Direct Code Execution Tutorial

Walkthrough

PLANETE Group
Inria Sophia Antipolis
Méditerranée

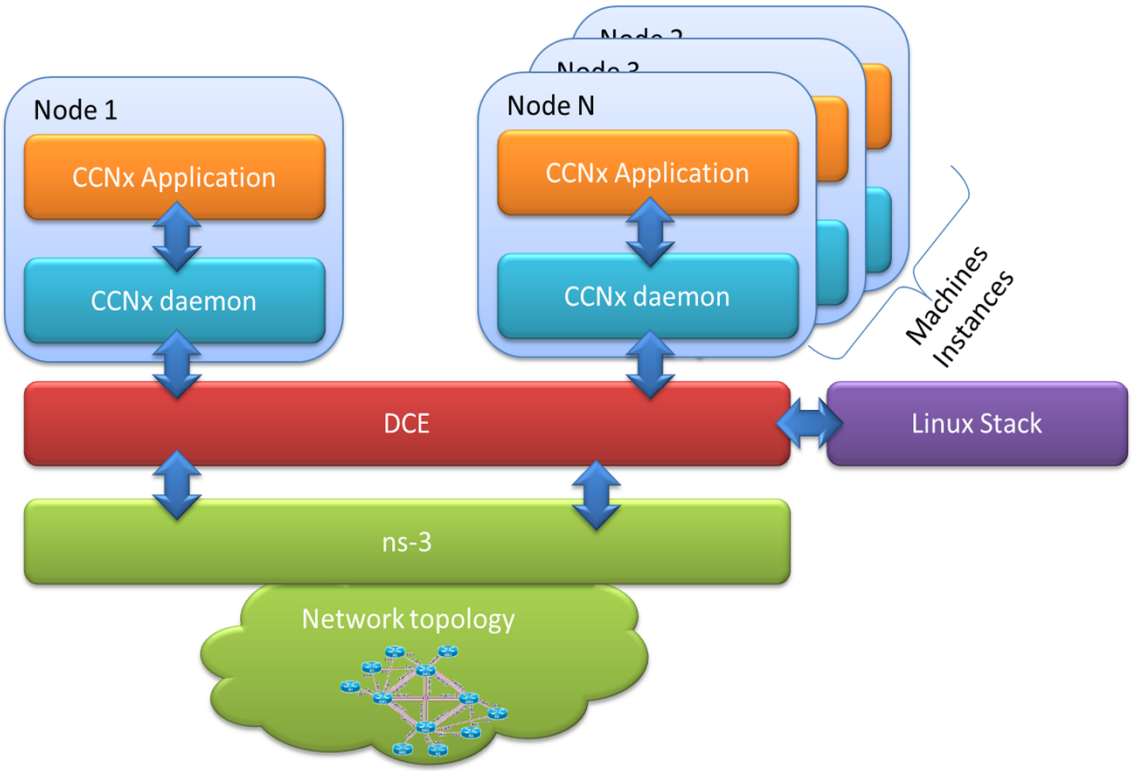
Daniel Camara, Hajime Tazaki, Frederic Urbani

13 March 2013

What is DCE

- **DCE (*Direct Code Execution*) is a generic tool that can be used with any available ICN code to provide a repeatable and fully controllable test environment for ICNs**
- **DCE can help on the development process by:**
  - Reusing existing layer 1/2/3/4 models from ns-3
  - Optionally reusing kernel layer 3/4 stack
  - Reusing existing code from third party implementations (e.g. Quagga, CCNx, thttp, wget, iperf….)
  - Being trivial to deploy multiple network nodes
  - Enabling the use of complex deployments(e.g. Mix of WiFi/WiMax/LTE setups)
  - Being easy to perform whole-system debugging
  - Being easy to perform whole-system tracing and analysis
  - Providing better scalability than testbeds/VMs

Architecture



- **The framework takes care of the virtualization of the CCNx instances and uses ns-3 to handle the topology and packet transmission issues**
  - CCNx can run either over the simulated ns-3 stack, what is faster, or over a real Linux stack, what is more realistic

Hands on, what do we need

- **What do you need to start using the framework!**
  - ns-3 – The network simulator
    - http://www.nsnam.org/
  - DCE - To integrate the protocols and the network simulator
    - http://www.nsnam.org/projects/direct-code-execution/
  - Applications
    - iperf, wget, thttpd

  * All software must be re-compiled with –fpic and linked with –pie to generate the code with Position Independent Code (PIC) and permit context switch

- **To make things easier**
  - Bake – Installation tool
    - http://planete.inria.fr/software/bake/index.html
  - Mercurial – source control management tool
    - http://mercurial.selenic.com/
  - Python – for running bake
    - www.python.org

