EECS123:

Introduction to Real-Time Distributed Programming

Config.ini File and GUI

UCI DREAM Lab

Outline

- Introduction to "config.ini"
- Configuration of "config.ini"
  - Master node
  - TMO network configuration manager (TNCM)
  - Clock synchronization
  - Virtual machine timeslice
  - Real-time multicast and memory-replication channels (RMMC)
  - Virtual machine for main application thread (VMAT)
  - Reliable messaging protocol (RMP)
  - Miscellaneous facilities
  - Middleware internals

UCI DREAM Lab

config.ini

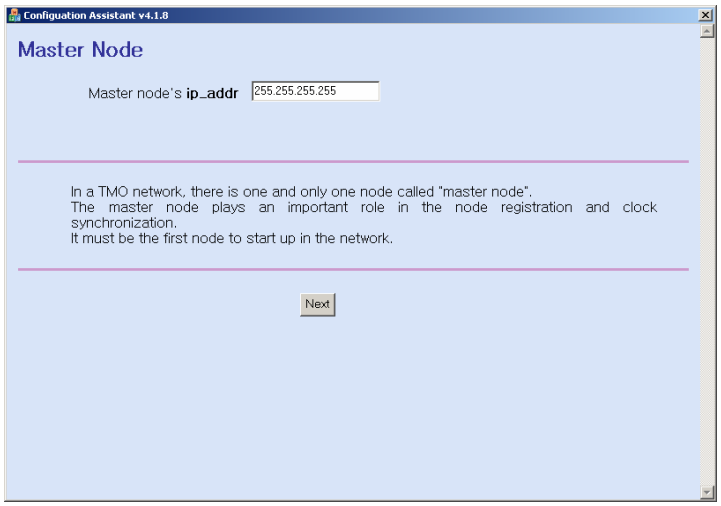
- Before running their TMO programs, application users have to specify several parameters related to the system configuration, middleware configuration, and network topology via "config.ini" file.
- The config.ini file must exist in the same directory where executable files reside, e.g., MyProject₩Debug.
- TMOToolkit provides a GUI, "TMO Configuration Assistant", for specifying those parameters and for generating the corresponding config.ini file based on the values of those parameters given by application users.
- This TMO Configuration Assistant also provides explanation of each parameter and a default value for it, whenever applicable.

config.ini: Start page

- Snapshot of TMO Configuration Assistant: Start page

config.ini: Master Node

- Snapshot of TMO Configuration Assistant: Master Node configuration

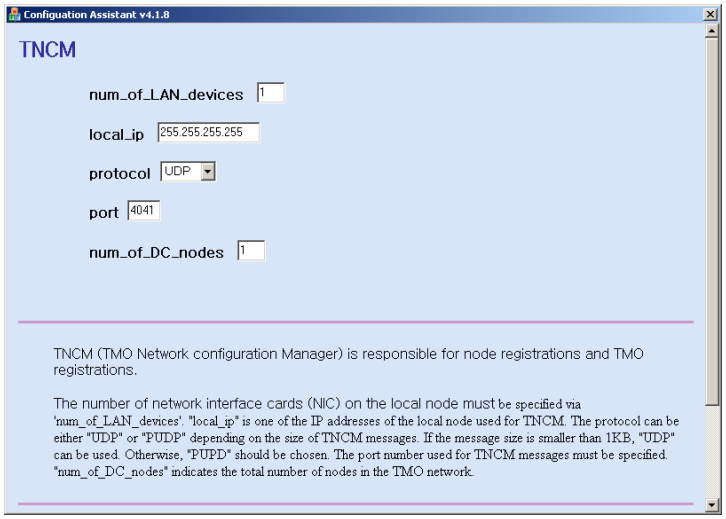


config.ini: Master Node

```
# This is a configuration file for TMOSM.
#
# In a TMO network, there is one and only one node called “master node”.
# The master node plays an important role in the node registration and clock
# synchronization. It must be the first node to start up in the network.
[master_node]
ip_addr = 128.195.164.50
```


