OverSim

A Flexible Overlay Network Simulation Framework

Ingmar Baumgart, Bernhard Heep, Stephan Krause

IEEE Global Internet Symposium 2007, Anchorage, AK, USA



Universität Karlsruhe (TH) Research University · founded 1825





- Overlay
 - Flexibility
 - Scalability
- Underlay

- Heterogeneity
- Terminal Mobility
- Backbone simulation



Stephan Krause

OverSim, GI07, Anchorage, AK, USA



Additional Requirements

Interoperability

- PlanetLab experiments
- Network emulation
- Debugging and Overlay visualization



OverSim, GI07, Anchorage, AK, USA



State of the Art

- No established standard overlay simulator
- Most promising: P2PSim, OverlayWeaver and PlanetSim
- P2PSim (http://pdos.csail.mit.edu/p2psim/)
 ③ Many implemented overlay protocols
 ⑧ No real world interconnection, no visualization
- OverlayWeaver (http://overlayweaver.sourceforge.net/)
 © Real world interconnection
 - ⊗ No simulation of underlying network, not scalable
- PlanetSim (http://planet.urv.es/planetsim/)
 - © Flexible and scalable
 - ⊗ Only very simplistic model of underlying network



• How to implement all required features?

- Overlay layer: Flexibility, scalability
- Underlay model: Heterogeneity, terminal mobility, backbone simulation
- Interoperability, network emulation
- Debugging and overlay visualization
- Featured Overlays and Applications
- Performance Evaluation and Validation

OverSim, GI07, Anchorage, AK, USA

Flexibility

- Modular architecture
 - Based on OMNeT++
- Layered architecture
 - Underlying network
 - Overlay layer
 - Application layer
- Consistent interfaces between layers



- UDP between network and overlay
- Common API between overlay and application
- → Exchange of one component is transparent to all other components

OverSim, GI07, Anchorage, AK, USA



- Tradeoff: Level of detail vs. simulation speed
- Maximum scalability → SimpleUnderlay
- Packets are exchanged directly between nodes
 - Minimizes number of events
- Simple scenario: Constant delay
- For more realistic delays:
 - Place nodes in Euclidian space
 - Delay is proportional to distance
 - No additional events needed \rightarrow Still scalable

Heterogeneity and Mobility

- Extension of the synthetic coordinate model
- Nodes are assigned to an "access net" with specific bandwidth and delay characteristics
- Calculated packet delay:

$$d_{e} = d_{A} + \frac{l_{p}}{b_{A}} + c \cdot ||A - B||_{2} + d_{B} + \frac{l_{p}}{b_{B}}$$



 Mobility: Move node to new coordinate, change access net characteristics and IP address



- For detailed simulation of the underlying network: INETUnderlay
- Based on INET framework for OMNeT++
- Simulates a complete IP backbone
- Access routers attached to backbone routers
 - Simulates all effects of heterogeneous access networks
 - Bandwidth, Delay, Packet loss, Queuing effects
- Mobility is simulated by moving a node from one access router to another

OverSim, GI07, Anchorage, AK, USA

- For PlanetLab experiments and interconnecting with real world implementations:
- Emulated nodes in real networks
- Event scheduler runs at real time
- Linux TUN device for exchanging packets with a real network
- External applications can be connected to the simulation

OverSim, GI07, Anchorage, AK, USA



Network Emulation

For demonstrational purposes:

- Real nodes in emulated networks
- Emulate overlay network with INETUnderlay
- Add a gateway node
 - This router features a TUN interface
- → External devices can communicate with simulated overlay