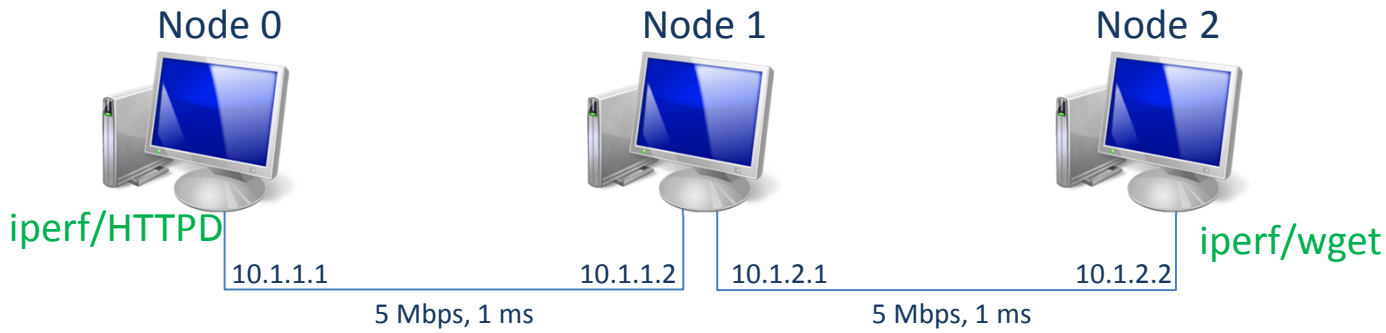
* The presented example can be found at http://www-sop.inria.fr/members/Frederic.Urbani/in-pro-gress/howto-ccn-exp1.html and the example code can be found in the root of DCE at dce/myscripts/ccn-exp1

The plan is

- **The plan is to present**
  - Installation
  - Examples of:
    - iperf with ns-3 stack
    - www server and wget with ns-3 stack
    - iperf with Linux stack
    - Van Jacobson's paper CCNx experiment

The shared scenario

- The shared scenario is a simple three nodes network



Step by step example
- Installing the required software

* Into a Linux machine

1) > mkdir dce_tutorial; cd dce_tutorial
2) > hg clone http://code.nsnam.org/bake bake
3) > export BAKE_HOME=`pwd`/bake
4) > export PATH=$PATH:$BAKE_HOME
5) > export PYTHONPATH=$PYTHONPATH:$BAKE_HOME
6) > mkdir DCE; cd DCE
7) > bake.py configure -e dce-ns3
8) > bake.py install
9) >. bakeSetEnv.sh

Step by step example
- What we need to do!

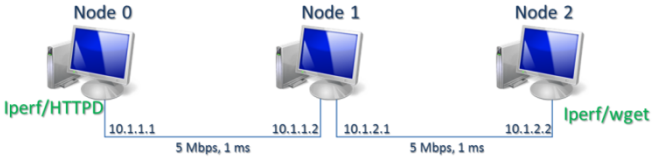
1. Create the nodes
2. Create stack
3. Create devices
4. Set addresses
5. Connect devices
6. Create DCE
7. Configuration the applications to run
8. Set start time for server and client
9. Set simulation time
10. Start simulation

Step by step example
- What we need to do!

1) Create the nodes
2) Create stack
3) Create devices
4) Set addresses
5) Connect devices
6) Create DCE
7) Configuration the applications to run
8) Set start time for server and client
9) Set simulation time
10) Start simulation

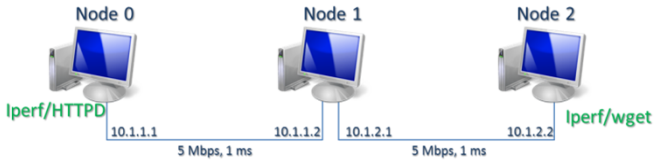
Standard ns-3 procedures
DCE specific

Step by step example
- iperf with ns-3 stack (I)




```
int main (int argc, char *argv[])
{
 // Node Container creation
 NodeContainer nodes;
 nodes.Create (3);

 // Linux stack creation
 InternetStackHelper stack;
 stack.Install (nodes);

 // For real time
 // GlobalValue::Bind ("SimulatorImplementationType", StringValue ("ns3::RealtimeSimulatorImpl"));
 // GlobalValue::Bind ("ChecksumEnabled", BooleanValue (true));

 // Device and channel  creation
 PointToPointHelper p2p;
 p2p.SetDeviceAttribute ("DataRate", StringValue ("5Mbps“));
 p2p.SetChannelAttribute ("Delay", StringValue ("1ms"));
```

