Large-scale Evaluation of Distributed Attack Detection

Thomas Gamer, **Christoph P. Mayer**

Institut für Telematik, Universität Karlsruhe (TH), Germany

Why Attack Detection still is important

- Distributed Denial-of-Service problem persists
  - Attack bandwidth exceeded 40 Gbit/s in 2008
  - Threatens not only servers, but provider infrastructure, too
    - Detection and mitigation still hard to achieve
    - Even harder in the core network
- But DDoS is just one example
  - Spam, botnets, worm propagations, …

> "*Our ability to effectively defend the network and its connected hosts continues to be, on the whole, ineffectual*"
>
> (Geoff Huston, IPJ, 2008)

- Lots of approaches exist in attack detection but …
  - how to evaluate them?
  - how to compare them with each other?

Thomas Gamer
Institut für Telematik
Universität Karlsruhe (TH)

Preconditions and Requirements

- How to evaluate (distributed) detection of large-scale attacks?
  - Internet or real networks, respectively
    - Normal operation must not be affected by evaluation
    - Isolation impossible
  - Testbed
    - Large testbeds are expensive
    - Administration and maintenance complex and time-consuming
  - Simulation
    - Controllable environment ensures repeatable and comparable setup

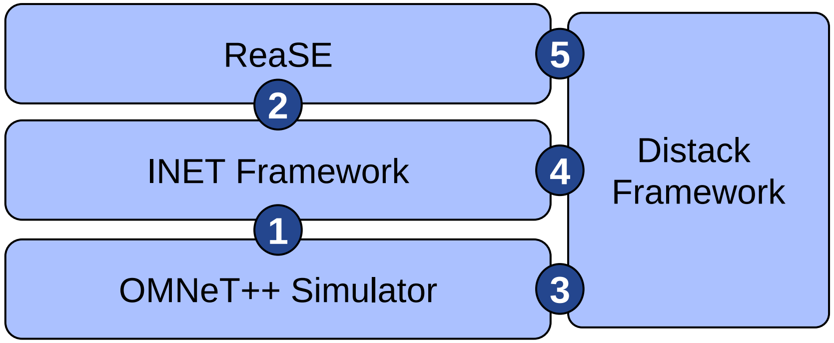
➜ Simulation toolchain for the large-scale evaluation of distributed attack detection

- Toolchain requirements
  - Simplicity and easy usability
  - Realistic simulation environments
  - Transparent deployment of attack detection in real systems
  - Tools should be well-concerted

Thomas Gamer
Institut für Telematik
Universität Karlsruhe (TH)

The Big Picture

- Components of the simulation toolchain
  - OMNeT++
  - INET Framework
    1. Extends OMNeT++ by Internet-specific protocols
  - ReaSE
    2. Adds special entities like clients, servers, or DDoS zombies
  - Distack Framework
    3. Loaded as shared library by OMNeT++
    4. Distributed attack detection is achieved based on INET protocols
    5. Integration of Distack as special entity *DistackOmnetIDS*



