Netsim	Basics
	To use the program a Distributed Algorithm or Protocol must be implented and compilated using the given ProtocolInterface.
	Then the compiled code can be loaded into Netsim for simulation using an appropriate computer network.
A simulation program for developing, demonstrating and testing distributed algorithms (protocols).	• You can get this network by editing a new network or by loading a previously saved network.
A Java program with Java interfaces.	Creating and/or modification of the network is done in edit mode.
	Before simulation the simulated protocol must be chosen. This is done in protocol mode. If you haven't loaded your protocol using parameters you can here instead load any compiled java class that offers the given interface. It is also possible to choose different protocols for different nodes in the network.
	• N.B. If you load a protocol with the same class name a second time it will not be loaded since the java machine thinks it already is loaded. If you have to correct your protocol you must restart the Netsim program before loading it (or give it a different class name).
	After having choosed both network and protocol the simulation can be started by entering simulation mode.
(29) - DISTRIBUTED SYSTEMS Netsim - Sven Arne Andreasson - Computer Science and Engineering	2 (29) - DISTRIBUTED SYSTEMS Netsim - Sven Arne Andreasson - Computer Science and Engineering
Initiating Simulation	The Protocol Interface
While entering the simulation mode the initiation operation public usid initiate (Nede Interface mullede)	□ The simulated protocol must offer the interface netsim.protocol.ProtocolInterface
for the protocols in the nodes is called. In most cases it is appropriate for the protocols to start to listen to its in links when this is called.	There is an easier way to implement the protocol by using netsim.protocol.ProtocolAdapter. This adapter implements the interface and also takes care of listening to the in-links keeping the operations in a monitor.
Then the simulation can be started by starting the simulation clock.	• By implementing the receiveMessage(netsim.protocol.Message m. netsim.protocol.InLink link)
□ To get things to happen at a specific node there should be a public void trigg() operation that can be called.	operation, a subclass will receive all in-messages in a safe way.
e.g. you can let your protocol send messages to neighbour nodes when the trig is activated.	• At initiation this subclass must start with calling
When performing the simulation the network graphics shows how the messages propagates and how the nodes can shift among different states.	{ super.initiate(myNode); // this will start a listening thread for
	<pre>// each in-link.</pre>
	In most cases it will be convenient to use this Adapter. If not, the threads that will be listening to the in-links must be implemented by the protocol.
	When simulating a computer communication protocol it might be better to not use the adapter, since it hides some of the difficulties to the simulator.
(29) - DISTRIBUTED SYSTEMS Netsim - Sven Ame Andreasson - Computer Science and Engineering	4 (29) - DISTRIBUTED SYSTEMS Netsim - Sven Ame Andreasson - Computer Science and Engineering



Compiling your algorithm

□ You can compile your javaclass as

javac -cp Netsimprotocol.jar:myclassdir MyClass.java

The jar-file with the interfaces that you will need for compilation can be found at the NETSIM homepage.

9 (29) - DISTRIBUTED SYSTEMS Netsim - Sven Arne Andreasson - Computer Science and Engineering

() CHALMERS

trigg

void trigg()

throws java.lang.Exception

This operation is called by the system every time that Trigg is clicked on the corresponding node. Can be used for starting actions by a single node.

Throws:

java.lang.Exception

stop

void stop()

This is called by the system when simulation ends or when Stop is clicked for this node. If the Stop is clicked a crash is supposed to be simulated by the protocol. In most cases this can be done by stopping listening to the inlinks.

ProtocolInterface

initiate

void initiate(<u>NodeInterface</u> myNode)

This operation is called by the system when initiating a simulation. It gives the reference to the NodeInterface. The protocol object is also supposed to start listening to its inlinks when getting this call. **Parameters:**

myNode - NodeInterface gives the reference to the NodeInterface

toString

java.lang.String toString()

The String returned by this operation is treated as the name of the protocol by the system and will be shown on the menues.

