPTIONS]
OPTIONS:
  [-f <Federate Name>]
  [-F <Federation Name>]
  [-d <debug level>]
  [-n <no. of feds>]
  [-c <no. timestep cycles>]
  [-l <federate lookahead>]

An example use of BmTimeAdv:

Fed1> BmTimeAdv -n2 -c1000
Fed2> BmTimeAdv -n2 -c1000

Ownership Benchmark Program

- **Description: Measures rate at which two federates can transfer attribute ownership**
- **Measurement Methodology**
  - **Sending federate registers an object and pushes ownership of an attribute to receiving federate**
  - **Sending federate then pulls ownership of the attribute back from receiving federate**
  - **Total elapsed time for push/pull is calculated by the sender**

Ownership Benchmark Usage

BmOwnership -h gives usage:

usage: BmOwnership [OPTIONS]
OPTIONS:
[-f <Federate Name>]
[-F <Federation Name>]
[-d <debug level>]
[-n <no. of federates>]
[-t <no. of tranfers>]
[-r] {indicates receiving federate}

An example use of BmOwnership:

Fed1> BmOwnership -n2 -t100
Fed2> BmOwnership -n2 -r

Outline

- **Configuring your federation execution**
- **This section covers verifying the execution and configurations to optimize performance**
  - **Testing the federation execution to ensure federates are following the plan specified in FEPW**
  - **Configuration of the federation execution and RTI components for several use cases using the RID Editor:**
    - **WAN configuration (higher latency+lower bandwidth)**
    - **Low latency (need latency within certain bound)**
    - **High update rates (need to push a lot of data in federation)**

Configuring Your Federation Execution

- **The FEPW serves as a federation's default configuration**
  - Before optimizing a configuration, it should be verified that the federates are fulfilling their responsibilities as specified in the FEPW
  - The Federation Verification Tool can be used to assess whether the federates have implemented the plan
    - Verifies that each federate is meeting their data exchange responsibilities as specified in the FEPW
    - If the federation is not implemented as planned then the plan or federates should be changed as required
    - If the federation is implemented as planned and it is not performing well enough the following use cases may be of help

WAN Configuration

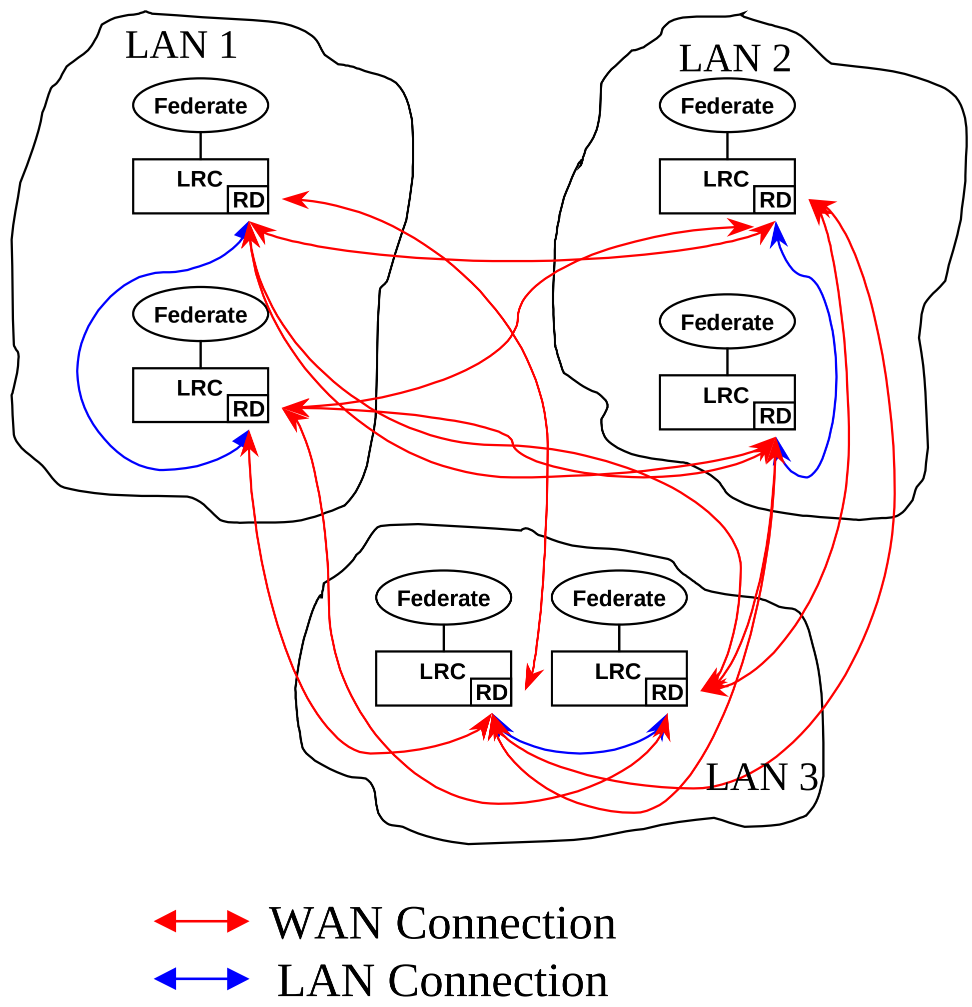
Integrated Training Program

- **Use Case**
  - **In this example we look at a federation with 3 geographically separated sites via an Internet based WAN connection**
    - **The federation is looking to reduce the bandwidth required by the federation**
  - **Considerations**
    - **Relocation of federates to LAN is not an option due to specialized equipment requirements, costs, etc.**
    - **Multicast is not currently supported by most Internet routers**
    - **WAN connections will typically have lower throughput and higher latency than LAN connections**
    - **Localizing traffic to LANs would reduce need to communicate over the WAN**

How Does It Work?

Integrated Training Program

- **Every reliable message is sent to each connection that is subscribed to that type of data**
  - **Object or interaction class**
- **Default configuration for reliable communications**
  - **Default is a completed graph where each federate connects with all other federates**
  - **6 federates with 2 federates per LAN = 24 WAN connections**
  - **Each federate has 4 WAN connections so 1 update can equal 4 WAN messages sent**



What Can We Do?