config.ini: TNCM

- Snapshot of TMO Configuration Assistant: TNCM configuration

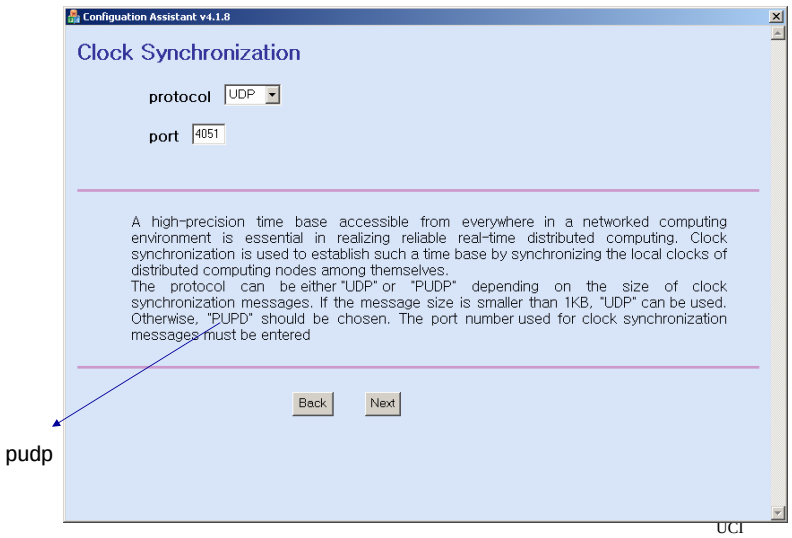


config.ini: TNCM

```
# TNCM (TMO Network Configuration Manager) is responsible for node
# registration and TMO registrations.  The number of network interface cards
# (NIC) in the local node must be specified via "num_of_LAN_devices"
# parameter. "local_ip" is one of the IP addresses of the local node used for
# TNCM. The protocol can be either "UDP" or "PUDP" depending on the size
# of TNCM messages. If the message size is smaller than 1KB, "UDP" can
# be used. Otherwise, "PUPD" should be chosen. The port number used for
# TNCM messages must be specified. "num_of_DC_nodes" indicates the total
# number of nodes in the TMO network.
[tncm]
num_of_LAN_devices = 1
local_ip = 128.195.164.50
protocol = UDP
port = 4041
num_of_DC_nodes = 1
```


config.ini: Clock Synchronization

- Snapshot of TMO Configuration Assistant: Clock Sync configuration

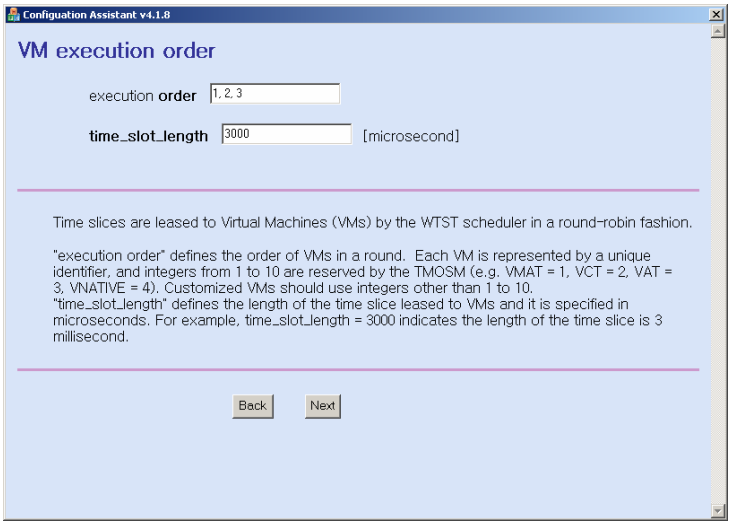


config.ini: Clock Synchronization

```
# A high-precision time base accessible from everywhere in a networked
# computing environment is essential in realizing reliable real-time
# distributed computing. Clock synchronization is used to establish such a
# time base by synchronizing the local clocks of distributed computing nodes
# among themselves. The protocol can be either "UDP" or "PUDP" depending
# on the size of clock synchronization messages. If the message size is
# smaller than 1KB, "UDP" can be used. Otherwise, "PUPD" should be chosen.
# The port number used for clock synchronization messages must be entered.
[clock_sync]
protocol = UDP
port = 4051
```


