Adaptive Event Dissemination for Peer-to-Peer Multiplayer Online Games

Gabriele D’Angelo
<gda@cs.unibo.it>
http://www.cs.unibo.it/gdangelo/

joint work with:
Stefano Ferretti and Moreno Marzolla

Barcelona, Spain

Distributed Simulation & Online gaming (DISIO), 2011
Presentation **outline**

- Multiplayer Online Games: scalability and responsiveness
- MOGs: a peer-to-peer approach
- Gossip protocols: fixed-probability and conditional broadcast
- Adaptive gossip protocols based on stimuli
- Adaptive gossip: implementation and variants
- Performance evaluation: simulation-based
- Performance evaluation: metrics
- Conclusions and future work
**MOGs: scalability and responsiveness**

- **Scalability** and **responsiveness** are among the main issues in the implementation of **Multiplayer Online Games (MOGs)**.

- Many architectures have been proposed:
  - client/server
  - mirrored servers
  - peer-to-peer

- In all of them, the **dissemination** of game events can be a very **costly** task.

- In terms of **scalability**, the **peer-to-peer approach** is very promising but with some open issues.
MOGs: a peer-to-peer approach

- Each peer locally manages its copy of the game state.
- The peers are organized in some form of overlay network (many different topologies can be used).
- The dissemination of game events is obtained by passing messages through the overlay.
- Gossip protocols are very simple and well suited for P2P systems.
- If all nodes in a network have to be reached by every generated message, then traditional gossip protocols are quite inefficient.
Gossip protocol: **probabilistic broadcast**

- If the message is locally generated then it is broadcasted to all neighbors, otherwise it is decided at random if it will be broadcasted or ignored

**PARAMETERS:**

- \( p_b \) = probability to broadcast a message

**ADDITIONAL MECHANISMS:**

- time to live (\texttt{ttl}) in each message
- local \texttt{cache} in each node

**ALGORITHM**

function \texttt{INITIALIZATION}()

\[ p_b \leftarrow \text{PROBABILITY\_BROADCAST()} \]

function \texttt{GOSSIP}(msg)

if (RANDOM() < \( p_b \) or FIRST\_TRANSMISSION())

then

for all \( n_j \) in \( \Pi_j \) do

SEND(msg, \( n_j \))

end for

end if
Gossip protocol: **fixed probability**

- For each received message, the node randomly selects those edges through which the message must be propagated.

**PARAMETERS:**

- \( v \) = threshold value

**ADDITIONAL MECHANISMS:**

- time to live (**ttl**) in each message
- local **cache** in each node

**ALGORITHM**

```plaintext
function INITIALIZATION()
  v ← CHOOSE_PROBABILITY()
end function

function GOSSIP(msg)
  for all \( n_j \) in \( \Pi_j \) do
    if RANDOM() < v then
      SEND(msg, \( n_j \))
    end if
  end for
end function
```
Adaptive gossip

- Is it possible to exploit the **characteristics of online games** to enhance the message distribution among nodes?
- Is it possible to build “**smarter**” gossip protocols?

- In a MOG, game events are generated at a rate that can be approximated using some (game dependent) probability distribution
- Periodically each node **checks the reception rate** of game events from all other nodes in the network
- If this rate is **lower than a threshold** value, then it can send one or more **stimuli** to **neighbor nodes**
Adaptive gossip: implementation and variants

- Many different implementations and variants are possible:

  - **stimuli associated to receivers** (alg. #1)
    
    upon reception of a stimulus from a neighbor, a peer increases its dissemination probability towards that node

  - **stimuli associated to generators** (alg. #2)
    
    the peer increases the dissemination probability of all messages from a given sender towards all its neighbors

  - **stimuli associated to generators and receivers** (alg. #3)
    
    in this case, the dissemination probability of all messages from a given sender and for a given neighbor is increased

    *note:* this variant is much more specific than the previous ones
Adaptive gossip: *stimulus management*

- In practice it is a “fixed probability” scheme in which the **probability to disseminate a message** to a given neighbor is **modified by the received stimuli**

  ![Diagram](image)

  - A given node receives a new stimulus (of magnitude $\sigma$) at times $t_0$, $t_1$, $t_2$ and $t_3$. At time $t_2$, the stimulus adds $\sigma$ to the current value of $v_p$ (*current dissemination probability*)

  - $v_p$ **decays linearly** to $v_0$ (*baseline dissemination probability*) after time $\Delta$ from the last received stimulus
Performance evaluation: simulation-based

- The following performance evaluation is based on simulation.
- We have designed and implemented a brand new simulator called Large Unstructured NEtwork Simulator (LUNES):
  
  http://pads.cs.unibo.it

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of nodes</td>
<td>100</td>
</tr>
<tr>
<td>number of edges per node</td>
<td>2</td>
</tr>
<tr>
<td>number of graphs per evaluation</td>
<td>100</td>
</tr>
<tr>
<td>construction method</td>
<td>Erdos-Renyi generator</td>
</tr>
<tr>
<td>cache size (local to each node)</td>
<td>256 slots</td>
</tr>
<tr>
<td>message Time To Live (ttl)</td>
<td>8</td>
</tr>
<tr>
<td>simulated time (gaming time)</td>
<td>5000 time-steps</td>
</tr>
<tr>
<td></td>
<td>(after building)</td>
</tr>
</tbody>
</table>
Performance evaluation: **metrics**

- **Coverage**
  - percentage of nodes that have received **all the messages** that have been produced during the whole game execution
    
    “are the game events received by all gamers?”