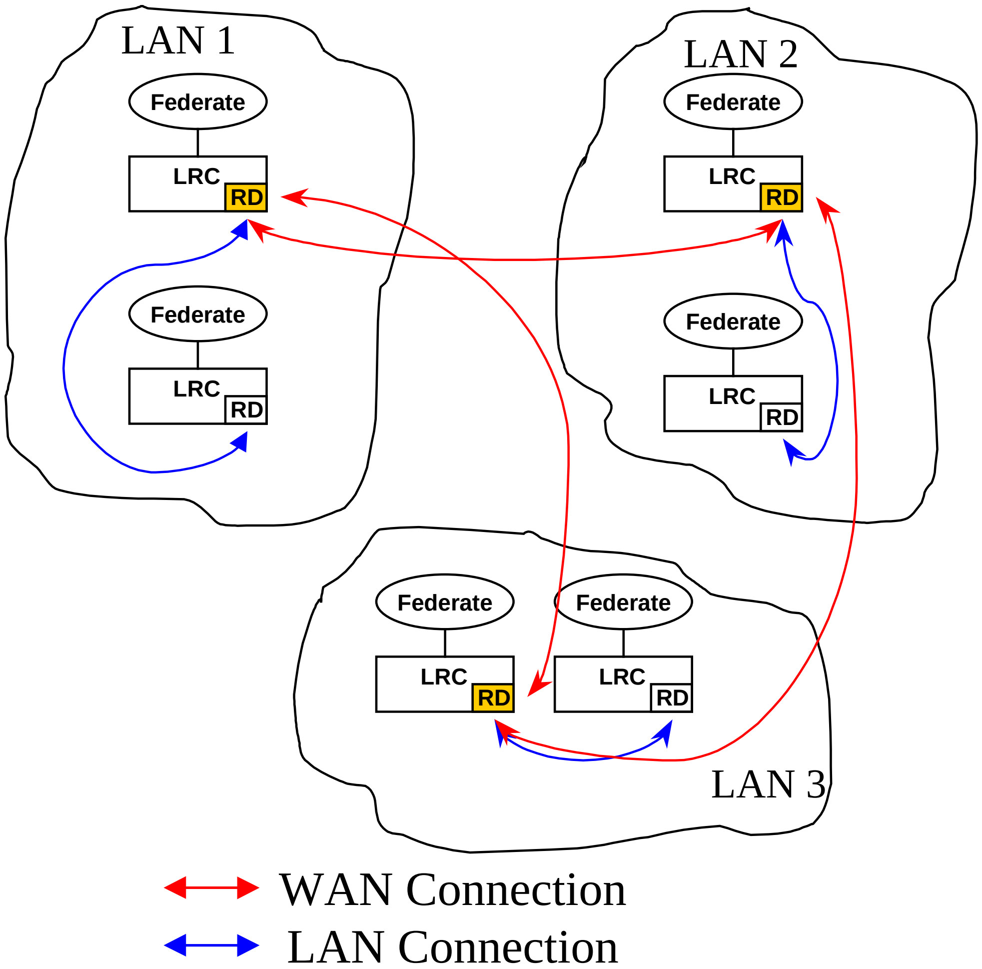
Integrated Training Program

- **Configuration steps to reduce WAN bandwidth requirements**
  - **Step 1: Reliable Distributor configuration**
    - **RID parameters can be used to configure routing of reliable traffic over the WAN**
      - **Reduces TCP/IP connections over WAN**
      - **Reduces packets sent over WAN**
      - **Does not require changes to FOM / federates**
      - **Requires changes to RID**
      - **Adds latency due to indirect routing (additional hops)**
  - **Step 2: Data Distribution Management Utilization**
    - **Routing spaces can be used to reduce unnecessary communications over WAN and LAN**
      - **Reduces packets sent over WAN and LAN**
      - **Requires changes to FOM, federates, and RID**

