API Documentation

API Documentation

June 14, 2009

Contents

1 Module egoistClt

Author: Vassilis Lekakis

Organization: Institute of Computer Science, F.O.R.T.H

Contact: lekakis@ics.forth.gr/lekakis@gmail.com/lex@umd.edu

See Also: *http://csr.bu.edu/sns/*, {http://www.ics.forth.gr/~lekakis}

Note: The process of detected not alive neighbors and what is happening after that, needs to be redesignned. For the time being, we just inoform the bootstrap node about the problem but nothing more. We left that not implemented on purpose because we believe that the strategy should be followed have to be related to the application that is going to be built atop the Egoist Overlay network.

To Do: The bootstraping from a proxy peer is not yet implemented

1.1 Functions

time2str()
method that returns slightly changed the value of that ctimes returns

findNodeIP()
utility to determine ip address of the node
Attention: tested for Linux (debian/ubuntu) and MAC

main()

1.2 Variables

Name	Description
BOOTSTRAP_PORT	**Value:** 61223
BOOT_DELAY	**Value:** 500
ERASE_HASH	**Value:** 300
HUGE_DELAY	**Value:** 2147482647
MIL	**Value:** 1000
NACK	**Value:** 'NACK'
NBR_CONNECTOR	**Value:** 'NEIGHBOR_CONNECTOR'
NBR_FACTORY	**Value:** 'NEIGHBOR_FACTORY'
NBR_IP	**Value:** 'NEIGHBOR_IP'
NBR_PORT	**Value:** 'NEIGHBOR_PORT'
NEIGHBOR_DELIMITER	**Value:** '|'
NET_PACKET	**Value:** 'NET'
OVERLAY_DELIMITER	**Value:** '#'
OVERLAY_PACKET	**Value:** 'OVER'
PACKET_TYPE	**Value:** 0
PACK_DELIMITER	**Value:** '@'
PING_FACTORY	**Value:** 'PING_FACTORY'
PING_PERIOD	**Value:** 35

Name	Description
PING_RESULTS	**Value:** 'PING_MEASURMENTS'
PING_TIMEOUT	**Value:** 3
PKT_IDS_LIMIT	**Value:** 5000
PKT_LIMIT	**Value:** 8190
PORT_DELIMITER	**Value:** ':'
REGISTER	**Value:** 'REG'
REMOVE	**Value:** 'RMV'
REMOVE_PERIOD	**Value:** 50
REWIRING_PERIOD	**Value:** 60
SEG	**Value:** 'SEG'
TEARDOWN	**Value:** 'TRD'
UPDATE	**Value:** 'UPT'
UPDATE_HASH	**Value:** 1
UPDATE_ID	**Value:** 2
UPDATE_SRC	**Value:** 3
UPDT_DELIMITER	**Value:** '**'
maxint	**Value:** 2147483647

1.3 Class EgoistNodeID

1.3.1 Methods

__init__(*self*, *ip*, *port*, *k*, *bootp*, *pboot*)

Parameters

    `ip:`   node's ip
                  (type=String)

    `port:`   egoist's listening port
                  (type=Integer)

    `k:`   number of outgoing links
                  (type=Integer)

    `bootp:` ip address of bootstrap node
                  (type=Integer)

    `pboot:` ip address of the proxy bootstap (not implemented)
                  (type=String)

1.4 Class BootstrapProtocol

twisted.internet.protocol.BaseProtocol ─┐

        twisted.internet.protocol.Protocol ─┐

      twisted.protocols.basic.LineOnlyReceiver ─┐

                   egoistClt.BootstrapProtocol

See Also: *http://twistedmatrix.com/documents/current/api/twisted.protocols.basic.LineOnlyReceiver.html*
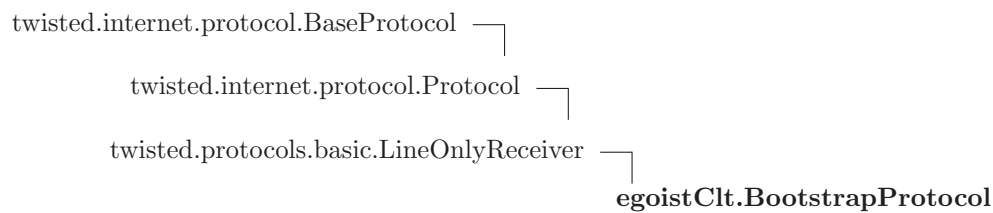
1.4.1 Methods