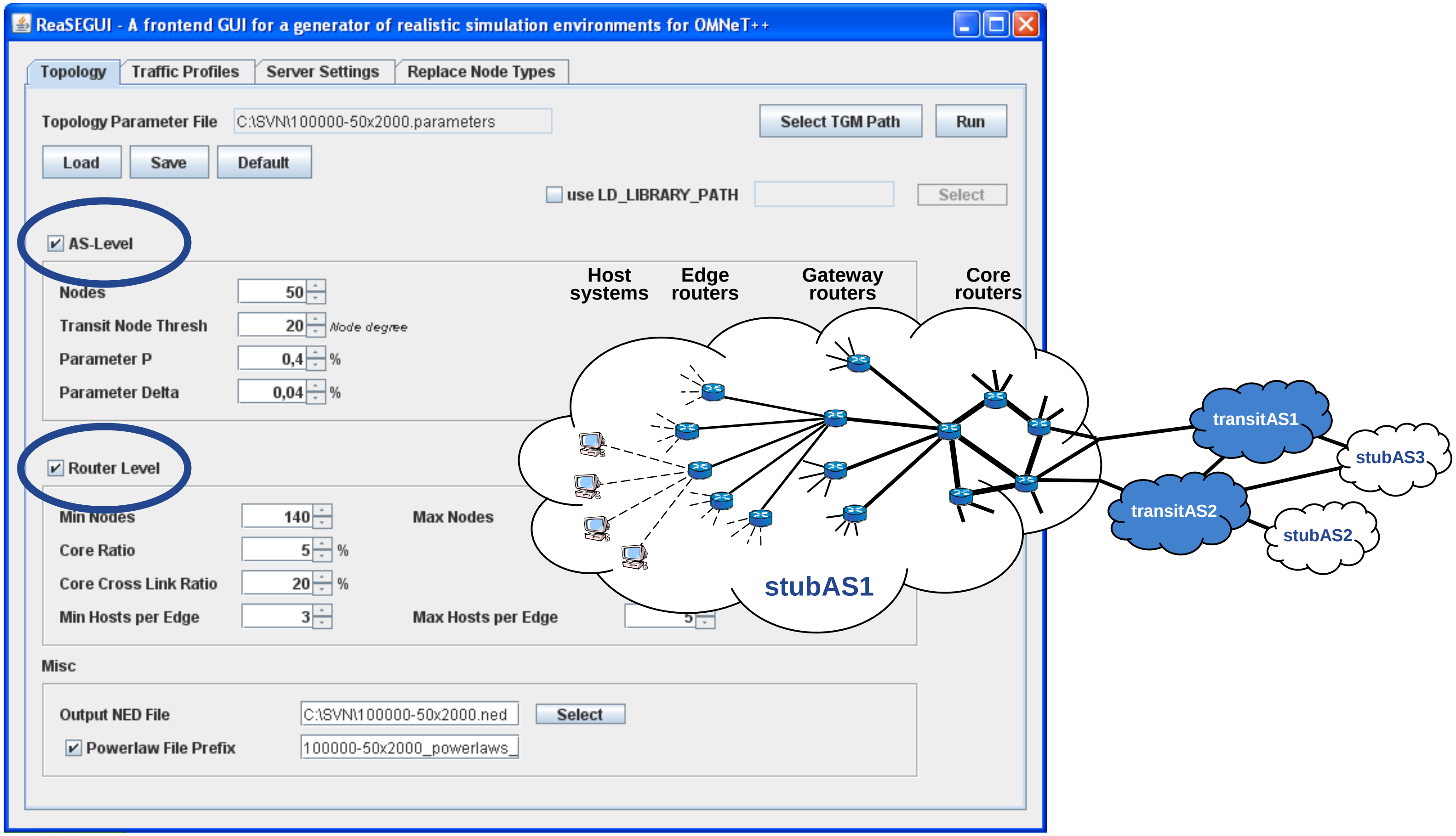
ReaSE – Overview

- Generation of a realistic simulation environment
  - Short paper [1] on basic principles last year
  - NEW Graphical user interface
    - Ensures simplicity and usability
    - Hides the actual implementations
  - NEW Open source release (July 2008)
    - Supports currently only OMNeT++ v3
    - Release of ReaSE for OMNeT++ v4 scheduled for next week

[1] Thomas Gamer, Michael Scharf, *Realistic Simulation Environments for IP-based Networks*, OMNeT++ Workshop, Mar, 2008.

ReaSE – Topology

- Create a network topology
  - NED file containing *Routers* and *StandardHosts*



ReaSE – Topology

- Create a network topology
  - NED file containing *Routers* and *StandardHosts*
- Add special entities for generation of background traffic
  - *InetUserHost*, *WebServer*, *StreamingServer*, …
- Define traffic profiles
  - Randomly selected by special entities during simulation
  - Aggregated traffic shows self-similar behavior

Thomas Gamer
Institut für Telematik
Universität Karlsruhe (TH)

ReaSE – Background Traffic

- Aggregated traffic shows self-similar behavior
  - Exemplary topology: About 50000 nodes in total
    - Divided into 20 Autonomous Systems
  - Calculation of Hurst parameter on every router
    - Based on the method of m-aggregated variances