Step 1: Reliable Distributor Configuration

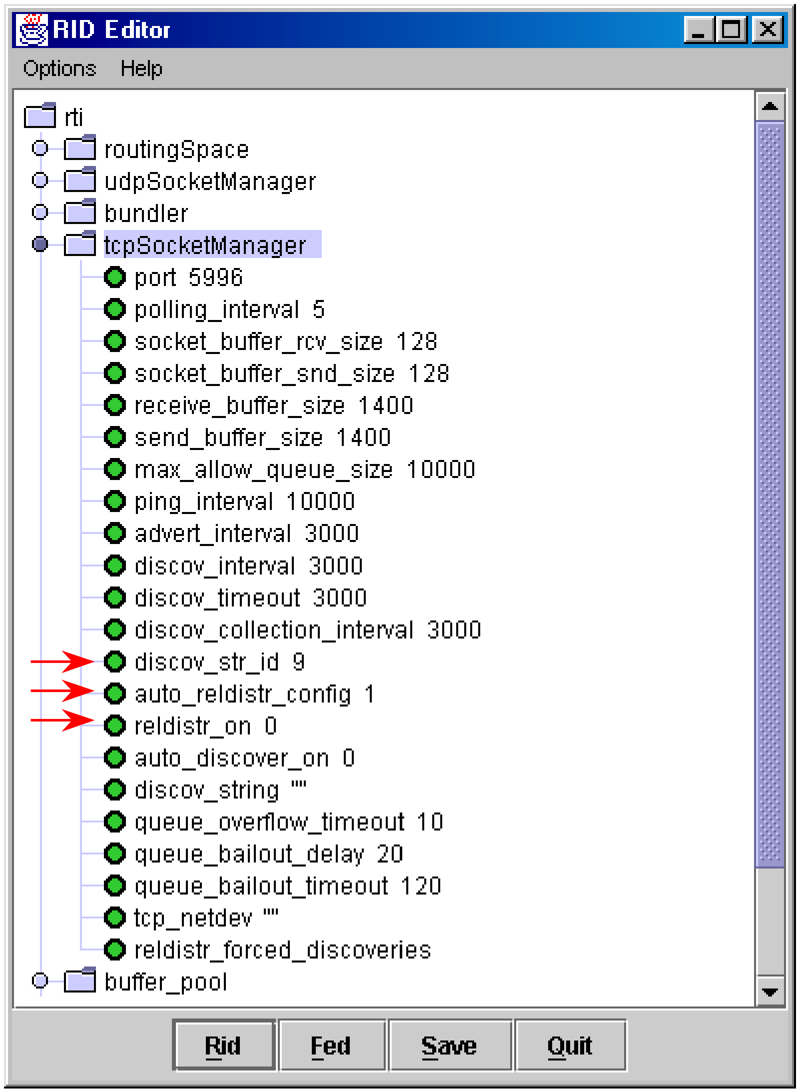
Integrated Training Program

- **Configure RID file for one reliable distributor per LAN**
  - **6 federates with 2 federates per LAN and 1 reliable distributor per LAN = 6 WAN connections**
  - **Each reliable distributor has 2 WAN connections so 1 update can equal 2 WAN messages sent**
  - **Messages sent by federates that are not acting as reliable distributors will incur an extra hop**



Step 1: Reliable Distributor Configuration

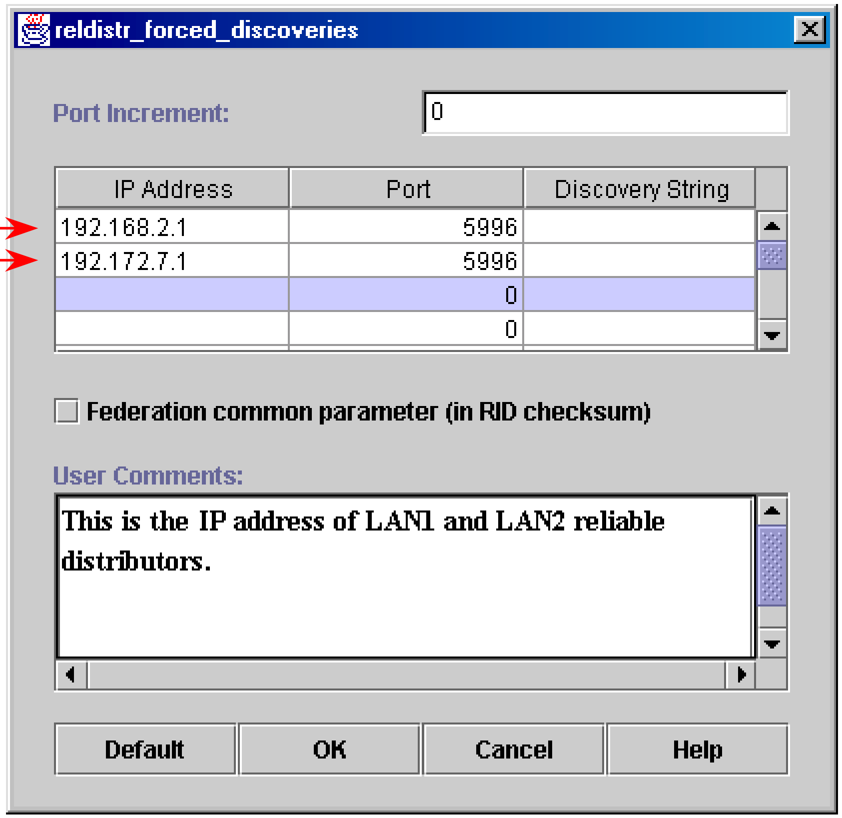
Integrated Training Program



- **We need 7 RID files to support this configuration**
  - **One for the rtiexec process**
  - **One for each LANs RD federate (3 total)**
  - **One for all federates on each LAN that are not RDs (1*3 LANs)**
- **RID file for rtiexec**
  - **Use the default RTI.rid file that comes with the RTI**
    - **Auto_reldistr_config=1**
    - **Reldistr_on=0**
    - **Auto_discover_on=0**
  - **It will auto-discover all the reliable distributors that are created**

Step 1: Reliable Distributor Configuration

Integrated Training Program



This example is for a RD federate's RID file.

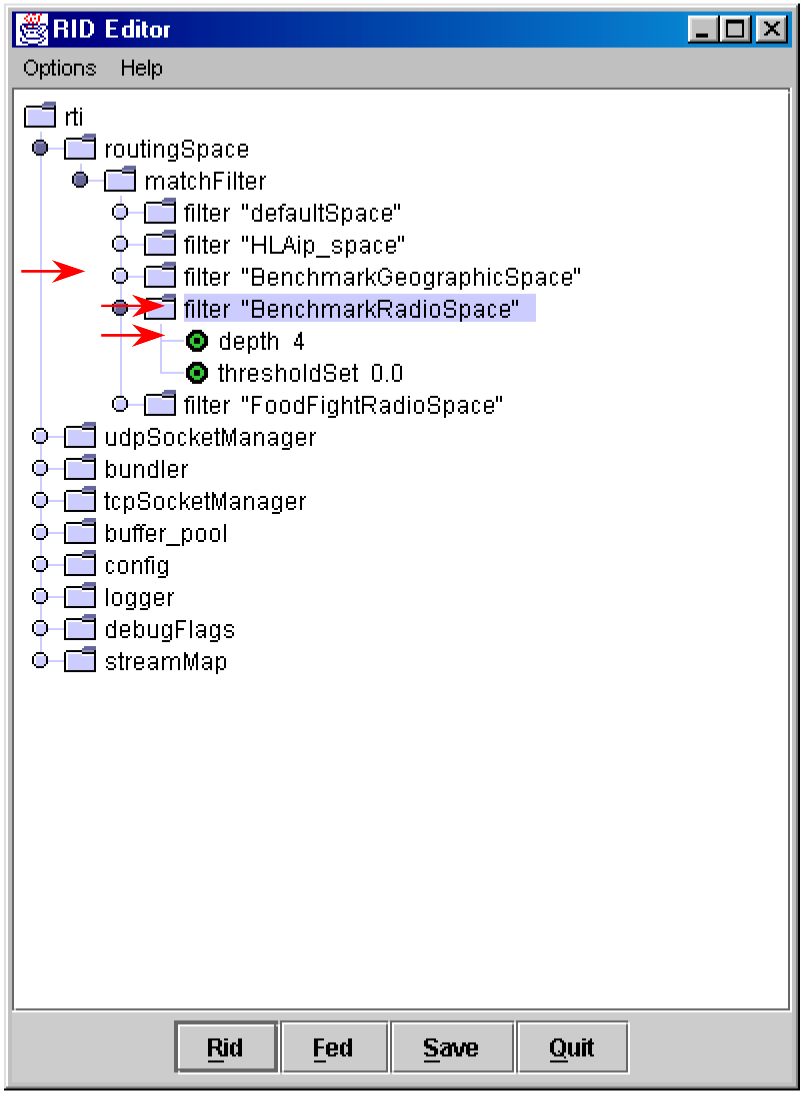
- A RID file for each RD federate on LAN
  - For all RD federate RID files change
    - Auto_reldistr_config=0
    - Reldistr_on=1
    - Auto_discover_on=1
  - Change reldistr_forced_discoveries values for each RD
    - Add IP address and port of RDs on other LANs
- A RID file for non-RD federates on LAN
  - For all non-RD federate RID files change
    - Auto_reldistr_config=0
    - Reldistr_on=0
    - Auto_discover_on=1
  - Change reldistr_forced_discoveries values for each LAN
    - Set IP address and port of RDs on federates' LAN