- **Delay**
  - average number of **hops** that are necessary to receive a message after its creation

  “is the dissemination of new events timely?”
Performance evaluation: **cost metrics**

- Defining an appropriate **cost metric** is necessary to compare all the dissemination protocols in the same conditions.

- **Overhead ratio**  
  \[ \rho = \frac{\text{delivered messages}}{\text{lower bound}} \]

  - **delivered messages** = *total number* of messages delivered in a simulation run by a specific dissemination protocol.

  - **lower bound** = *minimum number* of messages that have to be sent by a dissemination protocol that never sends duplicates but obtains full coverage.

- In the following we will compare all the dissemination protocols in terms of **coverage** and **delay** for many different overhead ratios.
Evaluation: **coverage** rate (%)
Evaluation: coverage rate (%)
Evaluation: coverage rate (%)

all protocols act as a full broadcast
Evaluation: delay (number of hops)

Dissemination protocol comparison: delay

- Adaptive dissemination, alg. #1
- Fixed probability
- Probabilistic broadcast

Overhead ratio ($\rho$) vs. Delay (number of hops)
Evaluation: **coverage** rate (%)
Evaluation: delay (number of hops)

Adaptive protocols comparison: delay

- Adaptive dissemination, alg. #1
- Adaptive dissemination, alg. #2
- Adaptive dissemination, alg. #3

Delay (number of hops) vs. Overhead ratio ($\rho$)
Tuning of the gossip protocol

- In terms of **coverage**, protocol #3 “stimuli associated to generators and receivers” is the clear **winner**
- In terms of **delay** it is comparable with the others
- Many parameters can be used for the tuning of **protocol #3**

<table>
<thead>
<tr>
<th>setup</th>
<th>monitoring period</th>
<th>stimulus magnitude</th>
<th>stimulus length</th>
<th>stimulus threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>50</td>
<td>0.5</td>
<td>1000</td>
<td>1</td>
</tr>
<tr>
<td>#2</td>
<td>50</td>
<td>0.5</td>
<td>5000</td>
<td>1</td>
</tr>
<tr>
<td>#3</td>
<td>50</td>
<td>0.5</td>
<td>1000</td>
<td>3/4</td>
</tr>
<tr>
<td>#4</td>
<td>50</td>
<td>0.7</td>
<td>10000</td>
<td>1</td>
</tr>
<tr>
<td>#5</td>
<td>30</td>
<td>0.25</td>
<td>10000</td>
<td>1</td>
</tr>
<tr>
<td>#6</td>
<td>30</td>
<td>0.25</td>
<td>10000</td>
<td>1/2</td>
</tr>
</tbody>
</table>
Evaluation: **coverage** rate (%)
Evaluation: delay (number of hops)

Protocol #3, different setups: delay

- adaptive dissemination, alg. #3, setup #1
- adaptive dissemination, alg. #3, setup #2
- adaptive dissemination, alg. #3, setup #3
- adaptive dissemination, alg. #3, setup #4
- adaptive dissemination, alg. #3, setup #5
- adaptive dissemination, alg. #3, setup #6

Delay (number of hops)

Overhead ratio (ρ)
Conclusions and future work

- It is possible to build **smarter gossip protocols**
- The characteristics of many systems can be exploited for the implementation of **adaptive dissemination protocols**
- We have proposed a new methodology for the **comparison** of dissemination protocols: **outcomes** and **cost** of each protocol have to be both considered
- Very simple mechanisms have shown good performance in terms of **coverage** and **delay**
- What about more complex forms of adaptivity?
- **LUNES**: a new simulator for the experimental evaluation of protocols on large scale networks
Further information

Gabriele D'Angelo, Stefano Ferretti, Moreno Marzolla

Adaptive Event Dissemination for Peer-to-Peer Multiplayer Online Games

Proceedings of the 2\textsuperscript{st} Workshop on Distributed Simulation and Online gaming (DISIO). Barcelona, Spain, March 2011

A draft version of this paper is freely available at the following link:


Gabriele D'Angelo

- E-mail: \texttt{<g.dangelo@unibo.it>}
- http://www.cs.unibo.it/gdangelo/
Adaptive Event Dissemination for Peer-to-Peer Multiplayer Online Games

Gabriele D’Angelo
<gda@cs.unibo.it>
http://www.cs.unibo.it/gdangelo/

joint work with:
Stefano Ferretti and Moreno Marzolla

Barcelona, Spain

Distributed Simulation & Online gaming (DISIO), 2011