Visualization and Debugging

 Topology visualization

 Message visualization

 Node state debugging

1	ev 🔂	a 🖅 💷 🔜 💼	×	11
	and) Simpleble			<u> </u>
	iora) simpleive	awork.ovenayrenminai[4].o	venay.cnord (id=64) (prox3104020)	
Info) Params	Gates Contents Sul	ibmodules	,
27 (objects	-		
-	Class	Name nume i nomodesent	Info Info	4
100	i	hutesFindNodeSent	0	
1000	i	missingPredecessorSt	tat 0	
100	i	missingSuccessorStab	bRe 0	
100	i	joinRetry	1	
-	NodeHandle	bootstrapNode	1.0.0.3:1024 11762ae7a2e3f11057ba75d5cfd042077738c32e 36	
100	NodeHandle	predecessorNode	1.0.0.4:1024 dcc2b95ab9e435675dd5c4b7506be81d07c84636 48	
(sto nap<	d::map <ove< th=""><th>rlayKey,NodeHandle: odeHandle>) SimpleNetwo</th><th>>)al[4].overlay.successorList.successorMap</th><th>d 61</th></ove<>	rlayKey,NodeHandle: odeHandle>) SimpleNetwo	>)al[4].overlay.successorList.successorMap	d 61
(sto map< eral	d::map <ove :OverlayKey,N Fields</ove 	rlayKey,NodeHandle: odeHandle>) SimpleNetwo	>)al[4].overlay.successorList.successorMap	D
(sto map< ral std:	d::map <ove :OverlayKey,N Fields ::map<0ver1ayK</ove 	rlayKey,NodeHandle: lodeHandle>) SimpleNetwo ey,NodeHandle> {	>)al[4].overlay.successorList.successorMap	L d 61
(sto map< ral std: truct	d::map <ove COverlayKey,N Fields ::map<overlayk t pair<*,*> su</overlayk </ove 	rlayKey,NodeHandle: odeHandle>) SimpleNetwo ey,NodeHandle> { ccessorMap[0] = 10b061e 2-2614075-54042027	>)al[4].overlay.successorList.successorMap ork.overlayTerminal[4].overlay.successorList.successorMap (ptr0x91 e987b34762cdd6fb8933f51555749187c6 ==> 1.0.0.3;1024	L d 61
(sto map< ral std; truct	d::map <ove COverlayKey,N Fields ::map<overlayk t pair<*,*> su 11762ae7 t pair<*,*> su</overlayk </ove 	rlayKey,NodeHandle: odeHandle>) SimpleNetwo ey,NodeHandle> { ccessorMap[0] = 10b061e a2e3f11057ba75d5cfd042077 ccessorMap[1] = 1448df5	>)al[4].overlay.successorList.successorMap prk.overlayTerminal[4].overlay.successorList.successorMap (ptr0x91 e987b34762cdd6fb8933f51555749187c6 ==> 1.0.0.3:1024 7738c32e 36 5fde193727b750b8b5c48c649fc69ab241 ==> 1.0.0.1:1024	L d 61
(sto map< ral std; truct	::map <ove OverlayKey,N Fields ::map<overlayk t pair<*,*> su 11762ae7 t pair<*,*> su 150ea85d</overlayk </ove 	rlayKey,NodeHandle: odeHandle>) SimpleNetwo ey,NodeHandle> { ccessorMap[0] = 10b061e a2e3f11057ba75d5cfd042077 ccessorMap[1] = 1448df5 f949e0d54134330260679151c	>)al[4].overlay.successorList.successorMap ork.overlayTerminal[4].overlay.successorList.successorMap (ptr0x91 e987b34762cdd6fb8933f51555749187c6 ==> 1.0.0.3:1024 7738c32e 36 5fde193727b750b8b5c48c649fc69ab241 ==> 1.0.0.1:1024 c941eda9 12	Id61
(sto map< ral std; truct truct	CoverlayKey,N Fields ::map <overlayk t pair<*,*> su 11762ae7 t pair<*,*> su 150ea85d t pair<*,*> su 385a5a44</overlayk 	rlayKey,NodeHandle: odeHandle>) SimpleNetwo ey,NodeHandle> { ccessorMap[0] = 10b061e a2e3f11057ba75d5cfd042077 ccessorMap[1] = 1448df5 f949e0d54134330260679151c ccessorMap[2] = 3794914 5163edcd8c2d2c10bd8ff8845	>)al[4].overlay.successorList.successorMap ork.overlayTerminal[4].overlay.successorList.successorMap (ptr0x91 e987b34762cdd6fb8933f51555749187c6 ==> 1.0.0.3;1024 7738c32e 36 5fde193727b70b8b5c48c649fc69ab241 ==> 1.0.0.1;1024 c941eda9 12 46363344200249b1c421b4cbd25168428b ==> 1.0.0.9;1024 54067dF3 108	ld61
(sto map< eral : std; truct truct	::map <ove OverlayKey,N Fields ::map<overlayk t pair<*,*> su 11762ae7 t pair<*,*> su 150ea85d t pair<*,*> su 385a5a44 t pair<*,*> su</overlayk </ove 	rlayKey,NodeHandle: odeHandle>) SimpleNetwo ey,NodeHandle> { ccessorMap[0] = 10b061e a2e3f11057ba75d5cfd042077 ccessorMap[1] = 1448df5 f949e0d54134330260679151c ccessorMap[2] = 3794914 5163edcd8c2d2c10bd8ff8845 ccessorMap[3] = 39b3ccc	<pre>>)al[4].overlay.successorList.successorMap</pre>	d61