Step by step example
- iperf with ns-3 stack (II)




```
//  Node0-Node1 setup
 Ipv4AddressHelper address;
 address.SetBase ("10.1.1.0", "255.255.255.252"); //  Node0-Node1 addresses

  NetDeviceContainer devices;
 devices = p2p.Install (nodes.Get (0), nodes.Get (1)); // connecting nodes
 Ipv4InterfaceContainer interfaces = address.Assign (devices); //  assign addresses

 //  Node1-Node2 setup
 devices = p2p.Install (nodes.Get (1), nodes.Get (2)); // connecting nodes
 address.SetBase ("10.1.2.0", "255.255.255.252"); //  Node1-Node2 addresses
 interfaces = address.Assign (devices); //  assign addresses

 // setup ip routes
 Ipv4GlobalRoutingHelper::PopulateRoutingTables ();
```

Step by step example
- iperf with ns-3 stack (III)

```
DceManagerHelper dceManager;
dceManager.Install (nodes);

DceApplicationHelper dce;
ApplicationContainer apps;
 dce.SetStackSize (1 << 20); // 1MB stack

dce.SetBinary ("iperf"); // Launch iperf client on node 0
dce.ResetArguments ();  // clean arguments
dce.ResetEnvironment (); // clean environnent
dce.AddArgument ("-c");  // client
dce.AddArgument ("10.1.2.2"); //target machine address
dce.AddArgument ("-i"); // interval
dce.AddArgument ("1");
dce.AddArgument ("--time"); // how long
dce.AddArgument ("10");
 apps = dce.Install (nodes.Get (0)); //install application
apps.Start (Seconds (0.7)); //start at 0.7 simulation time
apps.Stop (Seconds (20)); //stop at 20s        simulation time

dce.SetBinary ("iperf"); // Launch iperf server on node 2
dce.ResetArguments (); // clean arguments
dce.ResetEnvironment (); // clean environnent
dce.AddArgument ("-s"); // server
dce.AddArgument ("-P"); // number of paralell servers
dce.AddArgument ("1");
apps = dce.Install (nodes.Get (2));
apps = dce.Install (nodes.Get (2));
apps.Start (Seconds (0.6));
```

DCE Setup

Step by step example
- iperf with ns-3 stack (IV)

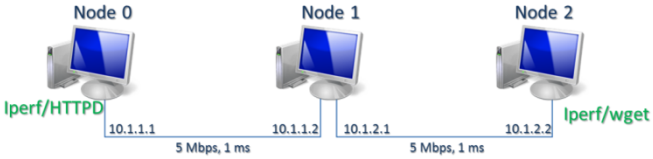

```
 // Simulation stop time
 Simulator::Stop (Seconds (40.0));

 // Run
 Simulator::Run ();

 // Stop
 Simulator::Destroy ();

 return 0;
}
```

Step by step example – iperf, ns-3

- **Generated**
  - elf-cache – program files
  - exitprocs – execution process information
  - files-0 files-2 – execution filesystem

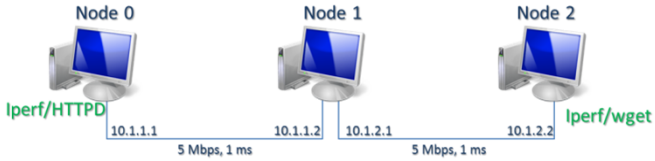
- **files-x**
  - var – “/root” of the machine
  - files-x/var/log/<pid>/
    - cmdline – command executed
    - status – execution information
    - stderr – standard error output
    - stdout – standard output
    - syslog – syslog output

Step by step example
- HTTP with ns-3 stack (I)




```
int main (int argc, char *argv[])
{
 // Node Container creation
 NodeContainer nodes;
 nodes.Create (3);

 // Linux stack creation
 InternetStackHelper stack;
 stack.Install (nodes);

 // For real time
 // GlobalValue::Bind ("SimulatorImplementationType", StringValue ("ns3::RealtimeSimulatorImpl"));
 // GlobalValue::Bind ("ChecksumEnabled", BooleanValue (true));

 // Device and channel  creation
 PointToPointHelper p2p;
 p2p.SetDeviceAttribute ("DataRate", StringValue ("5Mbps“));
 p2p.SetChannelAttribute ("Delay", StringValue ("1ms"));
```

Step by step example
- HTTP with ns-3 stack (II)




```
//  Node0-Node1 setup
 Ipv4AddressHelper address;
 address.SetBase ("10.1.1.0", "255.255.255.252"); //  Node0-Node1 addresses

  NetDeviceContainer devices;
 devices = p2p.Install (nodes.Get (0), nodes.Get (1)); // connecting nodes
 Ipv4InterfaceContainer interfaces = address.Assign (devices); //  assign addresses

 //  Node1-Node2 setup
 devices = p2p.Install (nodes.Get (1), nodes.Get (2)); // connecting nodes
 address.SetBase ("10.1.2.0", "255.255.255.252"); //  Node1-Node2 addresses
 interfaces = address.Assign (devices); //  assign addresses

 // setup ip routes
 Ipv4GlobalRoutingHelper::PopulateRoutingTables ();
```

Step by step example
- HTTP with ns-3 stack (III)