config.ini: Virtual Machine Timeslice

- Snapshot of TMO Configuration Assistant: VM TS configuration

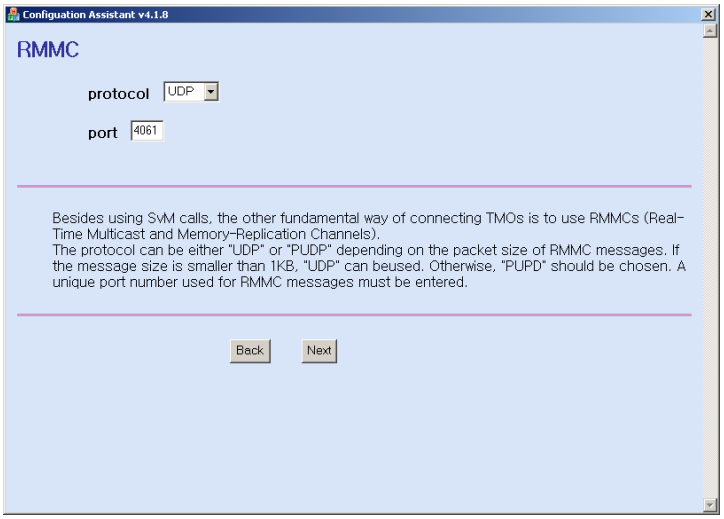


config.ini: VM Timeslice

```
# Time slices are leased to VMs by the WTST in a round-robin fashion.
# "order" defines the order of VMs in a round.  Each VM is represented by a
# unique identifier, and integers from 1 to 10 are reserved by the TMOSM
# (e.g. VMAT = 1, VCT = 2, VAT = 3,  VNATIVE = 4).
# Customized VMs should use integers other than 1 to 10.
# "time_slot_length" defines the length of the time slice leased to VMs and it
# is specified in microseconds. For example, time_slot_length = 3000
# indicates the length of the time slice is 3 millisecond.
[VM_exec_order]
order = 1, 2, 3
time_slot_length = 3000
```


config.ini: RMMC

- Snapshot of TMO Configuration Assistant: RMMC configuration

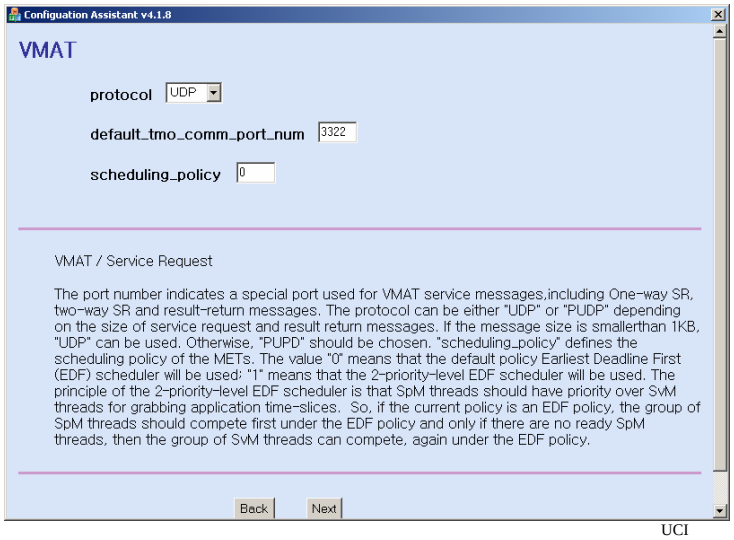


config.ini: RMMC

```
# Besides using SvM calls, the other fundamental way of connecting TMOs is
# to use RMMCs (Real-Time Multicast and Memory-Replication Channels).
# The protocol can be either "UDP" or "PUDP" depending on the size of
# RMMC messages. If the message size is smaller than 1KB, "UDP" can be
# used. Otherwise, "PUPD" should be chosen. A unique port number used for
# RMMC messages must be entered.
[rmmc]
protocol = PUDP
port = 4061
```