Router type	#	Scaling factor	Average	Standard deviation
Edge	873	100 ms	0.6226	0.0352
		1s	0.6395	0.0645
Gateway	55	100 ms	0.6771	0.0461
		1s	0.7234	0.0701
Core	14	100 ms	0.8220	0.0599
		1 s	0.8927	0.0525

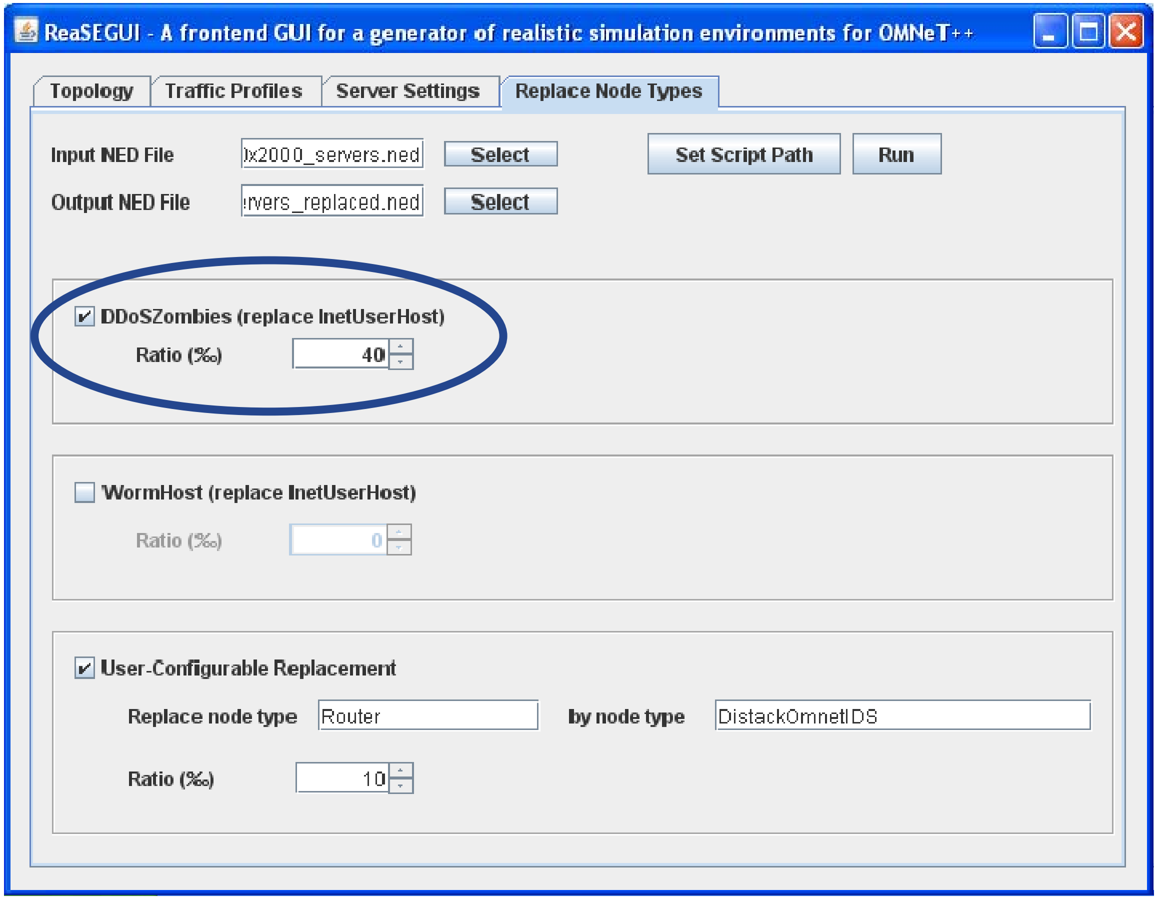
Hurst parameters of all routers

  - Topologies with 1k, 5k, 10k, and 100k nodes also show self-similarity

Thomas Gamer
Institut für Telematik
Universität Karlsruhe (TH)