Step 2: DDM Utilization

- Every reliable message is sent to each connection that is subscribed to that type of data
  - Object or interaction class
- DDM can be used to improve the connection selection
  - Adds region intersection in addition to checks for object class and interaction class subscription
  - Levies additional requirements on federation
    - Adding routing spaces to FOM and RID
    - Use RTI service calls within federates for DDM
    - May not reduce WAN communications depending on scenario and subscriptions of federates

Step 2: DDM Utilization

Integrated Training Program



- RID modifications for DDM
  - Add filter space name & values
  - This example shows a federation defined filter space called ‘BenchmarkRadioSpace’
- Optimization of DDM routing
  - Not available with Reliable delivery
  - See later use case for additional optimizations when multicast is available

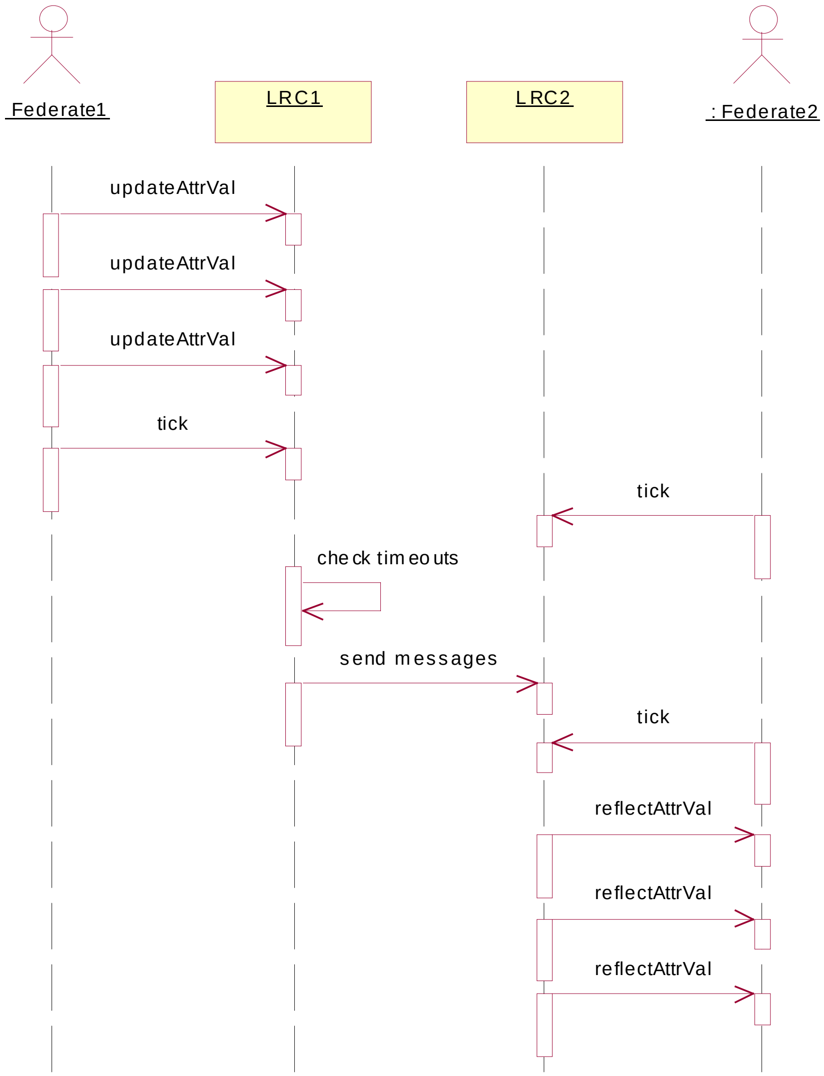
Low Latency

- **Use Case**
  - **In this example we look at a federation with a small number of hosts connected on a LAN. The federation is sending both reliable and best effort messages at a low rate (<1000/sec).**
    - **The federation is looking to reduce the average per message latency experienced by the federation**
  - **Considerations**
    - **LAN environment, small federation, and low update rate should not pose a problem for low latency**

How Does It Work?

Integrated Training Program

- **Each time a federate calls tick the RTI does stuff**
  - Determines whether there are messages ready to be sent
  - Reads the network
  - Does internal processing
  - Provides callbacks
- **UpdateAttrVals and sendInteractions are not sent upon each invocation**
  - Only after a subsequent tick or if a bundling timeout occurs during the invocation
- **Types of timeouts**
  - Bundling
  - Network polling