config.ini: VMAT

- Snapshot of TMO Configuration Assistant: VMAT configuration

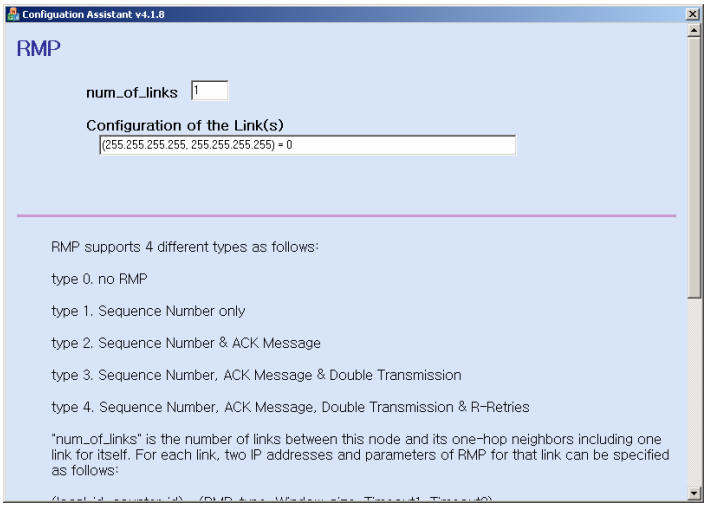


config.ini: VMAT

```
# The port number indicates a special port used for VMAT service messages,
# including One-way SR, two-way SR and result-return messages. The
# protocol can be either "UDP" or "PUDP" depending on the size of service
# request and result return messages. If the message size is smallerthan 1KB,
# "UDP" can be used. Otherwise, "PUPD" should be chosen.
# "scheduling_policy" defines the scheduling policy of the METs. The value "0"
# means that the default policy Earliest Deadline First (EDF) scheduler will be
# used; "1" means that the 2-priority-level EDF scheduler will be used. The
# principle of the 2-priority-level EDF scheduler is that SpM threads should
# have priority over SvM threads for grabbing application time-slices.  So, if
# the current policy is an EDF policy, the group of SpM threads should compete
# first under the EDF policy and only if there are no ready SpM threads, then
# the group of SvM threads can compete, again under the EDF policy.
[vmst]
default_tmo_comm_port_num = 3322
protocol = PUDP
scheduling_policy = 0
```


config.ini: Reliable Messaging Protocol

- Snapshot of TMO Configuration Assistant: RMP configuration



config.ini: RMP

```
# Reliable Messaging Protocol (RMP) provides a reliable message
# transmission over UDP/IP by utilizing double transmission and
# acknowledgement. Between any pair of nodes, a logical link can be
# established and its RMP parameters must be configured.

# RMP supports 4 different types as follows:
#   type 0. no RMP
#   type 1. Sequence Number only
#   type 2. Sequence Number & ACK Message
#   type 3. Sequence Number, ACK Message & Double Transmission
#   type 4. Sequence Number, ACK Message, Double Transmission
#                    & R-Retries
[rmp]
num_of_links = 2
(128.195.164.50, 128.195.164.50) = 0
(128.195.164.50, 128.195.164.55) = (1, 10, 1000)
```