ReaSE – Attack Traffic

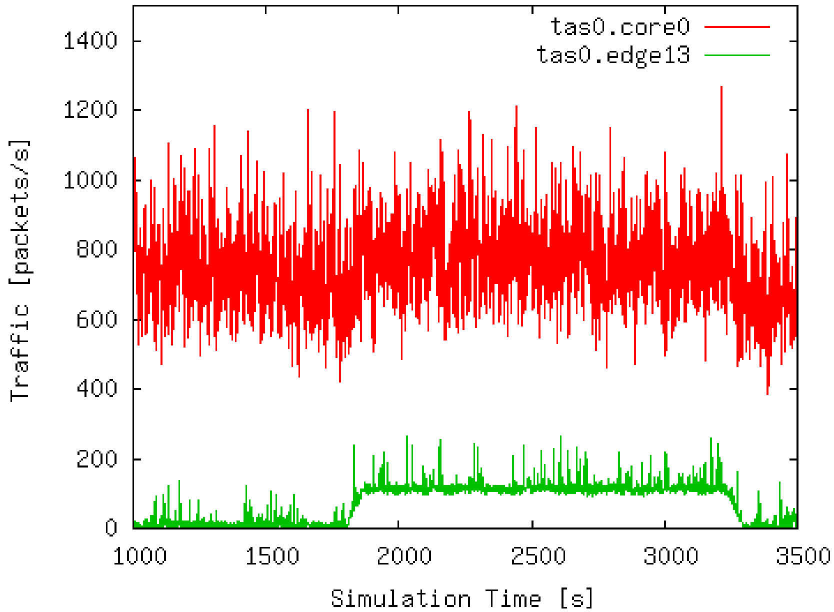
- Add special entities in regard to attack detection
  - *DDoSZombie*, *WormHost*, or *DistackOmnetIDS*



Thomas Gamer
Institut für Telematik
Universität Karlsruhe (TH)

ReaSE – Attack Traffic

- Example: Simulation of a DDoS attack



Traffic observed on two different routers in transit AS 0 during a DDoS attack

- 10440 entities within 20 AS
- ~40 DDoS zombies
  - Start of attack: 1600s
  - TCP SYN flooding
  - IP address spoofing
- Victim webserver resides in transit AS 0
- edge13 and core0 are part of the attack path

➔ Attack detection is not an easy task within the network

➔ Distributed detection may improve detection efficiency

Thomas Gamer
Institut für Telematik
Universität Karlsruhe (TH)

Distack – Overview

- Framework for anomaly-based attack detection
  - Publication [2] of architecture last year
  - NEW Enhancements in regard to usage within OMNeT++
    - Instantiation of multiple detection systems within simulation
    - Support for heterogeneous configuration of available instances
  - NEW Remote communication methods usable with OMNeT++
    - TCP sockets, path-coupled, ring-based
  - NEW Graphical user interface for scalable and easy configuration
    - Categories and available values are pre-defined

[2] Thomas Gamer, Christoph P. Mayer, Martina Zitterbart, *Distack – A Framework for Anomaly-based Large-scale Attack Detection*, SecurWare 2008, p. 34-40, Aug 2008.

