HttpTools: A Toolkit for Simulation of Web Hosts in OMNeT++

Kristján Valur Jónsson

Reykjavik University

March 6th, 2009

The presenter

Kristján Valur Jónsson is currently a Ph.D. student in computer science at **Reykjavik University** in Iceland, with the **Laboratory for Dependable Secure Systems** (LDSS) `http://ldss.ru.is`

Personal homepage:
`http://www.ru.is/kennarar/kristjanvj04`

e-mail:
kristjanvj04@ru.is

REYKJAVÍK UNIVERSITY
HÁSKÓLINN Í REYKJAVÍK

LDSS

Outline

1. Introduction
   - Outline of the talk
   - Introduction
   - Related work
2. The HttpTools toolkit
   - Introducing the components
   - Request and document generation model
   - Random and scripted operation
3. Application
   - Modeling of a DDoS attack
   - The scenario
   - Results
4. Conclusions
   - Current status
   - Conclusions and future work
   - Q & A

Introduction

- **Contribution:** `HttpTools`: A set of components for OMNeT++
- Simulation of Web browsers and servers
- Integrated within the INET framework
- Can employ the TCP/IP stack modeling of INET or direct message passing
- Can employ statistical distributions for browsing behavior and Web site composition as well as scripted models

Motivation

- The `HttpTools` project grew out of some of our OMNeT++ simulations for various projects and experimentation.
- Motivating research is in the field of Web applications, specifically distributed measurement of Web application properties.
- Our research requires large-scale modeling of Web hosts, which focuses on the end-points, rather than the intermediary network.

Related work

- Simulation of Web hosts in the context of networking research is not new.
- Several components exist, e.g. PackMime-HTTP (Cao et.al, 2004) for the *ns-2* simulator.
- Other Web traffic generators are SWING (Vishwanath and Vahdat, 2006) and SURGE (Barford and Crovella, 1998).
- Much of the previous work has been focused on generating HTTP traffic to investigate effects on the network infrastructure itself.

Cao, J., Cleveland, W., Gao, Y., Jeffay, K, Smith, F. and Weigle, M. *Stochastic models for generating synthetic HTTP source traffic*, INFOCOM, **2004**, 3, 1546-1557

Vishwanath, K. V. and Vahdat, A. *Realistic and responsive network traffic generation*, SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications, ACM, **2006**, 111-122

Barford, P. and Crovella, M. *Generating representative Web workloads for network and server performance evaluation* SIGMETRICS Perform. Eval. Rev., ACM, **1998**, 26, 151-160

Related work – OMNeT++

Previous efforts for OMNeT++ are as far as we are aware of:

- **INET framework**: `TCPBasicClientApp` and `TCPSrvHostApp` provided the starting point for this work.
- **OMNeT++ supplied components**: `httpclient` and `httpserver`
- **WebServer** project[1]

[1] Waldemar Kubassa, `http://metis.weia.po.opole.pl/~d18616`

The HttpTools components

`HttpTools` consists of three components:

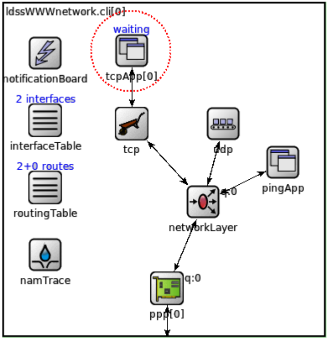
- **Browser** – `httptBrowser` – simulates usage patterns.
- **Server** – `httptServer` – simulates document generation.
- **Controller** – `httptController` – a omnipotent entity which helps browser to select servers at random. Supports imposing popularity distributions on the simulated server population – uniform, zipf, histogram, in addition to popularity modification events.

All code is open-source, GNU public license.

Project wiki and subversion source tree can be found at `http://code.google.com/p/omnet-httptools`.

INET integration

Server and browser components integrate into the `StandardHost` module of the INET framework. Plug in as tcpApp modules.



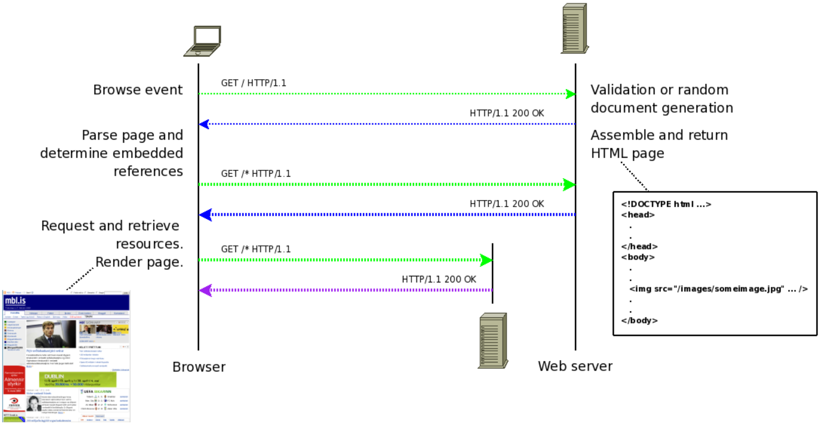
`DirectHost` module provided for direct message passing applications.

Traffic generation model

”Page-based” model:

- Browser components emulate actual browsers, i.e. generate HTTP GET requests, triggered by browse events.
- Server components respond to received HTTP GET requests by serving HTML documents or resources
- Browsers parse received HTML documents and extract embedded object references. The browser issues request for each object. HTTP 1.1 communications model emulated.