__init__(*self*)

---

getData(*self*, *msg*)

Utility method that returns the bootstarp information to the Bootstrap Factory when the peer has received and has parsed the data

Parameters
    `msg`: dummy
        (type=String)

Return Value
    Bootstap Information (Overlay Nodes and Graph)
    (type=python tuple)

---

sendMsg(*self*, *type*, *data*)

A wrapper method of Linereceiver's sendLine. We need this wrapper cause different types of packets needs different processing

Parameters
    `type`: packet type
        (type=String)
    `data`: data to be sent

---

lineReceived(*self*, *line*)

Method that defines the behavior of the Egoist peer upon the reception of the bootstraping information.

Overrides: twisted.protocols.basic.LineOnlyReceiver.lineReceived

---

__providedBy__(*...*)

Object Specification Descriptor

---

connectionLost(*self*, *reason=<*`twisted.python.failure.Failure` <class 'twisted.internet`....)

Called when the connection is shut down.
Clear any circular references here, and any external references to this Protocol. The connection has been closed.

Parameters

---

connectionMade(*self*)

Called when a connection is made.
This may be considered the initializer of the protocol, because it is called when the connection is completed. For clients, this is called once the connection to the server has been established; for servers, this is called after an accept() call stops blocking and a socket has been received. If you need to send any greeting or initial message, do it here.

---

dataReceived(*self, data*)

Translates bytes into lines, and calls lineReceived.

Overrides: twisted.internet.protocol.Protocol.dataReceived

---

lineLengthExceeded(*self, line*)

Called when the maximum line length has been reached. Override if it needs to be dealt with in some special way.

---

makeConnection(*self, transport*)

Make a connection to a transport and a server.
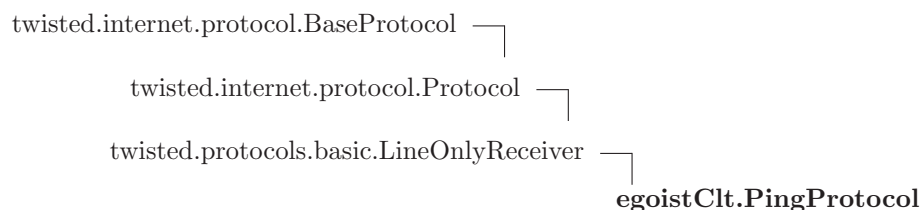This sets the 'transport' attribute of this Protocol, and calls the connectionMade() callback.

---

sendLine(*self, line*)

Sends a line to the other end of the connection.

1.4.2 Class Variables

Name	Description
MAX_LENGTH	The maximum length of a line to allow (If a sent line is longer than this, the connection is dropped). Default is 16384. **Value:** `16384`
__implemented__	**Value:** `<implementedBy twisted.internet.protocol.Protocol>`
__provides__	**Value:** `<zope.interface.declarations.ClassProvides object at 0x83...`
connected	**Value:** `0`
delimiter	The line-ending delimiter to use. By default this is '\r\n'. **Value:** `'\r\n'`
transport	**Value:** `None`

1.5 Class PingProtocol

twisted.internet.protocol.BaseProtocol ─┐

          twisted.internet.protocol.Protocol ─┐

       twisted.protocols.basic.LineOnlyReceiver ─┐

                                  egoistClt.PingProtocol

See Also: *http://twistedmatrix.com/documents/current/api/twisted.protocols.basic.LineOnlyReceiver.html*

1.5.1 Methods

__init__(*self*)

---

connectionMade(*self*)

Called when a connection is made.

This may be considered the initializer of the protocol, because it is called when the connection is completed. For clients, this is called once the connection to the server has been established; for servers, this is called after an accept() call stops blocking and a socket has been received. If you need to send any greeting or initial message, do it here.

Overrides: twisted.internet.protocol.BaseProtocol.connectionMade extit(inherited documentation)

---

sendMsg(*self, msg*)

---

lineReceived(*self, line*)

Override this for when each line is received.

Overrides: twisted.protocols.basic.LineOnlyReceiver.lineReceived extit(inherited documentation)

__providedBy__(*...*)

Object Specification Descriptor

---

connectionLost(*self, reason=*`<twisted.python.failure.Failure <class 'twisted.internet....)`

Called when the connection is shut down.

Clear any circular references here, and any external references to this Protocol. The connection has been closed.

