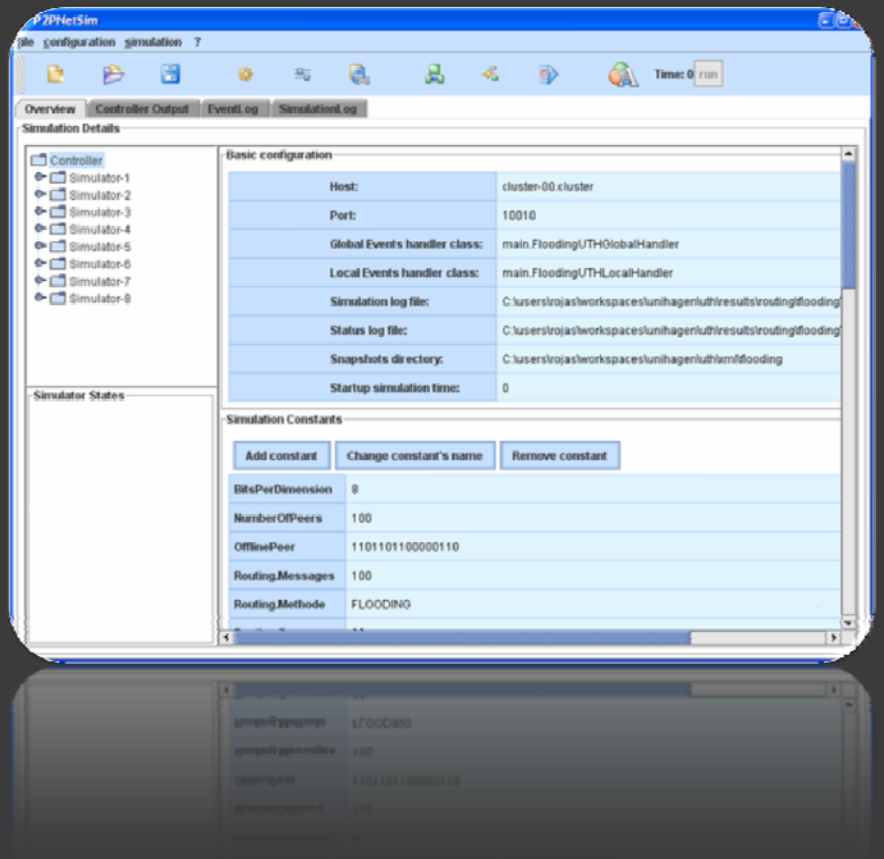


P2PNetSim

An Environment for analysis of large-scale
P2P Networks

Hauke Coltzau, Miguel Rojas
Prof. Dr. hab. Herwig Unger
FernUniversität in Hagen
Lehrgebiet Kommunikationsnetze

Outlook

- Introduction
- IR Problem in P2P
- IR Analysis & Projects
- P2PNetSim
- P2PNetSim – News / Enhancements
- Summary

Introduction

- We present the progress of P2PNetSim development
- P2PNetSim was started as a simulation tool initiative for large p2p networks by Coltzau at UniRostock
- Aiming to add flexibility to support the needs of current P2P research: Architecture, Usability and Scalability

IR Problem in P2P Networks

- The main challenge for information retrieval in P2P networks is to be able to guide the query to the sources that contain the most relevant answers in a fast and efficient way:: to reduce the number of messages sent and increase the hit rate of finding the requested resource

IR Analysis

- Estimate the results with mathematical models
- Simulation of the particular P2P network: suitable for experimenting with protocol modifications
- P2P network **simulators** are needed for analyzing the behavior of complex P2P systems