Overrides:

toString in class java.lang.Object

Returns:

String protocol name

10 (29) - DISTRIBUTED SYSTEMS Netsim - Sven Arne Andreasson - Computer Science and Engineering

(CHALMERS

ProtocolAdapter

ProtocolAdapter takes care of in-links, threads and monitor.

Constructor Summary

ProtocolAdapter()

Methods inherited from interface Netsim.protocol.ProtocolInterface

toString, trigg



Field Detail

myNode protected <u>NodeInterface</u> **myNode** A reference to the corresponding NodeInterface.

myNodeName

protected java.lang.String **myNodeName** Gives the nodes name, i.e. my name.

inLinks

protected <u>InLink</u>[] **inLinks** Gives an array of all inlinks to this node.

outLinks

protected <u>OutLink</u>[] **outLinks** Gives an array of all outlinks to this node.

running

protected boolean **running** is true until the stop() operation has been called. Then it will become false.

13 (29) - DISTRIBUTED SYSTEMS Netsim - Sven Arne Andreasson - Computer Science and Engineering

(M) CHALMERS

stop

public void stop()

This is called by the system when simulation ends or when Stop is clicked for this node. If the Stop is clicked a crash is supposed to be simulated by the protocol. This adpterclass makes it stop listening to its inlinks which in most cases will be enough for this simulation. If not the subclass most provide a stop() operation that must start with:

public void stop()
{
 super.stop();

otherwise the link listening threads will not end properly. **Specified by:** stop in interface ProtocolInterface

```
15 (29) - DISTRIBUTED SYSTEMS Netsim - Sven Arne Andreasson - Computer Science and Engineering
```

```
CHALMERS
```

Constructor Detail

ProtocolAdapter

public ProtocolAdapter()

Method Detail

initiate

public void initiate(NodeInterface myNode)

This must be called when initiating the simulation. If the subclass also needs initiation (which is most likely) it must first call this as:

public void initiate(NodeInterface myNode)

super.initiate(myNode);

.

Specified by:

initiate in interface ProtocolInterface

```
Parameters:
```

 $\verb"myNode" - NodeInterface"$

14 (29) - DISTRIBUTED SYSTEMS Netsim - Sven Ame Andreasson - Computer Science and Engineering

CHALMERS

 sendMessageSynchronized

 protected void sendMessageSynchronized(Message

 OutLink

 outLink

 throws

 java.lang.Exception

 If the subclass needs to send a message that is synchronized with the incoming messages threads, it can be

done by using this operation. Parameters: message - Message outlink - OutLink Throws:

java.lang.Exception

receiveMessage

protected abstract void **receiveMessage**(<u>Message</u> msg,

InLink myLink) throws java.lang.Exception

This operation will be called for every message that is received by the node. It gives the message and the corresponding nnlink- **Parameters:** msg - Message myLink - InLink **Throws:** java.lang.Exception

(M) CHALMERS

NodeInterface

getName

java.lang.String **getName**() returns the name of the node **Returns:** String

getInlinks

InLink[] getInlinks()
returns an array of all inlinks to the node
Returns:
InLink[]

getOutLinks

OutLink[] getOutLinks()
returns an array of all outlinks from the node
Returns:
OutLink[]

17 (29) - DISTRIBUTED SYSTEMS Netsim - Sven Arne Andreasson - Computer Science and Engineering

(W) CHALMERS

<u>NetworkBroken</u> - will only be thrown if the link is defined as safe NotFound - if there is no outlink with the corresponding inlink.

sendToAllOutlinks

void sendToAllOutlinks(Message message)

throws <u>NetworkBroken</u>

 $sends \ a \ clone \ of \ {\tt message} \ on \ all \ outlinks.$

Parameters:

message - Message Throws:

 $\underline{\texttt{NetworkBroken}}$

sendToAllBut

throws <u>NetworkBroken</u>

sends a clone of ${\tt message}$ on all outlinks except the outlink to ${\tt neigbourName}.$