Parameters

---

dataReceived(*self, data*)

Translates bytes into lines, and calls lineReceived.

Overrides: twisted.internet.protocol.Protocol.dataReceived

---

lineLengthExceeded(*self, line*)

Called when the maximum line length has been reached. Override if it needs to be dealt with in some special way.

---

makeConnection(*self, transport*)

Make a connection to a transport and a server.

This sets the 'transport' attribute of this Protocol, and calls the connectionMade() callback.

---

sendLine(*self, line*)

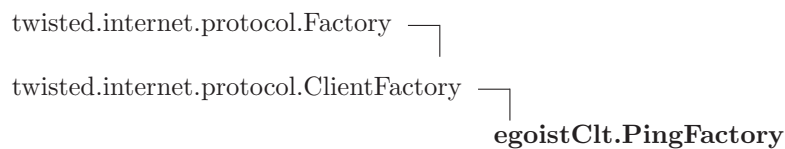Sends a line to the other end of the connection.

1.5.2 Class Variables

Name	Description
timeout	**Value: 3**

Name	Description
MAX_LENGTH	The maximum length of a line to allow (If a sent line is longer than this, the connection is dropped). Default is 16384. **Value:** `16384`
__implemented__	**Value:** `<implementedBy` `twisted.internet.protocol.Protocol>`
__provides__	**Value:** `<zope.interface.declarations.ClassProvides` `object at 0x83...`
connected	**Value:** `0`
delimiter	The line-ending delimiter to use. By default this is '\r\n'. **Value:** `'\r\n'`
transport	**Value:** `None`

1.6 Class PingFactory

twisted.internet.protocol.Factory ⌐

twisted.internet.protocol.ClientFactory ⌐

egoistClt.PingFactory

See Also: *http://twistedmatrix.com/documents/8.1.0/api/twisted.internet.protocol.ClientFactory.html*

1.6.1 Methods

__init__(*self*, *node*, *handler*)

__providedBy__(*...*)

Object Specification Descriptor

---

buildProtocol(*self*, *addr*)

Create an instance of a subclass of Protocol.
The returned instance will handle input on an incoming server connection, and an attribute "factory" pointing to the creating factory.
Override this method to alter how Protocol instances get created.

Parameters
    addr: an object implementing `twisted.internet.interfaces.IAddress`

---

clientConnectionFailed(*self*, *connector*, *reason*)

Called when a connection has failed to connect.
It may be useful to call connector.connect() - this will reconnect.

Parameters

---

clientConnectionLost(*self, connector, reason*)

Called when an established connection is lost.
It may be useful to call connector.connect() - this will reconnect.

Parameters

---

doStart(*self*)

Make sure startFactory is called.
Users should not call this function themselves!

---

doStop(*self*)

Make sure stopFactory is called.
Users should not call this function themselves!

---

startFactory(*self*)

This will be called before I begin listening on a Port or Connector.
It will only be called once, even if the factory is connected to multiple ports.
This can be used to perform 'unserialization' tasks that are best put off until things are actually running,
such as connecting to a database, opening files, etcetera.

---

startedConnecting(*self, connector*)

Called when a connection has been started.
You can call connector.stopConnecting() to stop the connection attempt.

Parameters
     `connector`: a Connector object.
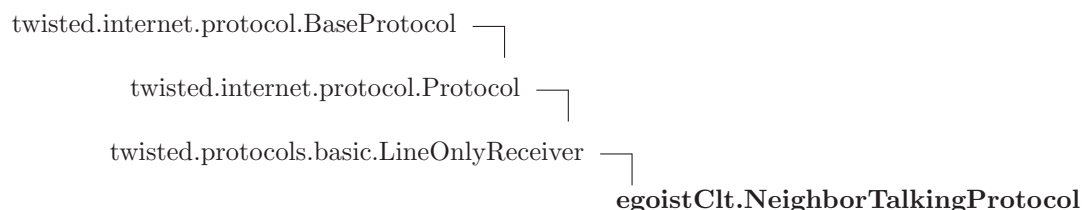
---

stopFactory(*self*)

This will be called before I stop listening on all Ports/Connectors.
This can be overridden to perform 'shutdown' tasks such as disconnecting database connections, closing
files, etc.
It will be called, for example, before an application shuts down, if it was connected to a port. User code
should not call this function directly.