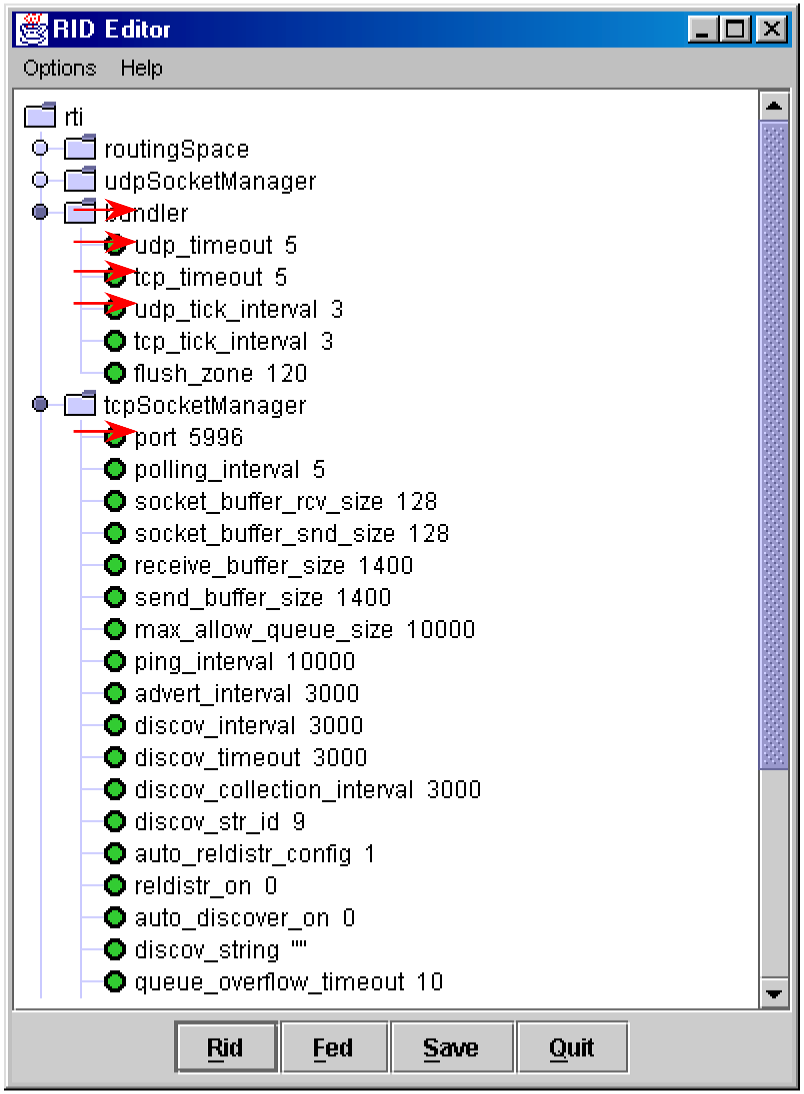


What Can We Do?

- Configuration steps to reduce average per message latency
  - Step 1: RID Configuration
    - Configure RID to reduce timeout values
      - TCP and UDP bundling
      - Network polling interval
    - Reducing timeout values will cause messages to be sent and read more frequently
  - Step 2: Tick approach
    - Alter the federate's internal process loop
      - Call tick more often
        - Gives the RTI the ability to send messages sooner
        - Gives the RTI the ability to read and reflect messages sooner

Step 1: RID Configuration

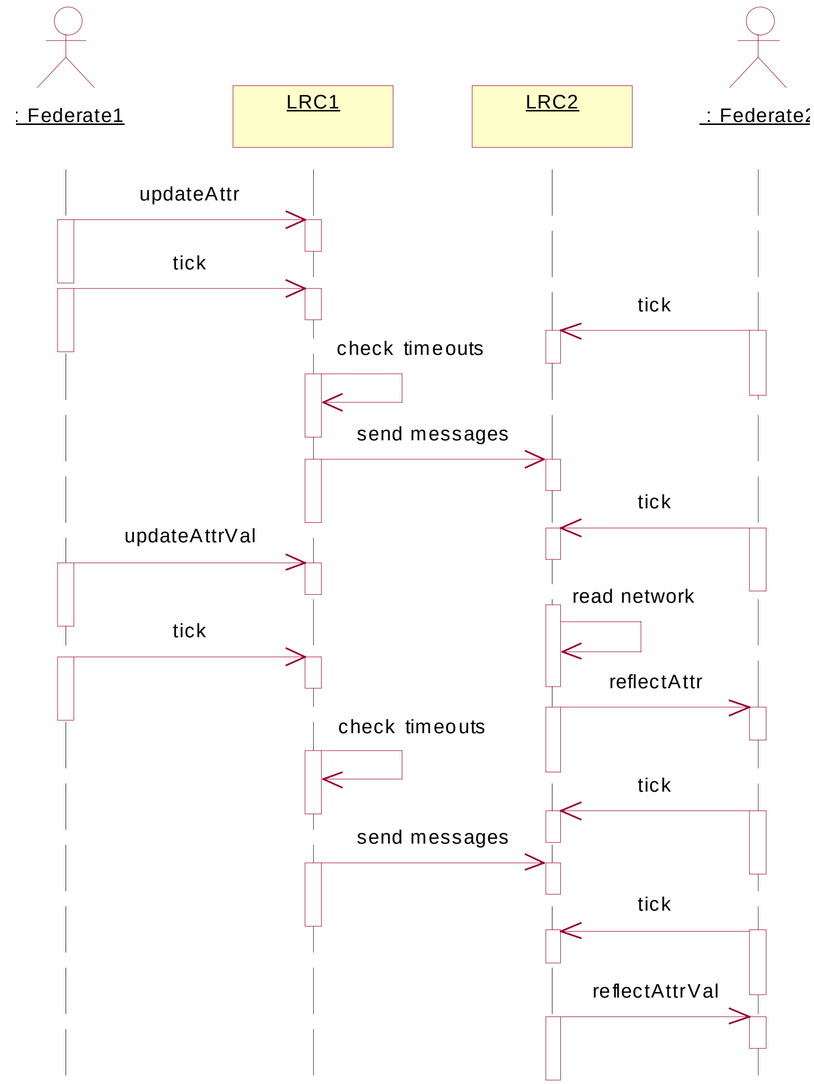
Integrated Training Program



- **Reduce bundler timeouts to zero effectively turns bundling off**
  - **upd_timeout=0 (ms)**
  - **tcp_timeout=0 (ms)**
- **Reducing tick intervals causes RTI to check timeouts more frequently**
  - **udp_tick_interval=1 (ms)**
  - **tcp_tick_interval=1 (ms)**
- **Reducing polling intervals causes RTI to read network more frequently**
  - **polling_interval=1 (ms)**