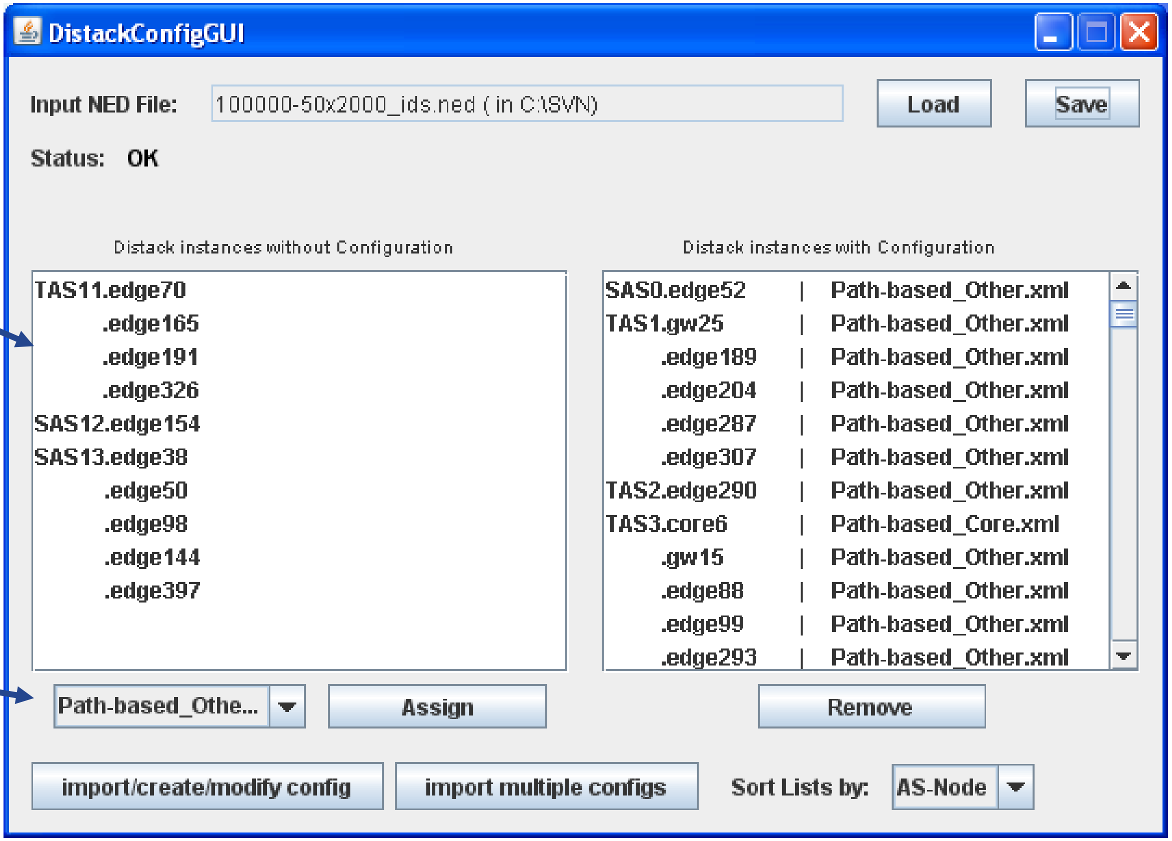
Thomas Gamer
Institut für Telematik
Universität Karlsruhe (TH)

Distack – Simulation Setup

- Scalable assignment of heterogeneous configurations to available Distack instances
  - Different sortings allow for easy grouping of instances

Currently unconfigured instances

Available configurations



Thomas Gamer
Institut für Telematik
Universität Karlsruhe (TH)

www.tm.uka.de

Evaluation of the Toolchain

- Goal of this evaluation is to provide users with a feeling about basic behavior of the toolchain
  - Basic parameters
    - CPU: Intel Xeon 5160 dualcore 3 GHz, 4 Mb shared L2 cache
    - RAM: 32 GB
    - Operating system: 64-bit Ubuntu Linux
    - OMNeT++ 3.4 and according INET framework
      - Compiled without Tcl support
  - Evaluation environments varied in
    - Topology size
    - Number of Autonomous Systems
    - Seeds for random number generators

Topology size	Number of AS			Seeds
1 000	5	10	20	20
5 000	10	20	50	20
10 000	10	20	50	10
50 000	20	50	100	5
100 000	20	50	100	5