1.6.2   Class Variables

Name	Description
__implemented__	**Value:** <implementedBy `twisted.internet.protocol.Factory`>
__provides__	**Value:** <`zope.interface.declarations.ClassProvides` `object at 0x83...`
noisy	**Value:** `True`
numPorts	**Value:** `0`

1.7   Class NeighborTalkingProtocol

twisted.internet.protocol.BaseProtocol ⌐

        twisted.internet.protocol.Protocol ⌐

      twisted.protocols.basic.LineOnlyReceiver ⌐

                                    egoistClt.NeighborTalkingProtocol

See Also: *http://twistedmatrix.com/documents/current/api/twisted.protocols.basic.LineOnlyReceiver.html*

1.7.1   Methods

---

connectionMade(*self*)

Called when a connection is made.

This may be considered the initializer of the protocol, because it is called when the connection is completed. For clients, this is called once the connection to the server has been established; for servers, this is called after an accept() call stops blocking and a socket has been received. If you need to send any greeting or initial message, do it here.

Overrides: twisted.internet.protocol.BaseProtocol.connectionMade extit(inherited documentation)

---

sendMsg(*self*, *msg*)

---

lineReceived(*self*, *line*)

Override this for when each line is received.

Overrides: twisted.protocols.basic.LineOnlyReceiver.lineReceived extit(inherited documentation)

__providedBy__(*...*)

Object Specification Descriptor

---

connectionLost(*self*, *reason=<*twisted.python.failure.Failure <class 'twisted.internet....*)

Called when the connection is shut down.

Clear any circular references here, and any external references to this Protocol. The connection has been closed.

Parameters

---

dataReceived(*self*, *data*)

Translates bytes into lines, and calls lineReceived.

Overrides: twisted.internet.protocol.Protocol.dataReceived

---

lineLengthExceeded(*self*, *line*)

Called when the maximum line length has been reached. Override if it needs to be dealt with in some special way.

---

makeConnection(*self, transport*)

Make a connection to a transport and a server.
This sets the 'transport' attribute of this Protocol, and calls the connectionMade() callback.

---

sendLine(*self, line*)

Sends a line to the other end of the connection.

1.7.2   Class Variables

Name	Description
MAX_LENGTH	The maximum length of a line to allow (If a sent line is longer than this, the connection is dropped). Default is 16384. **Value:** `16384`
__implemented__	**Value:** `<implementedBy twisted.internet.protocol.Protocol>`
__provides__	**Value:** `<zope.interface.declarations.ClassProvides object at 0x83...`
connected	**Value:** `0`
delimiter	The line-ending delimiter to use. By default this is '\r\n'. **Value:** `'\r\n'`
transport	**Value:** `None`

1.8   Class NeighborTalkingFactory

twisted.internet.protocol.Factory ⌐

twisted.internet.protocol.ClientFactory ⌐

egoistClt.NeighborTalkingFactory

1.8.1   Methods

__init__(*self, rmFunction, node*)

---

getNBRDeferred(*self*)

---

clientConnectionLost(*self, connector, reason*)
Called when an established connection is lost.
It may be useful to call connector.connect() - this will reconnect.

Overrides: twisted.internet.protocol.ClientFactory.clientConnectionLost extit(inherited documentation)

---

clientConnectionFailed(*self, connector, reason*)
Called when a connection has failed to connect.
It may be useful to call connector.connect() - this will reconnect.

Overrides: twisted.internet.protocol.ClientFactory.clientConnectionFailed extit(inherited documentation)

---

disconnectFromNeighbor(*self, connector*)

__providedBy__(*...*)

Object Specification Descriptor

---

buildProtocol(*self, addr*)

Create an instance of a subclass of Protocol.
The returned instance will handle input on an incoming server connection, and an attribute "factory" pointing to the creating factory.
Override this method to alter how Protocol instances get created.

Parameters
    addr: an object implementing `twisted.internet.interfaces.IAddress`

---

doStart(*self*)

Make sure startFactory is called.
Users should not call this function themselves!

---

doStop(*self*)

Make sure stopFactory is called.
Users should not call this function themselves!

---

startFactory(*self*)

This will be called before I begin listening on a Port or Connector.
It will only be called once, even if the factory is connected to multiple ports.
This can be used to perform 'unserialization' tasks that are best put off until things are actually running, such as connecting to a database, opening files, etcetera.

---

startedConnecting(*self, connector*)

Called when a connection has been started.
You can call connector.stopConnecting() to stop the connection attempt.

Parameters
    connector: a Connector object.