Step 2: Tick Approach

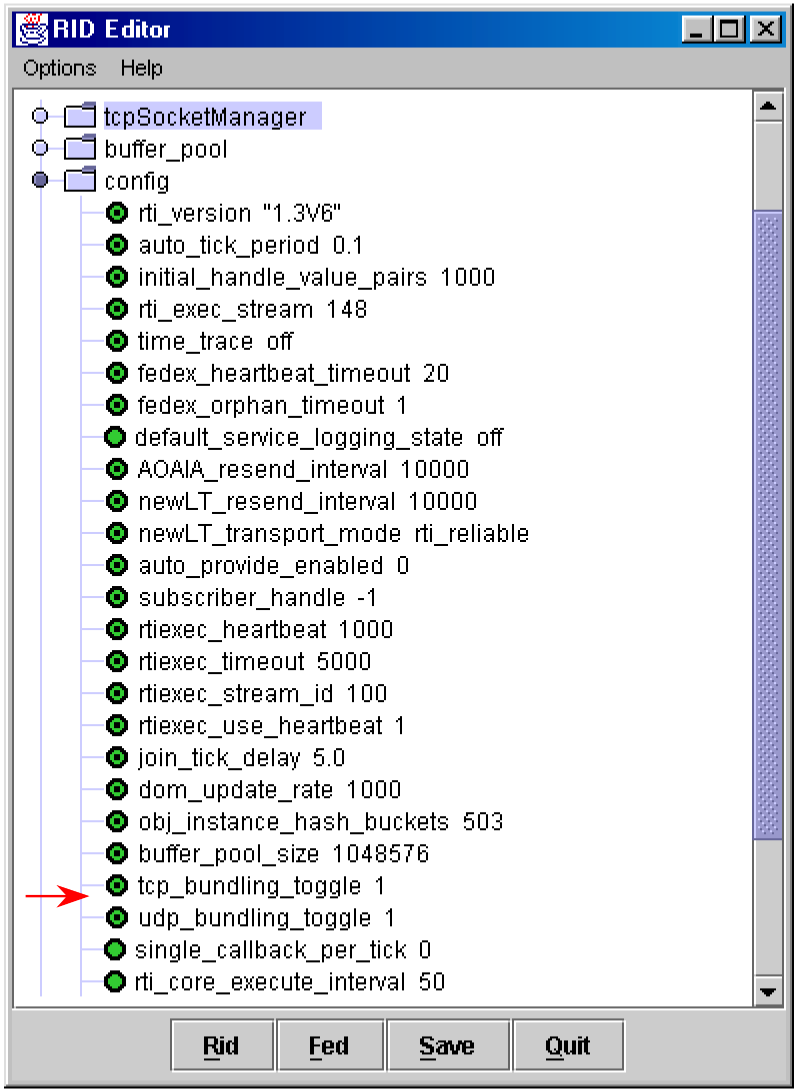
Integrated Training Program



- **While updating entity states call tick after each update**
  - **Use the tick() with no args**
  - **Modify RID value to limit the number of callbacks upon tick() - see next slide**
- **After processing current frame drain all pending callbacks from RTI**
  - **Call tick within a loop until tick returns FALSE (0)**

Step 2: Tick Approach

Integrated Training Program



- **Tick is called while state is being updated to decrease per message latency**
- **There is not a way to prevent callbacks when you call tick**
- **Modify RID to limit the number of callbacks to one per tick**
  - **single_callback_per_tick=1**

High Update Rate

- Use Case
  - In this example we look at a federation with a large number of hosts connected on a LAN. The federation is sending reliable messages at a low rate and best effort messages at a high rate (>1000/sec).
    - The federation is looking to increase the throughput (reduce dropped packet rate) experienced by the federation
  - Considerations
    - Multicast is supported by most LANs
    - LAN device will typically support higher throughput than computer NICs
    - Need to reduce number of packets received at each host
    - Since reliable messages are infrequent the default reliable distributor configuration is fine

How Does It Work?

- The RTI uses destination based routing to determine which federates should receive each message
  - For each message sent the RTI determines the set of federates that want the message based on subscriptions
  - It then compares the set of federates to a pre-allocated set of destination vectors defined in the RID
  - The destination vectors defined in the RID map to unique multicast groups
  - The LRC of a federate subscribes to the multicast groups that correspond to the destination vectors that it is a member
  - The network infrastructure filters the messages that are not sent to the destinations
    - Router, switches, network interface cards
    - These devices DO have a limit on the number of mcast groups they can support

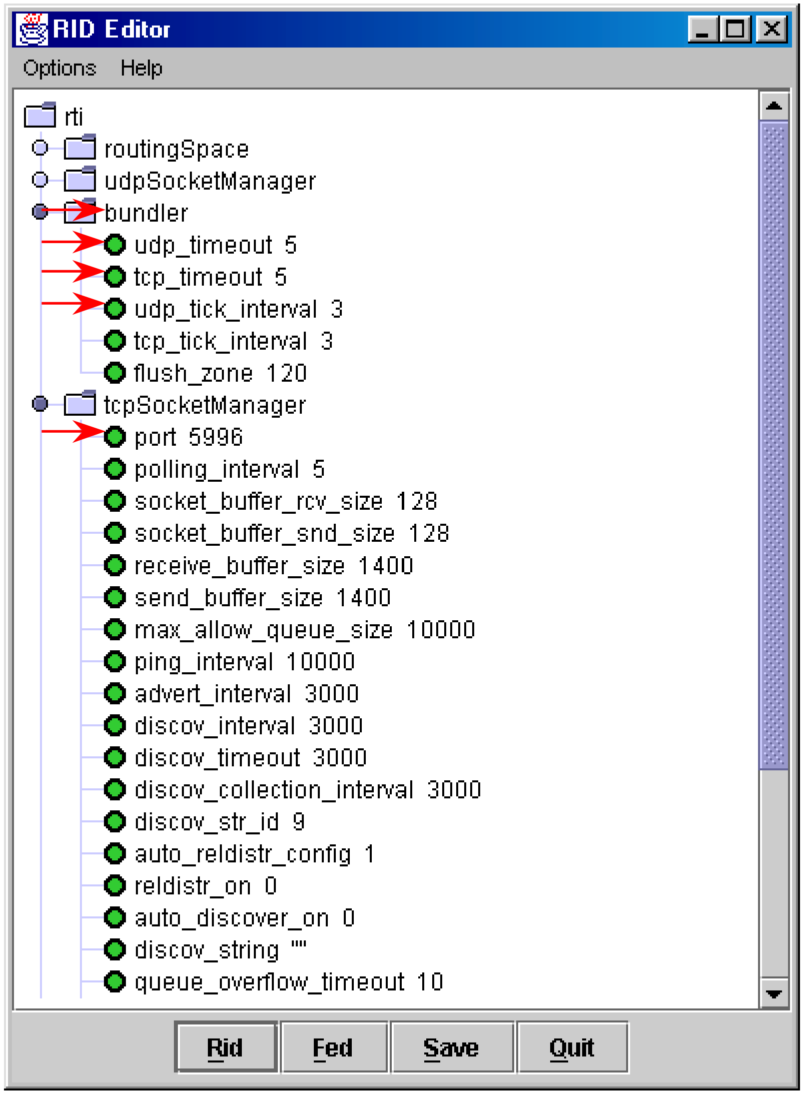
What Can We Do?