IR Projects

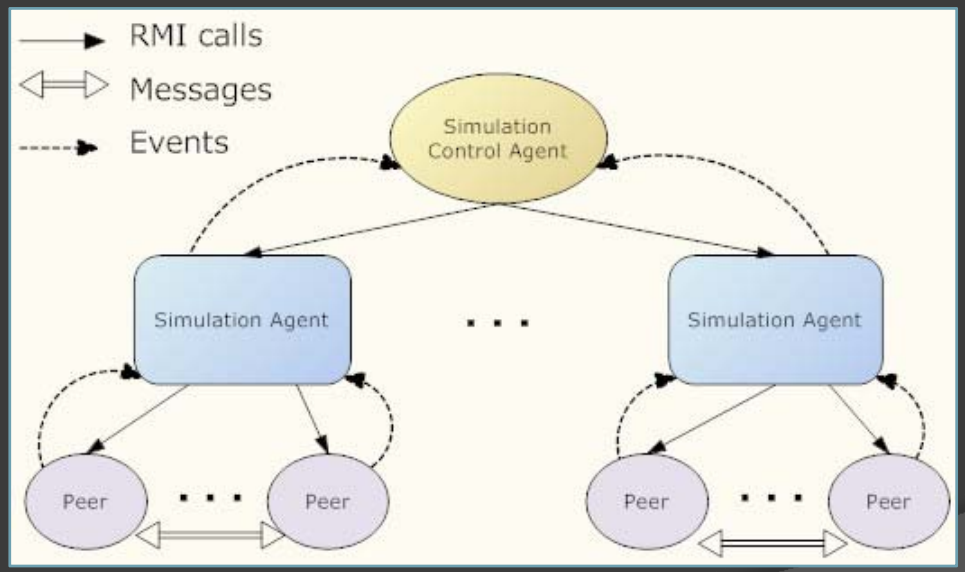
- Peerware, Anthill, Alvis and Nutsch, e.g.
- Most of current works on query log analysis are done for Web search engines: Silverstein et al. (Altavista)
- Beitzel et al. analyzed the changes of queries in terms of query popularity and uniqueness over time

P2PNetSim :: Main Features

- Discrete-event simulator
  - Usability: API and source code well documented
  - Scalability: More than 100000 nodes
  - Flexibility: enough to use with different P2P Protocols
  - Statistics: Mechanism to gather statistics
  - Parallel and Distributed Execution
  - Easy learning and using

P2PNetSim :: Architecture

- Simulation Control Agent (SCA)
- Simulation Agent (SA)
- Peers



P2PNetSim :: SCA

- Running the simulation, i.e. synchronizing the Simulation agents, distributing the current time and wait for the Simulation Agents to finish the current simulation step.
- Gathering status information from the peers via events, if necessary
- Controlling and Configuration of the simulation, i.e. adding, moving and deleting Simulation Agents and peers and change their configuration parameters