---

stopFactory(*self*)

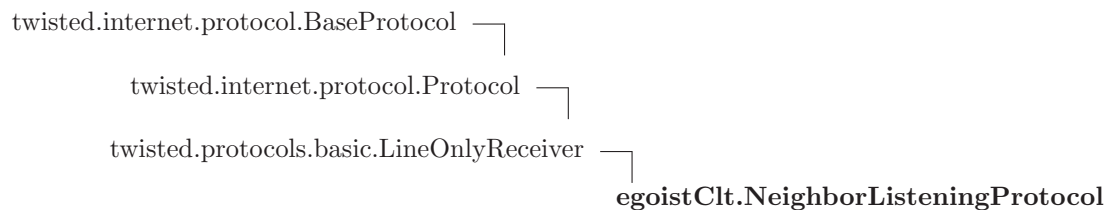This will be called before I stop listening on all Ports/Connectors.
This can be overridden to perform 'shutdown' tasks such as disconnecting database connections, closing files, etc.
It will be called, for example, before an application shuts down, if it was connected to a port. User code should not call this function directly.

1.8.2 Class Variables

Name	Description
`__implemented__`	**Value:** `<implementedBy` `twisted.internet.protocol.Factory>`
`__provides__`	**Value:** `<zope.interface.declarations.ClassProvides` `object at 0x83...`
noisy	**Value:** `True`
numPorts	**Value:** `0`

1.9 Class NeighborListeningProtocol

twisted.internet.protocol.BaseProtocol ———

     twisted.internet.protocol.Protocol ———

    twisted.protocols.basic.LineOnlyReceiver ———

            egoistClt.NeighborListeningProtocol

See Also: *http://twistedmatrix.com/documents/current/api/twisted.protocols.basic.LineOnlyReceiver.html*

1.9.1 Methods

---
connectionMade(*self*)

Called when a connection is made.

This may be considered the initializer of the protocol, because it is called when the connection is completed. For clients, this is called once the connection to the server has been established; for servers, this is called after an accept() call stops blocking and a socket has been received. If you need to send any greeting or initial message, do it here.

Overrides: twisted.internet.protocol.BaseProtocol.connectionMade extit(inherited documentation)

---
lineReceived(*self*, *line*)

Override this for when each line is received.

Overrides: twisted.protocols.basic.LineOnlyReceiver.lineReceived extit(inherited documentation)

__providedBy__(*...*)

Object Specification Descriptor

---
connectionLost(*self*, *reason*=`<twisted.python.failure.Failure <class 'twisted.internet....`)

Called when the connection is shut down.

Clear any circular references here, and any external references to this Protocol. The connection has been closed.

Parameters

---

dataReceived(*self, data*)

Translates bytes into lines, and calls lineReceived.

Overrides: twisted.internet.protocol.Protocol.dataReceived

---

lineLengthExceeded(*self, line*)

Called when the maximum line length has been reached. Override if it needs to be dealt with in some special way.

---

makeConnection(*self, transport*)

Make a connection to a transport and a server.
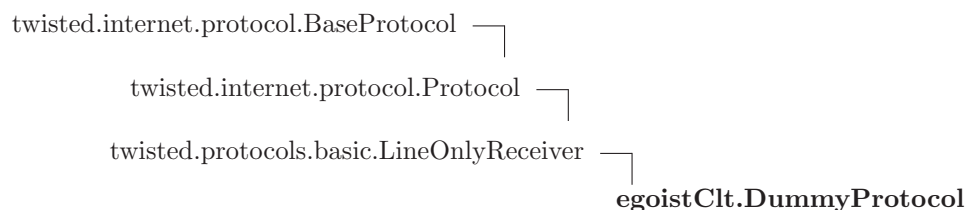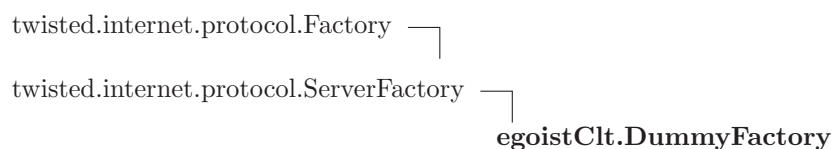This sets the 'transport' attribute of this Protocol, and calls the connectionMade() callback.

---

sendLine(*self, line*)

Sends a line to the other end of the connection.