Integrated Training Program

- **Configuration steps to increase throughput and reduce dropped packet rate**
  - **Step 1: RID Configuration**
    - **Turn bundling on to decrease # of packets sent**
    - **Provide optimal set of destination vectors**
    - **Specify each federates position in the vector**
  - **Step 2: Data Distribution Management Utilization**
    - **Routing spaces can be used to improve destination vector selection**
      - **Requires changes to FOM, federates, and RID**

Step 1: RID Configuration

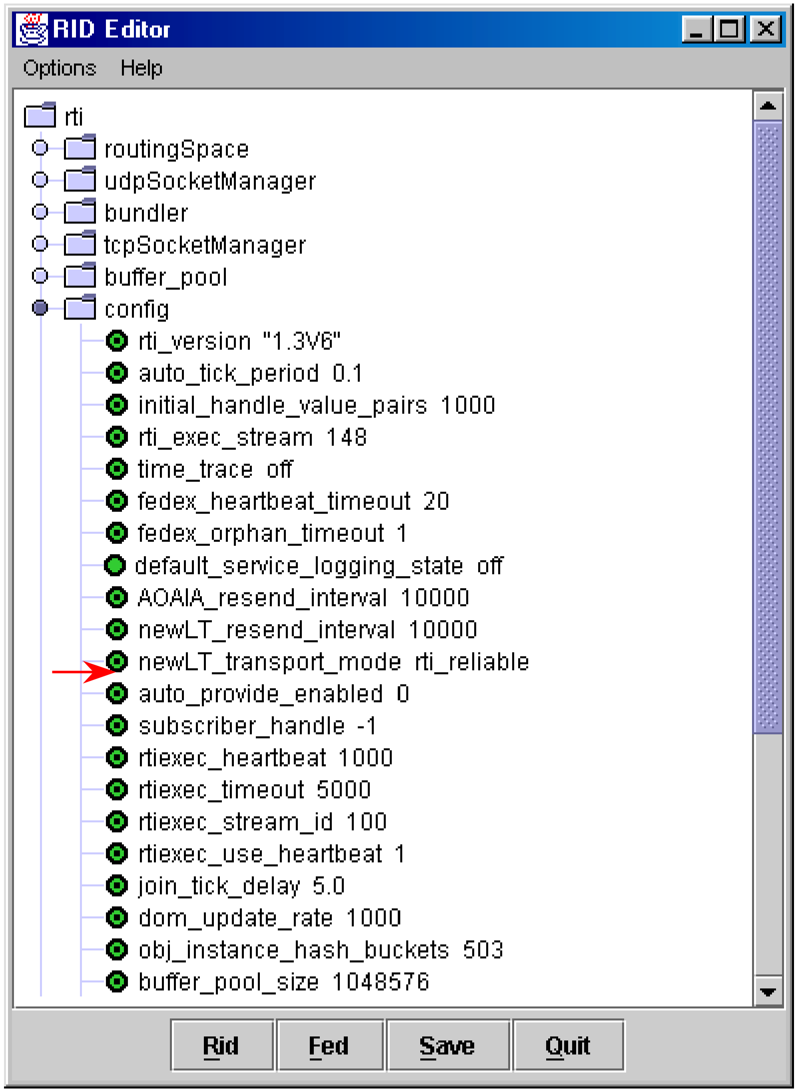
Integrated Training Program



- **Turn on bundling so that RTI sends fewer packets (although larger) over the network**
  - **This reduces number of received packets and thus kernel interrupts**
- **Increase bundler timeouts based on federation's latency tolerances**
  - **upd_timeout=50 (ms)**
  - **tcp_timeout=50 (ms)**