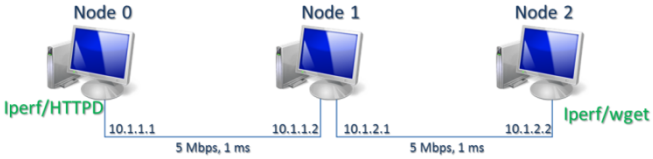
```
// Launch the server HTTP
 dce.SetBinary ("thttpd");
 dce.ResetArguments ();  // clean arguments
 dce.ResetEnvironment (); // clean environnent
 dce.SetUid (1);  // Set httpd for super user execution
 dce.SetEuid (1);
 apps = dce.Install (nodes.Get (0)); // install http daemon
 apps.Start (Seconds (1)); // start time

 // Launch the client WGET
 dce.SetBinary ("wget");
dce.ResetArguments ();  // clean arguments
 dce.ResetEnvironment (); // clean environnent
 dce.AddArgument ("-r"); // recursive wget
 dce.AddArgument ("http://10.1.1.1/index.html");
 apps = dce.Install (nodes.Get (2));
 apps.Start (Seconds (2)); // start time
```

DCE Setup

Step by step example
- HTTPf with ns-3 stack (IV)

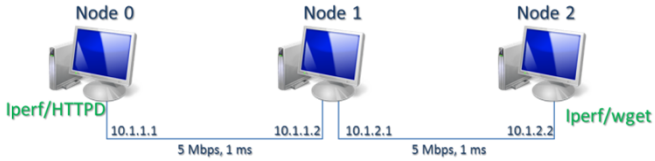

```
 // Simulation stop time
 Simulator::Stop (Seconds (40.0));

 // Run
 Simulator::Run ();

 // Stop
 Simulator::Destroy ();

 return 0;
}
```

Step by step example
- iperf with linux stack (I)




```
int main (int argc, char *argv[])
{
 // Node Container creation
 NodeContainer nodes;
 nodes.Create (3);

 // Linux stack creation
 dceManager.SetNetworkStack ("ns3::LinuxSocketFdFactory", "Library", StringValue ("liblinux.so"));
 LinuxStackHelper stack;
 stack.Install (nodes);

// For real time
 // GlobalValue::Bind ("SimulatorImplementationType", StringValue ("ns3::RealtimeSimulatorImpl"));
 // GlobalValue::Bind ("ChecksumEnabled", BooleanValue (true));

 // Device and channel  creation
 PointToPointHelper p2p;
 p2p.SetDeviceAttribute ("DataRate", StringValue ("5Mbps“));
 p2p.SetChannelAttribute ("Delay", StringValue ("1ms"));
```

Step by step example
- iperf with linux stack (II)




```
//  Node0-Node1 setup
 Ipv4AddressHelper address;
 address.SetBase ("10.1.1.0", "255.255.255.252"); //  Node0-Node1 addresses

  NetDeviceContainer devices;
 devices = p2p.Install (nodes.Get (0), nodes.Get (1)); // connecting nodes
 Ipv4InterfaceContainer interfaces = address.Assign (devices); //  assign addresses

 //  Node1-Node2 setup
 devices = p2p.Install (nodes.Get (1), nodes.Get (2)); // connecting nodes
 address.SetBase ("10.1.2.0", "255.255.255.252"); //  Node1-Node2 addresses
 interfaces = address.Assign (devices); //  assign addresses

 // setup ip routes
 Ipv4GlobalRoutingHelper::PopulateRoutingTables ();
```

Step by step example
- iperf with linux stack (III)

```
DceManagerHelper dceManager;
dceManager.Install (nodes);

DceApplicationHelper dce;
ApplicationContainer apps;
 dce.SetStackSize (1 << 20); // 1MB stack

dce.SetBinary ("iperf"); // Launch iperf client on node 0
dce.ResetArguments ();  // clean arguments
dce.ResetEnvironment (); // clean environnent
dce.AddArgument ("-c");  // client
dce.AddArgument ("10.1.2.2"); //target machine address
dce.AddArgument ("-i"); // interval
dce.AddArgument ("1");
dce.AddArgument ("--time"); // how long
dce.AddArgument ("10");
 apps = dce.Install (nodes.Get (0)); //install application
apps.Start (Seconds (0.7)); //start at 0.7 simulation time
apps.Stop (Seconds (20)); //stop at 20s        simulation time

dce.SetBinary ("iperf"); // Launch iperf server on node 2
dce.ResetArguments (); // clean arguments
dce.ResetEnvironment (); // clean environnent
dce.AddArgument ("-s"); // server
dce.AddArgument ("-P"); // number of paralell servers
dce.AddArgument ("1");
apps = dce.Install (nodes.Get (2));
apps = dce.Install (nodes.Get (2));
apps.Start (Seconds (0.6));
```