1.9.2 Class Variables

Name	Description
MAX_LENGTH	The maximum length of a line to allow (If a sent line is longer than this, the connection is dropped). Default is 16384. **Value:** `16384`
__implemented__	**Value:** `<implementedBy twisted.internet.protocol.Protocol>`
__provides__	**Value:** `<zope.interface.declarations.ClassProvides object at 0x83...`
connected	**Value:** `0`
delimiter	The line-ending delimiter to use. By default this is '\r\n'. **Value:** `'\r\n'`
transport	**Value:** `None`

1.10 Class DummyProtocol

twisted.internet.protocol.BaseProtocol ─┐

twisted.internet.protocol.Protocol ─┐

twisted.protocols.basic.LineOnlyReceiver ─┐

egoistClt.DummyProtocol

See Also: *http://twistedmatrix.com/documents/current/api/twisted.protocols.basic.LineOnlyReceiver.html*

1.10.1 Methods

---

lineReceived(*self*, *line*)
Override this for when each line is received.

Overrides: twisted.protocols.basic.LineOnlyReceiver.lineReceived extit(inherited documentation)

__providedBy__(*...*)

Object Specification Descriptor

---

connectionLost(*self*, *reason*=<twisted.python.failure.Failure <class 'twisted.internet....*)

Called when the connection is shut down.
Clear any circular references here, and any external references to this Protocol. The connection has been closed.

Parameters

---

connectionMade(*self*)

Called when a connection is made.
This may be considered the initializer of the protocol, because it is called when the connection is completed. For clients, this is called once the connection to the server has been established; for servers, this is called after an accept() call stops blocking and a socket has been received. If you need to send any greeting or initial message, do it here.

---

dataReceived(*self*, *data*)

Translates bytes into lines, and calls lineReceived.

Overrides: twisted.internet.protocol.Protocol.dataReceived

---

lineLengthExceeded(*self*, *line*)

Called when the maximum line length has been reached. Override if it needs to be dealt with in some special way.

---

makeConnection(*self*, *transport*)

Make a connection to a transport and a server.
This sets the 'transport' attribute of this Protocol, and calls the connectionMade() callback.

---

sendLine(*self*, *line*)

Sends a line to the other end of the connection.

1.10.2 Class Variables

Name	Description
MAX_LENGTH	The maximum length of a line to allow (If a sent line is longer than this, the connection is dropped). Default is 16384. **Value:** `16384`
__implemented__	**Value:** `<implementedBy twisted.internet.protocol.Protocol>`
__provides__	**Value:** `<zope.interface.declarations.ClassProvides object at 0x83...`
connected	**Value:** `0`
delimiter	The line-ending delimiter to use. By default this is '\r\n'. **Value:** `'\r\n'`
transport	**Value:** `None`

1.11 Class DummyFactory

twisted.internet.protocol.Factory ⌐

twisted.internet.protocol.ServerFactory ⌐

egoistClt.DummyFactory

See Also: *http://twistedmatrix.com/documents/8.2.0/api/twisted.internet.protocol.ServerFactory.html*

1.11.1 Methods

__providedBy__(...)

Object Specification Descriptor

---
buildProtocol(*self, addr*)

Create an instance of a subclass of Protocol.
The returned instance will handle input on an incoming server connection, and an attribute "factory" pointing to the creating factory.
Override this method to alter how Protocol instances get created.

Parameters
    addr: an object implementing `twisted.internet.interfaces.IAddress`

---
doStart(*self*)

Make sure startFactory is called.
Users should not call this function themselves!

---
doStop(*self*)

Make sure stopFactory is called.
Users should not call this function themselves!

---

startFactory(*self*)

This will be called before I begin listening on a Port or Connector.

It will only be called once, even if the factory is connected to multiple ports.

This can be used to perform 'unserialization' tasks that are best put off until things are actually running, such as connecting to a database, opening files, etcetera.

---

stopFactory(*self*)

This will be called before I stop listening on all Ports/Connectors.

This can be overridden to perform 'shutdown' tasks such as disconnecting database connections, closing files, etc.

It will be called, for example, before an application shuts down, if it was connected to a port. User code should not call this function directly.