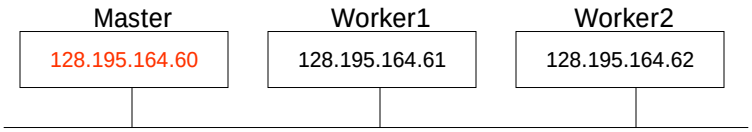

config.ini: RMP (cont.)

```
# "num_of_links" is the number of links between this node and its one-hop
# neighbors including one link for itself. For each link, two IP addresses and
# parameters of RMP for that link can be specified as follows:
#
#    (local_id, neighbor_id) = (RMP_type, Window_size, Timeout1, Timeout2)
#
# local_id is one of the IP addresses of the node;
# neighbor_id is the IP address of the other party;
# RMP_type can be '0' for no RMP,  '1' for RMP type 1 or '2' RMP type 2;
# Window_size is the number of messages that can be buffered  by the
# receiver;
# For RMP type 1, Timeout1 corresponds to the amount of time for the
# receiver to wait until the receiver decides that some messages are lost in
# case of out-of-order delivery, and Timeout 2 is not used;
# For RMP type 2, Timeout1 corresponds to the amount of time for the
# sender to wait until ack  messages are received, and Timeout 2 for the
# receivers to wait until the accumulated ACK messages are sent.
```

UCI
DREAM Lab

Connections among Nodes: Physical

Physical Connection



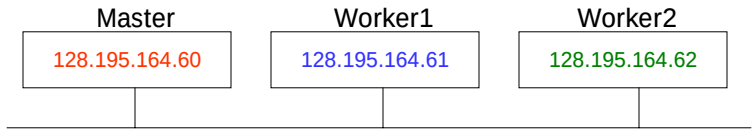
Master, Worker1, and Worker2

```
#config.ini
#
[master_node]
ip_addr = 128.195.164.60
#
#
........
```


UCI
DREAM Lab

Connections among Nodes: TNCM

TNCM



Master

```
......
[tncm]
num_of_LAN_devices = 1
local_ip = 128.195.164.60
protocol = UDP
port = 4041
num_of_DC_nodes = 3
#
......
```

Worker1

```
......
[tncm]
num_of_LAN_devices = 1
local_ip = 128.195.164.61
protocol = UDP
port = 4041
num_of_DC_nodes = 3
#
......
```

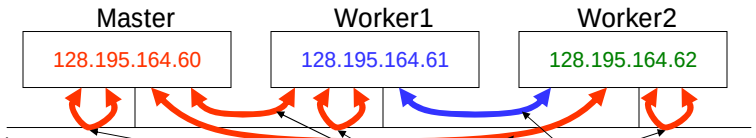
Worker2

```
......
[tncm]
num_of_LAN_devices = 1
local_ip = 128.195.164.62
protocol = UDP
port = 4041
num_of_DC_nodes = 3
#
......
```


UCI
DREAM Lab

Connections among Nodes: Logical

RMP and Logical Connections



Master

```
......
[rmp]
num_of_links = 3
(128.195.164.60, 128.195.164.60) = 0
(128.195.164.60, 128.195.164.61) = 0
(128.195.164.60, 128.195.164.62) = 0
......
```

mandatory connection

optional connection depending on the application scenario

Worker1

```
......
[rmp]
num_of_links = 3
(128.195.164.61, 128.195.164.61) = 0
(128.195.164.61, 128.195.164.60) = 0
(128.195.164.61, 128.195.164.62) = 0
......
```

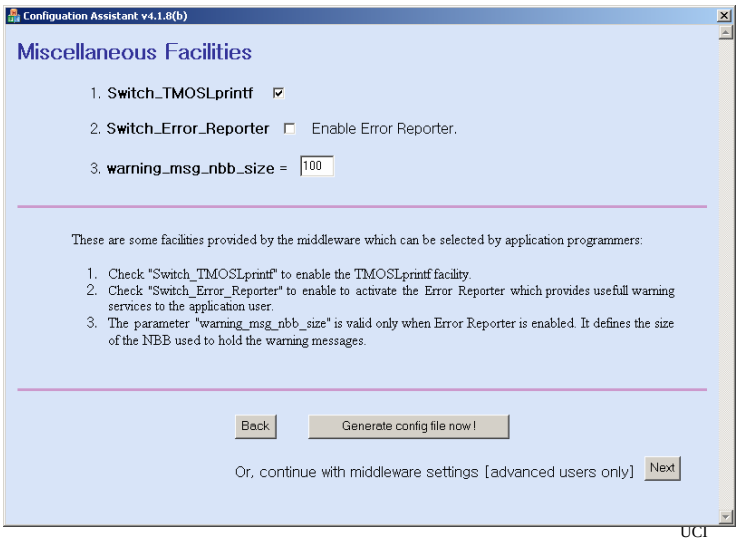
Worker2

```
......
[rmp]
num_of_links = 3
(128.195.164.62, 128.195.164.62) = 0
(128.195.164.62, 128.195.164.60) = 0
(128.195.164.62, 128.195.164.61) = 0
......
```