Decreasing number of seeds due to increasing simulation duration

Thomas Gamer
Institut für Telematik
Universität Karlsruhe (TH)
www.tm.uka.de

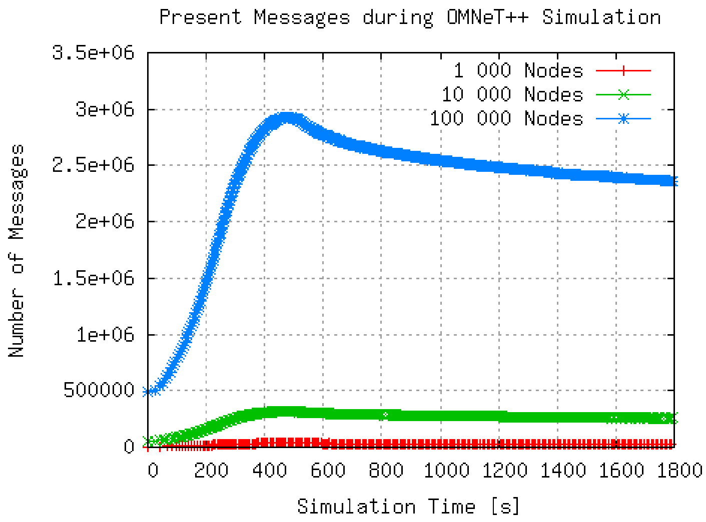
Evaluation of the Toolchain

- Goal of this evaluation is to provide users with a feeling about basic behavior of the toolchain
  - Evaluation parameters
    - Memory usage
      - Virtual size of the INET process read from *proc* filesystem
    - Duration
      - CPU time the INET process consumed
    - Messages created by OMNeT++ during simulation
      - Total number of messages
      - Number of present messages

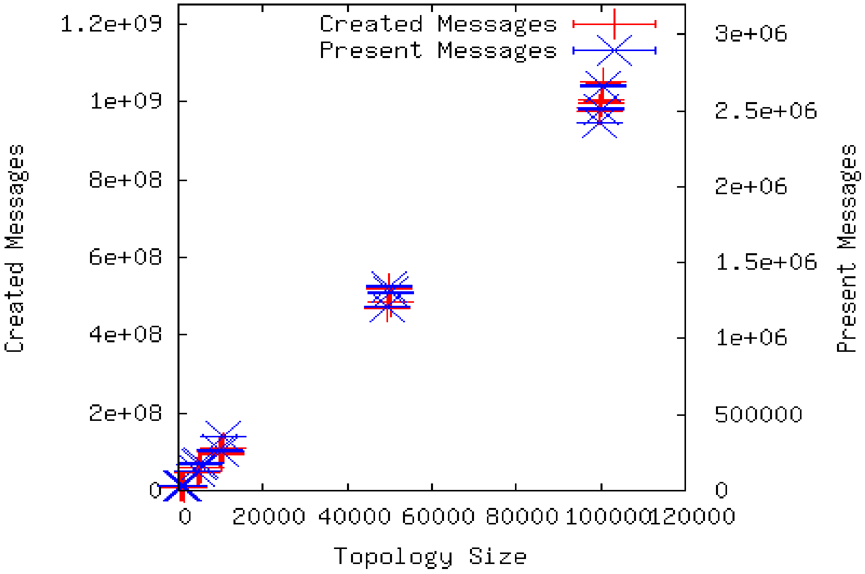
Thomas Gamer
Institut für Telematik
Universität Karlsruhe (TH)

OMNeT++ Messages

- Progress of present messages during a simulation
  - Simulated time: 1800 s



During a single simulation



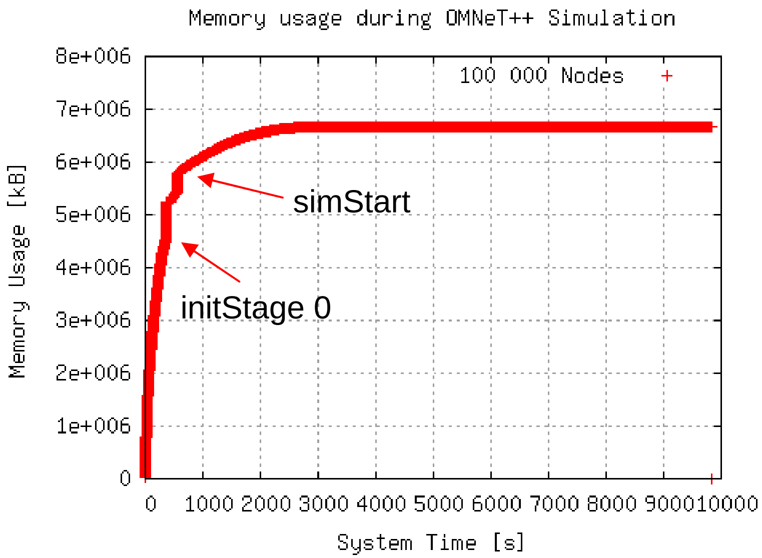
Summary of all simulations

- Linear increase of total and present messages
  - Proportional to topology size