DCE Setup

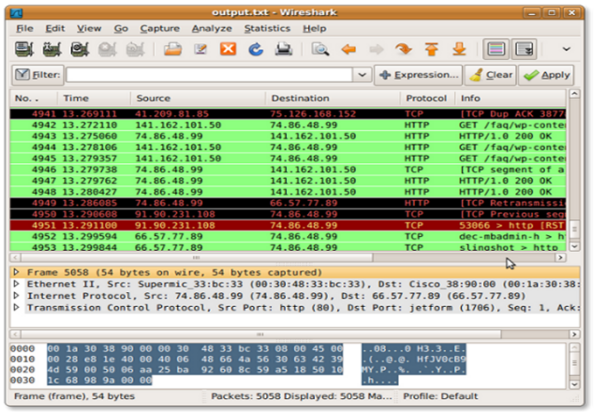
(Similar to the ns-3 stack one)

Step by step example
- iperf with linux stack (IV)

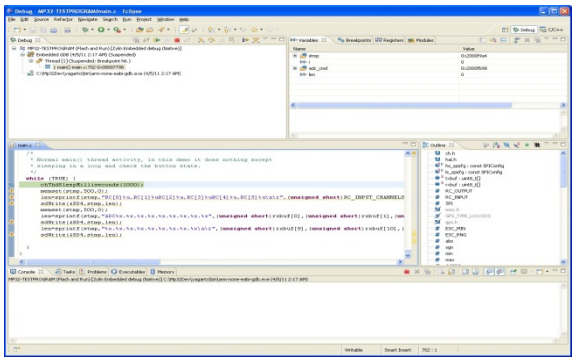

```
 // Simulation stop time
 Simulator::Stop (Seconds (40.0));

 // Run
 Simulator::Run ();

 // Stop
 Simulator::Destroy ();

 return 0;
}
```

Log and debug

- We can use the log facilities of ns-3 to create, for example pcap files

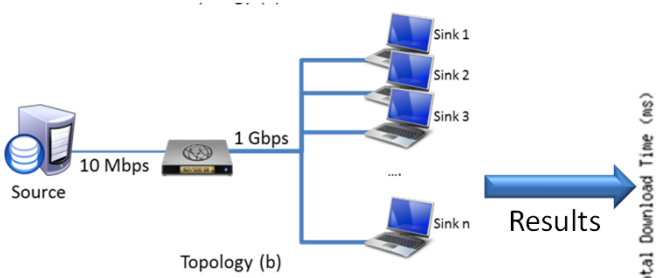


- Standard GDB

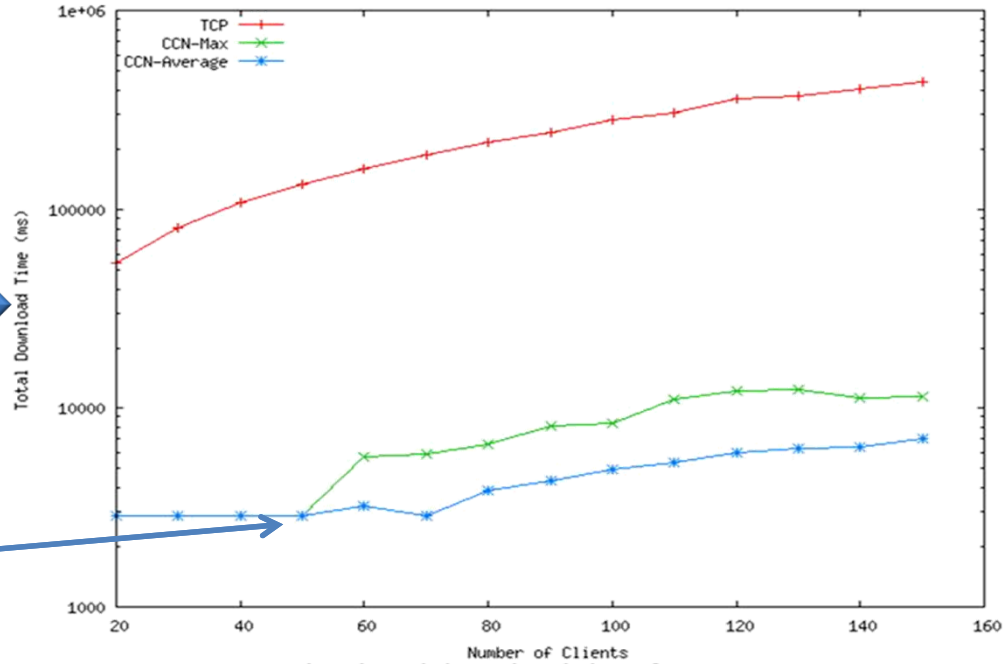


CCNx Proof of concept

- To evaluate the power of the proposed framework we implemented, and extended, the topologies proposed in the Jacobson's seminal CCNx [1] paper