1.11.2 Class Variables

Name	Description
__implemented__	**Value:** `<implementedBy` `twisted.internet.protocol.Factory>`
__provides__	**Value:** `<zope.interface.declarations.ClassProvides` `object at 0x83...`
noisy	**Value:** `True`
numPorts	**Value:** `0`

1.12 Class EgoistClientFactory

twisted.internet.protocol.Factory ⎯⎯

egoistClt.EgoistClientFactory

See Also: More information of fattories *http://twistedmatrix.com/documents/8.2.0/api/twisted.internet.protocol.Factory.htm*

1.12.1 Methods

__init__(*self, egoid*)

---

handleSinglePing(*self, data*)

When the deferred is triggered this method is called to update the local metric matricies of the peer

Parameters
    `data`: Peer and link quality
            (*type=python list*)

---

startFactory(*self*)

This will be called before I begin listening on a Port or Connector.

It will only be called once, even if the factory is connected to multiple ports.

This can be used to perform 'unserialization' tasks that are best put off until things are actually running, such as connecting to a database, opening files, etcetera.

Overrides: twisted.internet.protocol.Factory.startFactory extit(inherited documentation)

---

bootUpdate(*self*, *bsmsg*)

Sending updates to the bootstap node

Parameters
> **bsmsg:** the actual update message
> > *(type=String)*

---

propagateUpdates(*self*, *peermsg*, *just_connected*)

propagates updates about graph changes to the direct neighbors

Parameters
> **peermsg:**   the actual message
> **just_coonected:**   list with the new neighbors after a local search. For these neighbors we have to wait to initiate a connection to them first

---

isDuplicate(*self*, *hash*)

checks if the incoming packet have been received before

Parameters
> **hash:** the hash of the incoming update packet
> > *(type=String)*

Return Value
> True or False in repsect the result
> *(type=Boolean)*

---

isFresh(*self*, *src*, *id*)

checks if the incoming update is fresh. In other words if the counter for the peer that have generated the packet has a different value than the one specified in the packet header

Parameters
> **src:** the peer have generated the update
> > *(type=String)*
> **id:** counter value that was found in the packet header
> > *(type=integer)*

Return Value
> the result of the check
> *(type=Boolean)*

---

ping(*self*)

periodic method that is used to infer the link qualities to each one of the rest of the nodes

---

removeNode(*self*, *node*)

---

when is a node is marked as dead. It removes it from the internal routing tables as well as sends a remove update to the bootstrap node

Parameters
   node: The node will be removed
            (type=String)

---

processUpdate(*self*, *hash*, *id*, *src*, *rcvPkt*)

---

method that processes the incoming updates packets as well as decided if will forward them to its neighbors

Parameters
   hash: hash located in the update header
               (type=String)
   id: packet id that defines freshness
               (type=Integer)
   src: packet originator
               (type=String)
   rcvPkt: the actual recieved data
               (type=String)

---

formOverlay(*self*)

---

method is called to initiate the first neighbors of node inside the overlay. Regarding the status of the network this method decides if the periodic ping, and localSearch will be enabled

---

selectRandomNeighbors(*self*, *inNeighborsNeed*=False, *insufficientOverlay*=False)

---

method called when a node has lost its neighbors during two rewiring periods.

Attention: not thoroughly tested

---

localSearch(*self*)

The heart of egoist. Local search is called every T=REWIRING_PERIOD selected randomly one of the existing links and applies the local search heuristic to the rest of the potential links in order to select the one that provides the minimum cost for reaching the rest of the network. In the beginning are checked some 'corner' cases when the egoist peer has lost the one of its neighbors or all of them

---

__providedBy__(...)

Object Specification Descriptor

---

buildProtocol(*self, addr*)

Create an instance of a subclass of Protocol.

The returned instance will handle input on an incoming server connection, and an attribute "factory" pointing to the creating factory.

Override this method to alter how Protocol instances get created.

Parameters

      addr: an object implementing `twisted.internet.interfaces.IAddress`

---

doStart(*self*)

Make sure startFactory is called.

Users should not call this function themselves!

---

doStop(*self*)

Make sure stopFactory is called.

Users should not call this function themselves!

---

stopFactory(*self*)

This will be called before I stop listening on all Ports/Connectors.

This can be overridden to perform 'shutdown' tasks such as disconnecting database connections, closing files, etc.

It will be called, for example, before an application shuts down, if it was connected to a port. User code should not call this function directly.

1.12.2 Class Variables

Name	Description
__implemented__	**Value:** `<implementedBy twisted.internet.protocol.Factory>`
__provides__	**Value:** `<zope.interface.declarations.ClassProvides object at 0x83...`
noisy	**Value:** `True`
numPorts	**Value:** `0`

Index