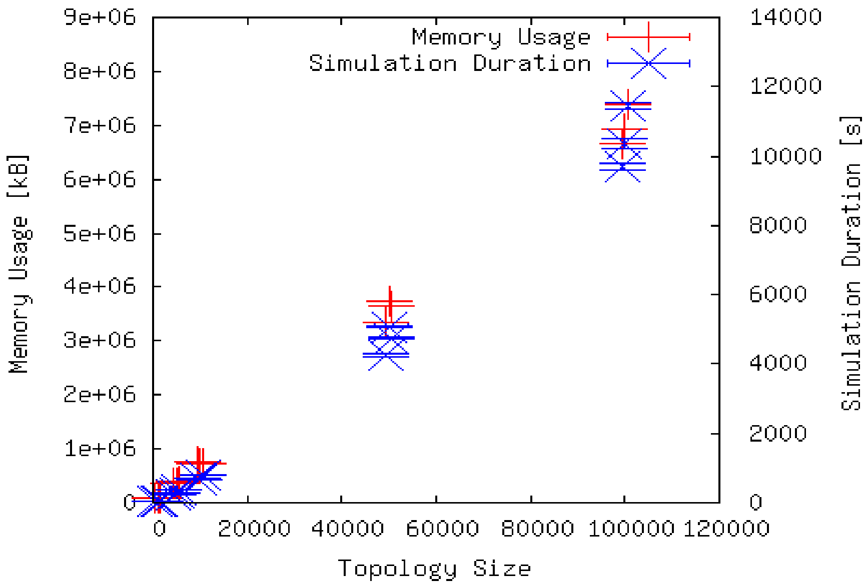
Thomas Gamer
Institut für Telematik
Universität Karlsruhe (TH)

Duration and Memory Usage

- Simulation duration and progress of memory usage
  - Simulated time: 1800 s



Memory usage during a single simulation



Summary of all simulations

- Memory usage and simulation duration increase linearly
  - Increase of simulation duration more than proportional
    - ev/sec seems not to be independent of topology size

Thomas Gamer
Institut für Telematik
Universität Karlsruhe (TH)
www.tm.uka.de

Memory Usage of Distack

- Additional integration of Distack instances
  - Exemplary topology
    - 10 000 nodes, 20 AS
  - Basic memory consumption without Distack
    - 738 478 kB
  - Shared library and dependencies
    - Need for about 6 MB of memory
  - Memory usage per Distack instance
    - About 40 kB for instantiation and traffic measurement

Thomas Gamer
Institut für Telematik
Universität Karlsruhe (TH)

www.tm.uka.de

Conclusion and Outlook

- Valuable features of our toolchain
  - Generation of realistic simulation environments
  - Transparent integration of a real attack detection system
  - Graphical user interfaces for simplification and usability
  - Scalable resource consumption
    - Major memory consumption caused by instantiation of modules
  - …and considered best: ***it's open source***

http://www.tm.uka.de/ReaSE

http://www.tm.uka.de/Distack

- Open challenges
  - Integration of traffic traces into the toolchain
  - Evaluation of an actual distributed attack detection
  - Finishing and releasing new versions for OMNeT++ 4.0

Thomas Gamer
Institut für Telematik
Universität Karlsruhe (TH)

www.tm.uka.de

Thank you!

Questions?

http://www.tm.uka.de/ReaSE

http://www.tm.uka.de/Distack

Thomas Gamer
Institut für Telematik
Universität Karlsruhe (TH)