UCI
DREAM Lab

config.ini: Miscellaneous Facilities

- Snapshot of TMO Configuration Assistant: Miscellaneous



config.ini: Miscellaneous Facilities

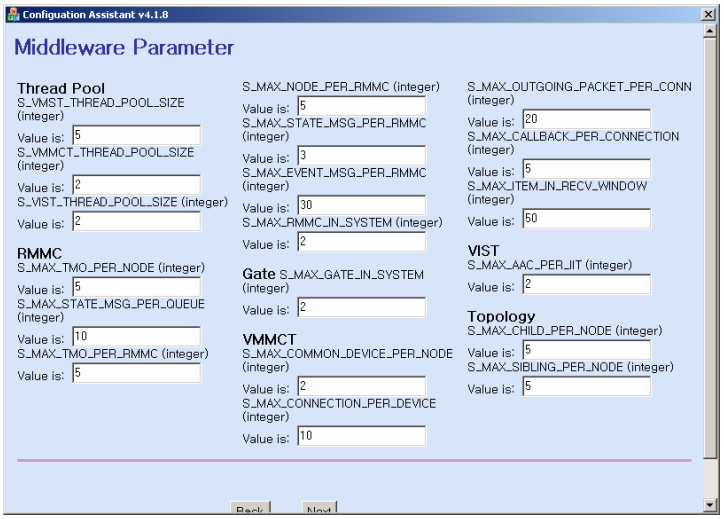
```
# These are some facilities provided by the middleware which can be
# selected by application programmers.
#
# 1. TMOSLprintf is one of the most useful facilities which is recommended
# to be used in the TMO application.
# 2. Error Reporter provides some warning services.
# 3. Parameter “warning_msg_nbb_size” is valid only when Error Reporter
# is enabled. It defines the size of NBB which is used to contain warning
# messages.
[misc]
Switch_TMOSLprintf = ON
Switch_Error_Reporter = OFF
warning_msg_nbb_size = 100
```


config.ini: Miscellaneous Facilities

```
# Switch_TMOSLprintf
#       Users can enable or disable the TMOSLprintf function at config.ini
# or at runtime using ResumeTMOSLprintf or SuspendTMOSLprintf APIs.
#
# Switch_Error_Reporter
#       Once a user activates the Error Reporter Warning service, VCT will
# issue warning messages whenever it finds that the ORT has already
# passed when it recognizes an arriving message for the first time as an
# RMMC message.  Such warning messages will be delivered to an NBB.
# This NBB is of a special type in that the overflow is ignored. Application
# can retrieve warning messages from the NBB through another TMOSL
# APIs. This service can be activated and deactivated by application
# through calling certain TMOSL APIs or the configuration in the config.ini.
```


config.ini: Middleware Internals

- Snapshot of TMO Configuation Assistant: Middleware Internals



config.ini: Middleware Internals

```
# These parameters are for configuration of middleware internals.
[mw_config_param]
# Internal TMO middleware configuration parameters for advanced users
#
################ Thread Pool ################
id = S_VMST_THREAD_POOL_SIZE
type = integer
value = 5
#
id = S_VMMCT_THREAD_POOL_SIZE
type = integer
value = 2
#
id = S_VIST_THREAD_POOL_SIZE
type = integer
value = 2
#
# much much more ...
```

UCI
DREAM Lab