(sto map< eral ` truct truct truct	CoverlayKey,N Fields ::map <overlaykey,n pair<*,*> su 11762ae7 t pair<*,*> su 150ea85d t pair<*,*> su 385a5a44 t pair<*,*> su 37995c6</overlaykey,n 	rlayKey,NodeHandle: odeHandle>) SimpleNetwo ey,NodeHandle> { ccessorMap[0] = 10b061e a2e3f11057ba75d5cfd042077 ccessorMap[1] = 1448df5 f949e0d54134330260679151c ccessorMap[2] = 3794914 5163edcd8c2d2c10bd8ff8845 ccessorMap[3] = 39b3ccc 5ade30afac968783e39e184fe ccessorMap[4] = 93233c2	<pre>>)al[4].overlay.successorList.successorMap</pre>	id61
(stc map≺ rral ` std; truct truct truct	::map <ove OverlayKey,N Fields ::map<overlayk t pair<*,*> su 11762ae7 t pair<*,*> su 150ea85d t pair<*,*> su 385a5a44 t pair<*,*> su 387995c6 t pair<*,*> su 33690527</overlayk </ove 	rlayKey,NodeHandle: odeHandle>) SimpleNetwo ey,NodeHandle> { ccessorMap[0] = 10b061e a2e3f11057ba75d5cfd042077 ccessorMap[2] = 3794914 5163edcd8c2d2c10bd8ff8845 ccessorMap[3] = 39b3ccc 5ade30afac968783e39e184fe ccessorMap[4] = 93233c2 7301d5d3c0ec9ddaafd2a9ecb	<pre>>)al[4].overlay.successorList.successorMap</pre>	1 d 61
(sto map< eral ` std; truct truct truct truct	::map <ove OverlayKey,N Fields :map<overlayk t pair<*,*> su 11762ae7 t pair<*,*> su 150ea85d t pair<*,*> su 385a5a44 t pair<*,*> su 37995c6 t pair<*,*> su 93e90527 t pair<*,*> su 93e90527</overlayk </ove 	rlayKey,NodeHandle: odeHandle>) SimpleNetwo ey,NodeHandle> { ccessorMap[0] = 10b061e a2e3f11057ba75d5cfd042077 ccessorMap[1] = 1448df5 f949e0d54134330260679151c ccessorMap[2] = 3794914 5163edcd8c2d2c10bd8ff8845 ccessorMap[3] = 39b3ccc 5ade30afac968783e39e184fe ccessorMap[3] = 39b3ccc 5ade30afac968783e39e184fe ccessorMap[3] = 93233c2 7301d5d3c0e9ddaafd2a9ecb ccessorMap[5] = b265d08 491b86e359023682a0c9931c6	<pre>>)al[4].overlay.successorList.successorMap</pre>	1 d 614
(stc map< eral ` std; truct truct truct truct truct	::map <ove OverlayKey,N Fields ::map<overlayk t pair<*,*> su 11762ae7 t pair<*,*> su 150ea85d t pair<*,*> su 385a5a44 t pair<*,*> su 387995c6 t pair<*,*> su 332995c7 t pair<*,*> su 532998d t pair<*,*> su</overlayk </ove 	rlayKey,NodeHandle: odeHandle>) SimpleNetwo ey,NodeHandle> { ccessorMap[0] = 10b061e a2e3f11057ba75d5cfd042077 ccessorMap[1] = 1448df5 f949e0d54134330260679151c ccessorMap[2] = 3794944 5163edcd8c2d2c10bd8ff8845 ccessorMap[2] = 3794323c2 5ade30afac968783e39e184fe ccessorMap[4] = 93233c2 7301d5d3c0ec9ddaafd2a9ecb ccessorMap[5] = b265d08 491b86e359023682a0c99a1c6 ccessorMap[6] = d1d8a61	<pre>>)al[4].overlay.successorList.successorMap</pre>	1 d 614
(stc map< iral std; truct truct truct truct truct	::map <ove OverlayKey,N Fields ::map<overlayk t pair<*,*> su 11762ae7 t pair<*,*> su 150ea85d t pair<*,*> su 385a5a44 t pair<*,*> su 37995c6 t pair<*,*> su 93e90527 t pair<*,*> su b32b998d t pair<*,*> su d29e6f14</overlayk </ove 	rlayKey,NodeHandle: odeHandle>) SimpleNetwo ey,NodeHandle> { ccessorMap[0] = 10b061e a2e3f11057ba75d5cfd042077 ccessorMap[1] = 1448df5 f949e0d54134330260679151c ccessorMap[2] = 3794914 5163edcd8c2d2c10bd8ff8845 ccessorMap[2] = 3794914 5163edcd8c2d2c10bd8ff8845 ccessorMap[3] = 39b3ccc 5ade30afac968783e339e184fe ccessorMap[4] = 93233c2 7301d5d3c0ec9ddaafd2a9ecb ccessorMap[5] = b265d08 491b86e359023682a0c99a1c6 ccessorMap[6] = d1d8a61 02b883370249d54edc26a343f ccessorMap[7] = dt6c005	<pre>>)al[4].overlay.successorList.successorMap</pre>	L d 61.

