

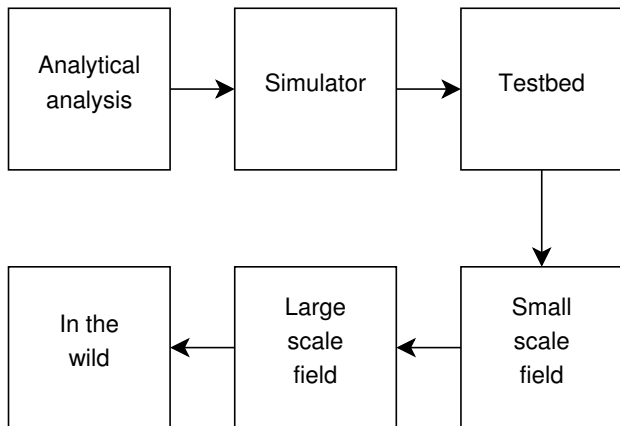
Experimentation with CCN

Daniel Camara, Frederic Urbani,
Mathieu Lacage, Thierry Turletti, Walid Dabbous

INRIA, Planète

Sept, 13th 2012

Idealized Protocol Evaluation



A Development Tool

A Development Tool

Development of Production Code:

A Development Tool

Development of Production Code:
Debugging

A Development Tool

Development of Production Code:

Debugging

Validation

A Development Tool

Development of Production Code:

Debugging

Validation

Automated Testing (fast, deterministic)

Testbeds ?

Testbeds ?

No:

Testbeds ?

No:

Deployment too slow and unreliable

Testbeds ?

No:

Deployment too slow and unreliable

Not completely deterministic

Testbeds ?

No:

Deployment too slow and unreliable

Not completely deterministic

Distributed debugging is hard

Testbeds ?

No:

Deployment too slow and unreliable

Not completely deterministic

Distributed debugging is hard

Distributed tracing is hard

Traditional Simulators ?

Traditional Simulators ?

No:

Traditional Simulators ?

No:

Not the real code

Proposed Solution

Proposed Solution

Simulate the CCNx code before deployment:

Proposed Solution

Simulate the CCNx code before deployment:

Reuse existing layer 1/2/3/4 models from ns-3

Proposed Solution

Simulate the CCNx code before deployment:

- Reuse existing layer 1/2/3/4 models from ns-3

- Trivial to deploy multiple network nodes

Proposed Solution

Simulate the CCNx code before deployment:

- Reuse existing layer 1/2/3/4 models from ns-3

- Trivial to deploy multiple network nodes

- Easy whole-system debugging

Proposed Solution

Simulate the CCNx code before deployment:

- Reuse existing layer 1/2/3/4 models from ns-3

- Trivial to deploy multiple network nodes

- Easy whole-system debugging

- Easy whole-system tracing and analysis

Proposed Solution

Simulate the CCNx code before deployment:

- Reuse existing layer 1/2/3/4 models from ns-3

- Trivial to deploy multiple network nodes

- Easy whole-system debugging

- Easy whole-system tracing and analysis

- Optionally reuse kernel layer 3/4 stack

Proposed Solution

Simulate the CCNx code before deployment:

- Reuse existing layer 1/2/3/4 models from ns-3

- Trivial to deploy multiple network nodes

- Easy whole-system debugging

- Easy whole-system tracing and analysis

- Optionally reuse kernel layer 3/4 stack

- Much better scalability than testbeds/VMs

How do I use it ?

How do I use it ?

Roughly:

- Download/install ns-3

- Download/install ns-3 DCE module

- Recompile CCNx with magic option

- Write simulation script

- Run

- Analyse traces

How do I use it ?

Roughly:

- Download/install ns-3

- Download/install ns-3 DCE module

- Recompile CCNx with magic option

- Write simulation script

- Run

- Analyse traces

The details: <http://goo.gl/yfgwZ>

What we would like to learn

What we would like to learn

If you tried DCE/CCNx:

What sucks about it ?

How can we improve it ?

How can we write more test scripts for CCNx ?

What we would like to learn

If you tried DCE/CCNx:

What sucks about it ?

How can we improve it ?

How can we write more test scripts for CCNx ?

If you did not try it:

Why ?

What is missing to make you try it ?

Thank you!

Contact:

daniel.camara@inria.fr

frederic.urbani@inria.fr

mathieu.lacage@cutebugs.net

Related Work

	NDN over ONL	NEPI	ccnSim	CCNPL-Sim	DCE	ndnSim
Type	testbed	testbed	sim	sim	sim	sim
Real code	+	+	-	-	+	-
Scalability	-	-	+++	?	+	++
Deployment	-	+	++	++	+	++
Debugger	-	-	+	+	+	+
Tracing	-	+	+	+	+	+

Scalability

Memory:

Fixed, 1MB/node ccnx

Fixed, 0.02MB/node ns-3 TCP/IP

Fixed, 0.5MB/node linux TCP/IP

Linear in number of bytes of packets in flight

CPU:

Real time limit: 20 nodes, 200K file transfer