Results



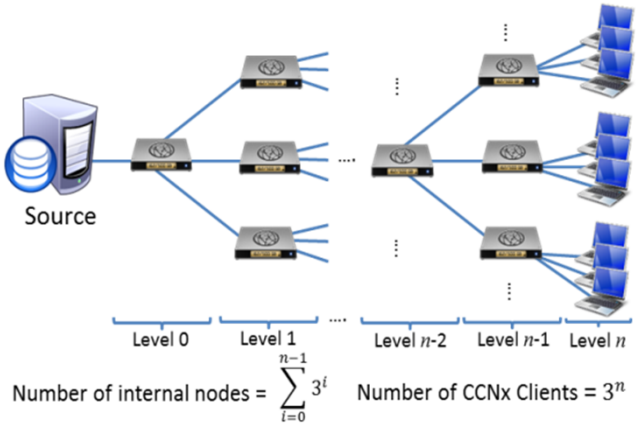
Simulated download time for 3KByte files in the scenario for topology (b) (log scale)

Increasing the number of nodes, we managed to observe the increasing in the CCNx downloading time, what is not possible with small topologies

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, R. L. Braynard (PARC) Networking Named Content, CoNEXT 2009, Rome, December, 2009

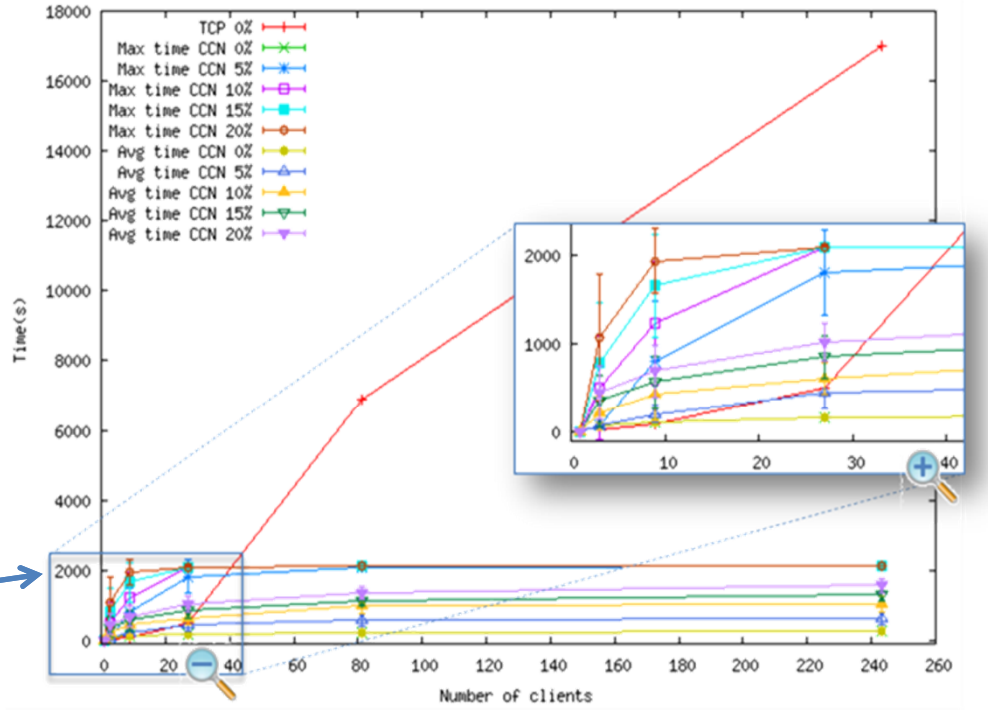
CCNx Proof of concept

- **Extended ternary three scenario with random link bottlenecks**



Topology (c)

Results



Simulated download time for 3KByte files for topology (c) with different bottlenecks percentage

We also managed to easily extend the topology to a tree like topology to observe the CCNx behavior on other more interesting scenarios, adding random bottleneck over the network

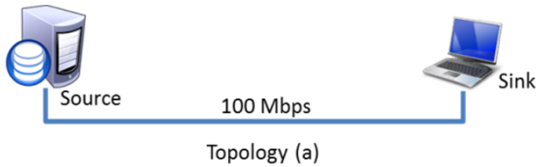
Step by step CCNx example - What we need to do!