Stephan Krause

11

OverSim, GI07, Anchorage, AK, USA

Overlay Protocols

- Implemented overlay protocols
 - Structured: Chord, Pastry, Koorde, Broose
 - Unstructured: GIA
 - Gaming protocols/Multicast: VAST
- Base class for structured overlays to facilitate implementation of new overlay protocols
 - Overlay message handling (RPC, statistics)
 - Generic lookup function
 - Bootstrapping support
 - Overlay topology visualization

OverSim Performance Evaluation

- Compare simulation speed with P2PSim
- Underlay model:
 - P2PSim and OverSim/SimpleUnderlay:
 - Constant delay of 50 msec
 - OverSim/INETUnderlay:
 - Randomly generated backbone (40 routers)
- Number of nodes: 1,000, 10,000 and 100,000
- Simulated time period: 1,000 seconds
- Overlay: Chord
 - Stabilize interval: 20 sec
 - Fix fingers interval: 120 sec
 - Each node sends a query every 10 sec

Performance Evaluation Results



ELEMATICS

OverSim, GI07, Anchorage, AK, USA

Validation: Chord Results





Stephan Krause

15

OverSim, GI07, Anchorage, AK, USA

Institute of Telematics Universität Karlsruhe (TH)

www.tm.uka.de



Conclusion & Future Work

OverSim is a new, modular overlay simulator

- Focus on scalability and flexibility
- Various underlay models for maximum speed, maximum precision or real network use
- Many implemented overlay protocols
- Easily extendable

• Future work:

- New overlay protocols and applications
 Kademlia, Scribe, i3, P2PSIP
- Import models from topology generators
- Import datasets from internet latency measurements

http://www.oversim.org/ Download now! Try now! Contribute now!



Universität Karlsruhe (TH) Research University • founded 1825

TELEMATICS Institute of Telematics