Parameters:

neigbourName - String

message-Message

Throws:

 $\underline{\texttt{NetworkBroken}}$ - will only be thrown if the link is defined as safe

sendTo

void sendTo(java.lang.String neighbourName,

Message message) throws NetworkBroken, NotFound

sends a clone of message to the neighbour node with the name neighbourName.
Parameters:
neighbourName - String
message - Message
Throws:
NetworkBroken - will only be thrown if the link is defined as safe
NotFound - if there is no outlink with the corresponding neighbour name.

sendTo

void **sendTo**(<u>InLink</u> neigbour, <u>Message</u> message) throws <u>NetworkBroken</u>,

$\underline{\texttt{NotFound}}$

sends a clone of message on the outlink corresponding to inlink neigbour.

Parameters: neigbour - InLink message - Message

Throws:

18 (29) - DISTRIBUTED SYSTEMS Netsim - Sven Ame Andreasson - Computer Science and Engineering

CHALMERS

sendToAllBut

setTimeOut

<u>TimeOut</u> setTimeOut (int time, TimeoutInterface customer)

to set a timeout after time time units **Parameters:** customer - TimeoutInterface

19 (29) - DISTRIBUTED SYSTEMS Netsim - Sven Arne Andreasson - Computer Science and Engineering

CHALMERS

operations that only change node color:

setWaken

void setWaken()

will set the node color on the screen to "waken" for the corresponding node. Only for graphical appearance!

setReady

void setReady()

will set the node color on the screen to "ready" for the corresponding node. Only for graphical appearance!

setIdle

void setIdle()

will set the node color on the screen to "idle" for the corresponding node. Only for graphical appearance!

setActive

void setActive()

will set the node color on the screen to "active" for the corresponding node. Only for graphical appearance!

setError

void setError() will set the node color on the screen to "error" for the corresponding node. Only for graphical appearance!

21 (29) - DISTRIBUTED SYSTEMS Netsim - Sven Arne Andreasson - Computer Science and Engineering

() CHALMERS

• operations that is used for onput/output of data:

createVisibleInteger

VisibleInteger createVisibleInteger(java.lang.String name,

int startValue) returns an object that can be seen on the screen. This object can also be used for input of data to the protocol.

Parameters:

name - String will appear as label for the value

startValue-int

Returns:

VisibleInteger

createVisibleString

VisibleString createVisibleString(java.lang.String name,

java.lang.String startValue)

returns an object that can be seen on the screen. This object can also be used for input of data to the protocol. **Parameters:**

name - String will appear as label for the value

startValue - String

Returns:

VisibleString

createVisibleBoolean

23 (29) - DISTRIBUTED SYSTEMS Netsim - Sven Arne Andreasson - Computer Science and Engineering



(iii) CHALMERS

() CHALMERS

setError

void setError(java.lang.String ErrorMessage) **Parameters:**

ErrorMessage - will be shown in a dialog menue if not the node already was in error state. will set the node color on the screen to "error" for the corresponding node.

22 (29) - DISTRIBUTED SYSTEMS Netsim - Sven Arne Andreasson - Computer Science and Engineering

VisibleBoolean createVisibleBoolean(java.lang.String name, boolean startValue)

returns an object that can be seen on the screen. This object can also be used for input of data to the protocol.

Parameters:

name - String will appear as label for the value startValue - boolean Returns:

VisibleBoolean

void writeLogg(java.lang.String row)

What is written to this log can be read by opening the log for a node on the screen. There will also be common log for all the nodes where all log data can be read in global time order.