What do you need to start using the framework!
- CCNx – The ICN implementation we will use
  https://www.ccnx.org/

1) Create the nodes
2) Create the files to transmit
3) Create the CCNx configuration files
4) Connect the nodes
5) Configure nodes
6) Create stack
7) Define the topology and configure routes
8) Configure environment for CCNx
9) Define the nodes that will receive the CCNx server and client
10) Set start time for server and client
11) Set simulation time
12) Start simulation

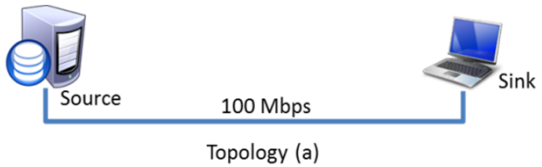
Standard ns-3 procedures
Standard CCNx procedures
DCE specific

Step by step example
- How we do the required steps (create nodes)




```
int main (int argc, char *argv[]) {
  //Initalization, default values
  std::string animFile = "NetAnim.xml";
  std::string packetSize = "1024";
  std::string ccnTransport = "udp"; // or tcp
  std::string dataRate = "100Mbps";
  bool ccnTest = 0;

  // Create nodes (ns-3)
  NodeContainer nodes;
  nodes.Create (2); // two nodes
```

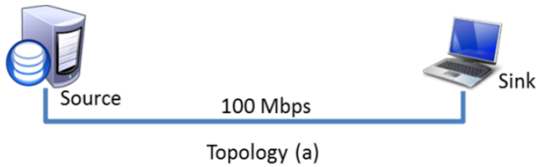
Step by step example
- How we do the required steps (Create files)



Topology (a)

```
// First create a 6 mb file filled with A.
FILE *f = fopen ("/tmp/bigone", "w");
char buffer[1024];  memset (buffer, 'A', sizeof(buffer));
for (int i = 0; i < (6 * 1024); i++)    {
   fwrite (buffer, sizeof(buffer), 1, f);
}
fclose (f);

// 2nd create configuration file for the CCN daemon running on second node
// content : add ccnx:/ [udp|tcp]  10.1.1.1 9695
//  ---> Forward interrests using udp or tcp to the other node.  std::ofstream osf3
("/tmp/ccnd1.conf", std::fstream::trunc);
osf3 << "add ccnx:/ " << ccnTransport << " 10.1.1.1 9695" << std::endl;  osf3.close ();

// and last create configuration file for the CCN daemon running on first node
// content : add ccnx:/ [udp|tcp]  10.1.1.2 9695
//  ---> Forward interrests using udp or tcp to the other node.  std::ofstream osf4
("/tmp/ccnd0.conf", std::fstream::trunc);  osf4 << "add ccnx:/ " << ccnTransport << " 10.1.1.2 9695" << std::endl;
osf4.close ();
```

Step by step example
- How we do the required steps (Connect nodes)

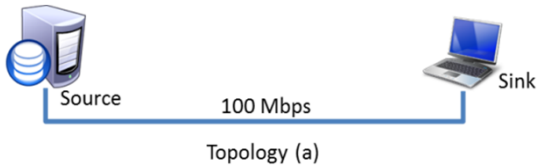


Topology (a)

```
// Connect nodes
PointToPointHelper pointToPoint; // Connection point to point
// Configure channel
pointToPoint.SetDeviceAttribute ("DataRate", StringValue (dataRate));
pointToPoint.SetChannelAttribute ("Delay", StringValue ("1ms"));

// install connection among nodes
NetDeviceContainer devices;
devices = pointToPoint.Install (nodes);
// configure connection
devices.Get (0)->SetMtu (10000);
devices.Get (1)->SetMtu (10000);
```

```
// Create stack
DceManagerHelper dceManager;  // DCE will manage the network stack
InternetStackHelper stack;  // Default ns-3 stack
stack.Install (nodes);
```

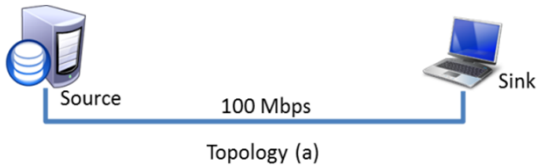
Step by step example
- How we do the required steps (Connect nodes)



Topology (a)

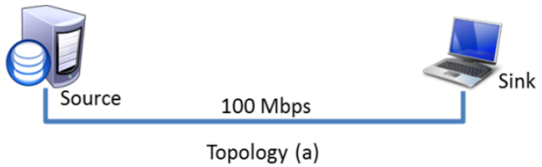
```
// Create the machines addresses
Ipv4AddressHelper address;
address.SetBase ("10.1.1.0", "255.255.255.252");
Ipv4InterfaceContainer interfaces = address.Assign (devices);

 // name machines
Names::Add ("NODE_Zero", nodes.Get (0));
Names::Add ("NODE_One", nodes.Get (1));

// setup ip routes
Ipv4GlobalRoutingHelper::PopulateRoutingTables (); // standard ns-3 method to populate routing tables
dceManager.Install (nodes);
```