Communications model



Browse event

GET / HTTP/1.1

Validation or random document generation

HTTP/1.1 200 OK

Assemble and return HTML page

Parse page and determine embedded references

GET /* HTTP/1.1

HTTP/1.1 200 OK

Request and retrieve resources. Render page.

GET /* HTTP/1.1

HTTP/1.1 200 OK

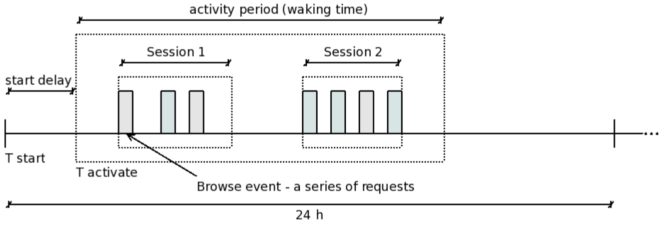
```
<!DOCTYPE html ...>
<head>
  .
  .
</head>
<body>
  .
  .
  <img src="/images/someimage.jpg" ... />
  .
  .
</body>
```

Browser

Web server

Browsers request model

Observed behavior of browsing users: Periodic bursts of activity, divided by the period of time needed to read a produced page, do some other work and to sleep.


activity period (waking time)
Session 1
Session 2
start delay
T start
T activate
Browse event - a series of requests
24 h

Bursts of browse events, separated by periods of rest.

Modes of use

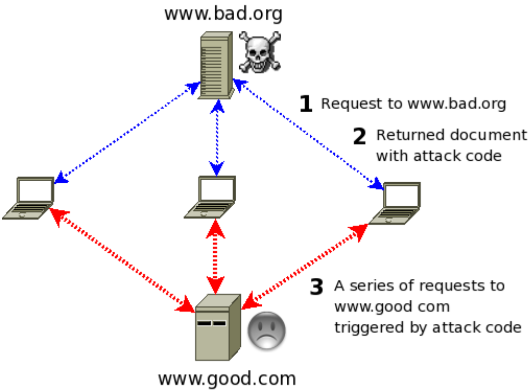
1. **Random browsing and document generation**:
   - Browser components generate requests according to statistical models for user behavior.
   - Servers respond to requests by generating documents and other resources using statistical models for HTML page composition and object type and size distributions.
   - The controller component is queried by the browser for each request to select the communications partner.
2. **Scripted browsing and document generation**:
   - Browsers generate browse events to specific sites at specific scripted intervals.
   - Servers respond to requests using a predefined page structure and resources.

Sample application

The test case presented is modeling of a Web-based **distributed denial-of-service attack**, based on the *Puppetnets* work (Lam et.al, 2006). One or more compromised servers serve HTML pages with embedded JavaScript attack code. Unsuspecting browsers visiting those sites will execute the attack code.

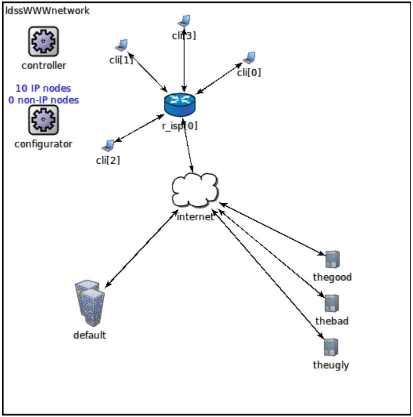
Lam, V., Antonatos, S., Akritidis, P. and Anagnostakis, K. *Puppetnets: Misusing Web Browsers as a Distributed Attack Infrastructure* CCS06, **2006**

The puppets strike



www.good.com receives unwanted traffic, resulting in spending of resources – a DDoS attack.

The scenario



Screen shot of a simplified OMNeT++ DDoS simulation.

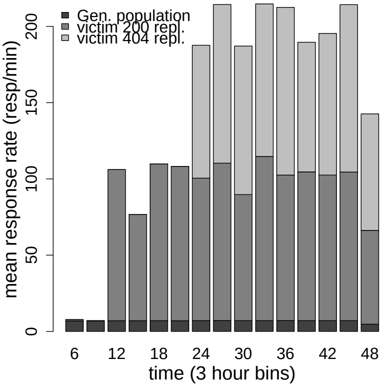
The setup

- 10000 browser nodes. Moderate request rate.
- 1000 server nodes.
- A victim – *www.good.com*.
- Two attackers, *www.bad.org* and *www.ugly.org*.
- Random request mode with uniform server popularity.
- Direct message passing used to simulate a high capacity network.

Attack code is executed by a browser which happens to hit either www.bad.com or www.ugly.com. Attack only executed once per browser hit.

- **www.bad.com** is activated at T=12h and serves attack code which requests between 100 and 200 copies of the *same image* from **www.good.com**. Tricks used to prevent browser caching in real attacks.
- **www.ugly.com** is activated at T=24h and serves attack code which sends between 100 and 200 *random URLs* to *www.good.com*.

Traffic at www.good.com



3 hour bins. Attacker 1 (www.bad.org) comes online at T=12h, attacker 2 (www.ugly.org) at T=24h.

Current status

- HttpTools is a work in progress and is expected to evolve over the next months.
- Primarily driven by my own needs.
- Community input appreciated.

Conclusions and future work

- A set of components for simulation of Web hosts in OMNeT++.
- A contribution to the OMNeT++ community and integrates into the INET framework.
- Future work includes more detailed modeling of Web applications, e.g. sophisticated AJAX applications and mashups, where the underlying application engines periodically send requests to servers. This model of communications is considerably different from HTTP/1.1.

Questions?