Step 1: RID Configuration

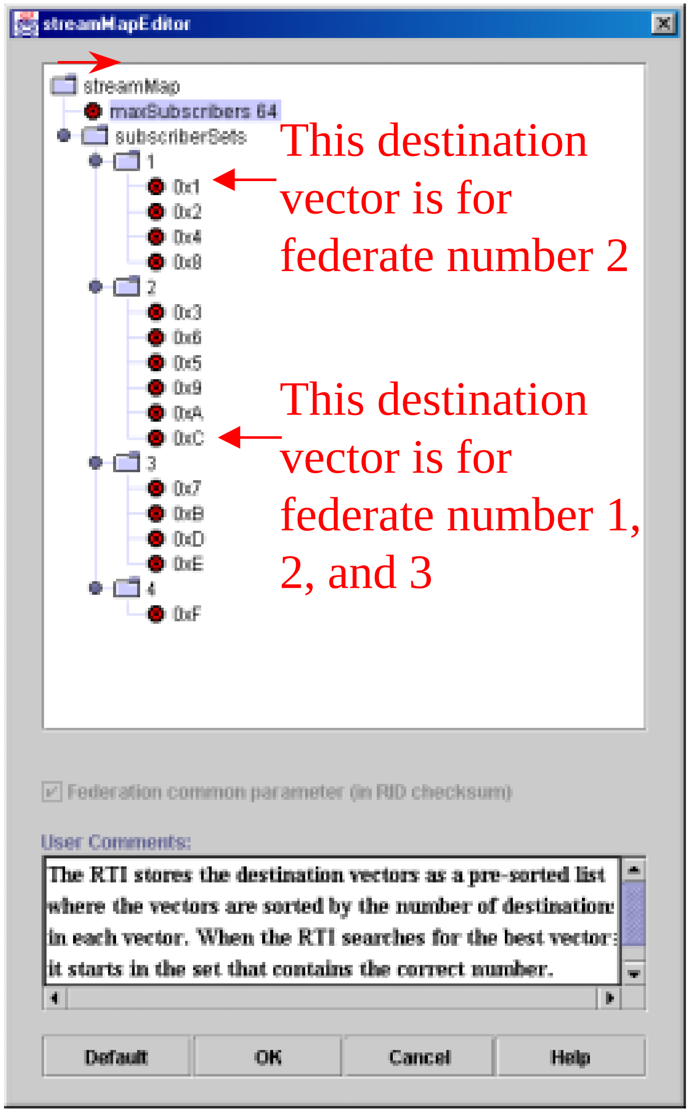
Integrated Training Program



- **Since the RTI routes the data based on the destinations that are interested in it the RTI needs to have a unique value for each federate (federate handle)**
  - **By default the RTI automatically assigns the federate handle**
  - **For us to plan the optimal set of destination vectors we need control of the handle assignment**
  - **Set the subscriber_handle parameter to a positive unique value for each federate**
  - **Requires a unique RID file for each federate**

Step 1: RID Configuration

Integrated Training Program



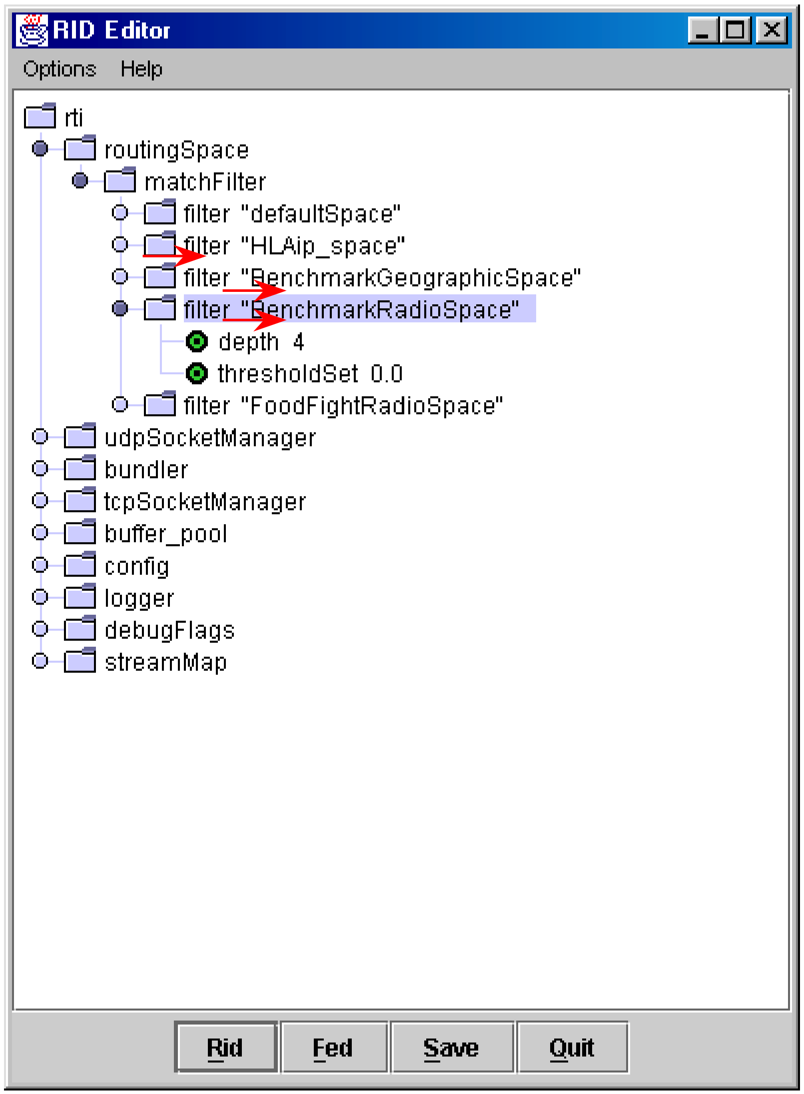
- **Determine the number of multicast groups supported by your hardware**
  - **Set maxSubscribers to that number minus 3**
- **Analyzing the FEPW for your federation**
  - **Determine objects and interactions with highest update rates**
  - **Determine subscription set of federates for those data items**
  - **Create binary vector that contains 1's in the bit position for each of the subscribing federate's handle value (rightmost bit = 1)**
  - **Covert the binary vector value to hexadecimal value and insert into the appropriate subscriber set**
  - **Repeat for less frequently updated data until number of destination vectors equals maxSubscribers**

Step 2: DDM Utilization

- **DDM can be used to improve the destination vector selection**
  - **Adds region intersection in addition to checks for object class and interaction class subscription**
  - **Levies additional requirements on federation**
    - **Adding routing spaces to FOM and RID**
    - **Use RTI service calls within federates for DDM**
    - **May not reduce number of packets received at each host depending on scenario, subscriptions of federates, and network hardware characteristics**

Step 2: DDM Utilization

Integrated Training Program



- **RID modifications for DDM**
  - **Add filter space name & values**
    - **ThresholdSet is not used by RTI anymore but - the parser expects it to be there**
    - **Depth is the maximum depth of the tree data structures used to store regions inside the RTI (see RIDeditor Help files)**
  - **This example shows a federation defined filter space called 'BenchmarkRadioSpace'**
- **Optimization of DDM routing**
  - **Need to reevaluate the set of destination vectors that were defined in Step 1**
  - **Optimal set of destination vectors is probably different at times in execution - requires more multicast groups to have more destination vectors**