Step by step example
- How we do the required steps (Configure CCNx)




```
// creates the ccn
CcnClientHelper dce;
ApplicationContainer apps, putter, getter;
dce.SetStackSize (1 << 20);

// Launch ccnd on each nodes.
dce.SetBinary ("ccnd");
dce.ResetArguments ();
dce.ResetEnvironment ();
dce.AddEnvironment ("CCND_CAP", "50000");
dce.AddEnvironment ("CCN_LOCAL_PORT", "9695");
dce.AddEnvironment ("CCND_CAP", "");
dce.AddEnvironment ("CCND_DEBUG", "2");
dce.AddEnvironment ("CCND_AUTOREG", "");
dce.AddEnvironment ("CCND_AUTOREG", "");
dce.AddEnvironment ("CCND_LISTEN_ON", "");
dce.AddEnvironment ("CCND_MTU", "");
dce.AddEnvironment ("CCND_LOCAL_SOCKNAME", "");
dce.AddEnvironment ("CCND_DATA_PAUSE_MICROSEC", "");
dce.AddEnvironment ("CCND_KEYSTORE_DIRECTORY", "");
```

Step by step example
- How we do the required steps (Configure CCNx Nodes)



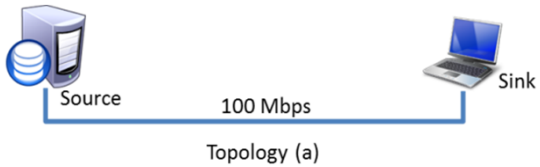

```
// Define CCN start times
apps = dce.Install (nodes.Get (0));
apps.Start (Seconds (0.03));

apps = dce.Install (nodes.Get (1));
apps.Start (Seconds (0.03));
```

```
// Configure ccn daemon on node 0 to forward interrests to node 1
dce.SetBinary ("ccndc");     // Define DCE binary
dce.ResetArguments ();
dce.ResetEnvironment ();
dce.AddEnvironment ("HOME", "/root");
// USE TO FIND keystore under $HOME/.ccnx/.ccnx_keystore
dce.AddArgument ("-f");
dce.AddArgument ("/tmp/ccnd0.conf");
dce.AddFile ("/tmp/ccnd0.conf", "/tmp/ccnd0.conf");
apps = dce.Install (nodes.Get (0));
apps.Start (Seconds (0.04));
```

Step by step example

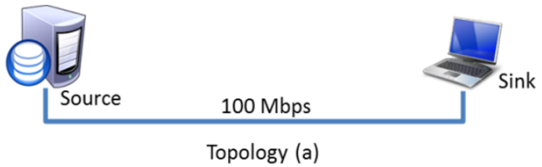
- How we do the required steps (CCNx server and client)

```
// PUBLISH DATA FILE ON NODE 1
dce.ResetArguments ();
dce.ResetEnvironment ();
dce.SetBinary ("ccnsendchunks");
dce.AddEnvironment ("HOME", "/root");
dce.SetStdinFile ("/tmp/bigone");
dce.AddFile ("/tmp/bigone", "/tmp/bigone");
dce.AddArgument ("-b");
dce.AddArgument (packetSize);
dce.AddArgument ("ccnx:/DATAFILE");
putter = dce.Install (nodes.Get (1));
putter.Start (Seconds (0.05));

// NODE 0 RETRIEVE THE DATA FILE
dce.ResetArguments ();
dce.ResetEnvironment ();
dce.SetBinary ("ccncatchunks2");
dce.SetStdinFile ("");
dce.AddArgument ("ccnx:/DATAFILE");
dce.AddEnvironment ("HOME", "/root");
getter = dce.Install (nodes.Get (0));
getter.Start (Seconds (1.0));
```



Topology (a)

Step by step example
- How we do the required steps (CCNx set start)

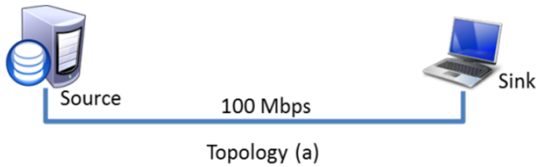


Topology (a)

```
// Set CCNx finish time for client and server
dce.ResetArguments ();
dce.ResetEnvironment ();
dce.SetBinary ("ccndsmoketest");
dce.SetStdinFile ("");
dce.AddArgument ("kill");
apps = dce.Install (nodes.Get (0));
apps.Start (Seconds (59.0));
apps = dce.Install (nodes.Get (1));
apps.Start (Seconds (59.0));

// add pcap log on the connection
pointToPoint.EnablePcap (std::string ("dce-ccn-xchg-") + packetSize, devices.Get (0), false, false);
```

Step by step example
- How we do the required steps (start simulation)

```
// Initiate and finish ns-3 simulation
 Simulator::Stop (Seconds (60.0));
 Simulator::Run ();
 Simulator::Destroy ();
}
```



Topology (a)

* Again, the presented example can be found at http://www-sop.inria.fr/members/Frederic.Urbani/in-progress/howto-ccn-exp1.html and the example code can be found in the root of DCE at dce/myscripts/ccn-exp1

Thank you!

Direct Code Execution Tutorial
Walkthrough

PLANETE Group
Inria Sophia Antipolis
Méditerranée

13 March 2013