Parameters:

row - String



OutLink	getLongestLinkTime
	int getLongestLinkTime()
sendMessage	returns the longest time (in time units) that it will take for a message to pass the link It can be used when
void sendMessage (Message message)	setting timeouts.
throws NetworkBroken	Returns:
A clone of message will be sent on the link.	int
Parameters:	
message - Message	
Throws:	
NetworkBroken - will only be thrown if the link is defined as safe	
getOutNodeName	
jeve long (tring stoutNedeNere()	
Java. Lang. Stiffing getout Notestande()	
Batures	
String	
Sumg	
selviarked	
void setMarked (boolean marked)	
can be used to give the corresponding link a different color on the screen. Only used for graphical apperence!	
25 (29) - DISTRIBUTED SYSTEMS Netsim - Sven Arne Andreasson - Computer Science and Engineering	26 (29) - DISTRIBUTED SYSTEMS Netsim - Sven Ame Andreasson - Computer Science and Engineering
InLink	getLongestLinkTime
	int getLongestLinkTime()
receiveMessage	returns the longest time (in time (initial that it will take for a message to pass the link. It can be used when
Message receiveMessage()	setting timenets
throws NetworkBroken	Returns:
Will wait for a message on the inlink and then deliver it when a message arrives	int
Returns:	
Message	setTimeOut
m	set i integrat
Throws:	
Throws: NetworkBroken - will only be thrown if the link is defined as safe	TimeOut setTimeOut (int time,
Inrows: NetworkBroken - will only be thrown if the link is defined as safe	TimeOut setTimeOut(int time, TimeoutInterface customer)
Inrows: NetworkBroken - will only be thrown if the link is defined as safe getInNodeName	<u>TimeOut</u> setTimeOut (int time, <u>TimeoutInterface</u> customer) to set a timeout after time time units. If the inlink already has got a timeout set this will be canceled and
Inrows: NetworkBroken - will only be thrown if the link is defined as safe getInNodeName	<pre><u>TimeOut</u> setTimeOut(int time, <u>TimeoutInterface</u> customer) to set a timeout after time time units. If the inlink already has got a timeout set this will be canceled and replaced by this.</pre>
Inrows: NetworkBroken - will only be thrown if the link is defined as safe getInNodeName java.lang.String getInNodeName() Cince the serve of the server and ice saidthere	<pre>TimeOut setTimeOut(int time,</pre>
Inrows: NetworkBroken - will only be thrown if the link is defined as safe getInNodeName java.lang.String getInNodeName() Gives the name of the corresponding neighbour. Batumeri	<pre>TimeOut setTimeOut(int time,</pre>
Inrows: NetworkBroken - will only be thrown if the link is defined as safe getInNodeName java.lang.String getInNodeName() Gives the name of the corresponding neighbour. Returns: String	<pre>TimeOut setTimeOut(int time,</pre>
Inrows: NetworkBroken - will only be thrown if the link is defined as safe getInNodeName java.lang.String getInNodeName() Gives the name of the corresponding neighbour. Returns: String	<pre>TimeOut setTimeOut(int time,</pre>
Inrows: NetworkBroken - will only be thrown if the link is defined as safe getInNodeName java.lang.String Gives the name of the corresponding neighbour. Returns: String	<pre>TimeOut setTimeOut(int time,</pre>
Inrows: NetworkBroken getInNodeName java.lang.String getInNodeName() Gives the name of the corresponding neighbour. Returns: String setMarked	<pre>TimeOut setTimeOut(int time,</pre>
Inrows: NetworkBroken getInNodeName java.lang.String getInNodeName() Gives the name of the corresponding neighbour. Returns: String setMarked void setMarked(boolean marked)	<pre>TimeOut setTimeOut(int time,</pre>

27 (29) - DISTRIBUTED SYSTEMS Netsim - Sven Arne Andreasson - Computer Science and Engineering

Message

clone

Message clone()

This must return a **proper** clone of the message object. This is very important since it otherwise can be improper influences among the protocol instances that will ruin the simulation. **Returns:**

Message

getTag

java.lang.String getTag() This defines how the message will be tagged on the screen.

Returns: String

29 (29) - DISTRIBUTED SYSTEMS Netsim - Sven Arne Andreasson - Computer Science and Engineering

CHALMERS