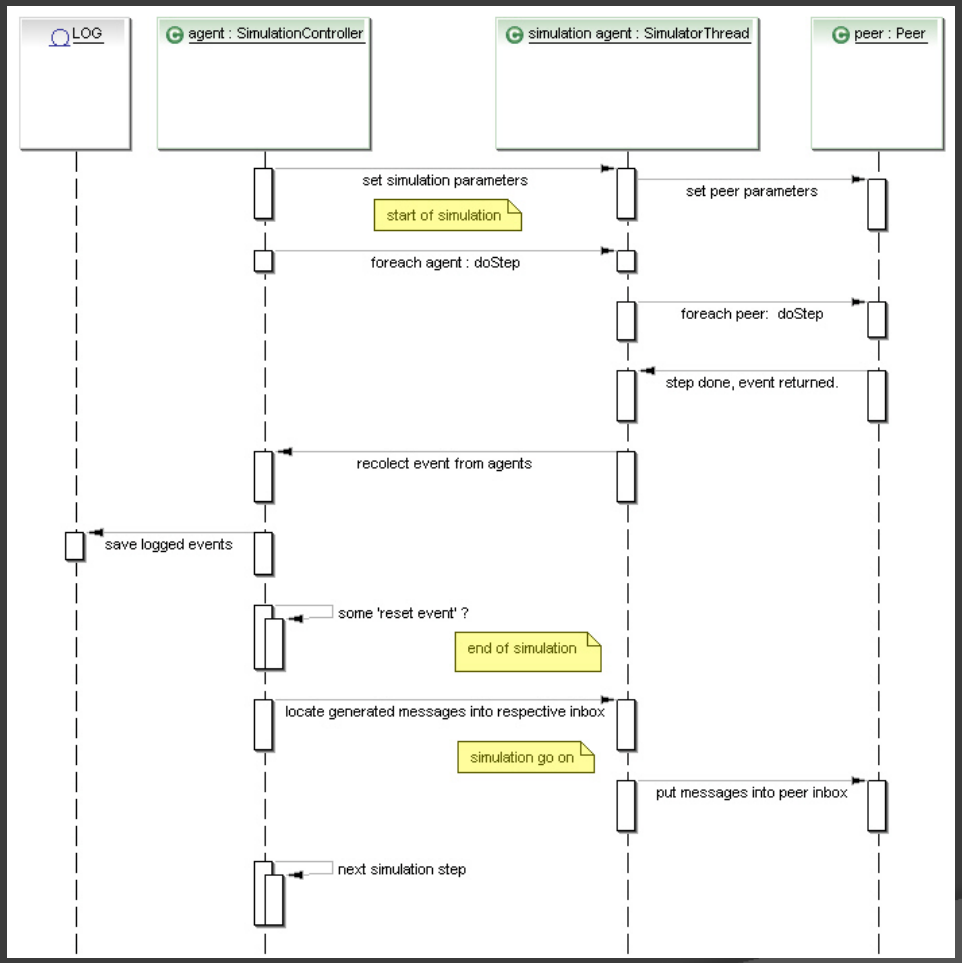
P2PNetSim :: SA

- Are able to run the actual simulation, representing one IPsubnet each
- They communicate with each other using RMI mechanisms
- The central point of events generated by peers located in a subnet

P2PNetSim :: Peers

- Are provide by an interface to access the incoming and outgoing messages
- Simulation tool is able to deliver messages from one peer to another
- Properties, such as address number, bandwidth, class name, etc, are initialized by the Simulation Agent

P2PNetSim :: LifeCycle



P2PNetSim :: API

```java
import de.uniro.p2psim.core.Address;
import de.uniro.p2psim.core.Message;
import de.uniro.p2psim.core.Peer;
import de.uniro.p2psim.core.event.PeerEvent;
import de.uniro.p2psim.util.Param;

public class KompassPeer extends Peer
{
    public void configure() // fetch parameters from xml configuration
    {
        try
        {
            ttl = getParameter( "TTL" ).getValue();
        }
        catch ( Exception e )
        {
            ttl = "-1";
        }
    }

    // injection of peer properties
    public boolean setParameter( String key, Param value )
    {
        if ( key != null && key.equals( "TTL" ) )
        {
            ttl = value.getValue();
        }

        return super.setParameter( key, value );
    }

    protected void handleMessage( Message msg, long currentTime )
    {
        // process message
    }
```

```java

    protected Address next( Message msg )
    {
        // select next peer

        return new Address();
    }

    public PeerEvent doSim( long currentTime ) // simulation step
    {
        PeerEvent result = new PeerEvent();
        try
        {
            Message m = receive();
            while ( m != null)
            {
                handleMessage( m, currentTime );
                Address next = next( m );
                send( m, next );

                m = receive();
            }
            result.setMessage( "SUCESS" );
        }
        catch ( Exception e )
        {
            result.setMessage( "FAILURE" );
            result.setPause( true );
            result.setReset( true );
        }

        return result;
    }

    protected String ttl;
}
```

P2PNetSim :: Notable News

- This includes recent news items developed in last months:
  - SCA
  - SA
  - Peers
  - GUI
  - Utilities

P2PNetSim :: SCA

- GlobalEventHandler ::
  - recolects events from Simulation Agents
- Access to all peer status
- Access to constants
- Injection of parameters to peers via RemotePeer

P2PNetSim :: SA

- LocalEventHandler
  - recolects events from simulated peers
  - Default implementation forwards all events
- New Simulator Agent ID/Name
- Access to Constants
- Full access to simulated peers (located in its sub-network)

P2PNetSim :: Peers

- ‚Send‘ :: Receive‘ message methods
- setParameter
  - Injection of new parameters from events handler
- doSend returns now a PeerEvent object

P2PNetSim :: GUI

- New TabbedPanel design:
  - Simulation status, Console, Log uses now bigger panels.
- Constants Panel
- Simulator States:
  - Shows SA : running or idle
- EventsHandler
  - Global and Local event handlers

P2PNetSim :: Utilities

- Examples according to new changes
  - GNUTELLA Implementation
- New Templates for Graphs creation
  - BidirectionalGraph.*createSmallWorldThroughEdgeReassignment*
  - UnidirectionalGraph.*createSmallWorldThroughPreferentialAttachment*
- ANT integration
  - Some ANT scripts has been added
  - Foreach file in dir → run simulation

P2PNetSim :: Example

- P2PNetsim - view

P2PNetSim :: Summary

- We have a simulator that can be useful for IR research
- Parallel, Distributed and heterogenious simulations
- *Sugestions are always welcome in order to increase functionality of P2PNetSim*