



UNICAMP



Mini-CCNx: uma Plataforma de Prototipagem Rápida para Redes Orientadas a Conteúdo

Carlos Manuel Silvestre Cabral – RA: 059595

Orientador: Christian Esteve Rothenberg

Co-orientador: Maurício Ferreira Magalhães

Agenda

1. Introdução
 1. Contextualização/Motivação
 2. Objetivos/Escopo
2. Revisão Bibliográfica
 1. Características e Propostas do Modelo CCN
 2. Plataformas Experimentais para o Modelo CCN
3. Projeto e Implementação
 1. Abordagem
 2. Visão Geral
 3. Implementação
4. Metodologia e Resultados
5. Conclusão

*Mini-CCNx: uma Plataforma de Prototipagem
Rápida para Redes Orientadas a Conteúdo*
Carlos Cabral – RA:059595

INTRODUÇÃO

Contextualização/Motivação

- Redes Orientadas a Conteúdo (ROCs)
 - Conteúdo como elemento central das redes
 - Eficiência na distribuição de conteúdos – *caching*
 - Conteúdo nomeado
 - Segurança diretamente aplicada aos conteúdos
 - Roteamento baseado nos nomes
 - DONA (2007), CCN (2009), PSIRP (2009), NetInf (2009)
- Pesquisa experimental – crucial para a adoção global
 - Tema novo, promissor e complexo
 - Plataformas experimentais para ROCs
 - Pequena quantidade, baixa maturidade e documentação falha



Lacuna

Objetivos

- Desenvolver uma nova ferramenta experimental focada nas ROCs
- Deve apresentar um conjunto de características para facilitar e agilizar o desenvolvimento e validação de novas propostas para as ROCs:
 1. Flexibilidade
 - Agilidade para criação de diversas topologias e cenários
 - Configuração de parâmetros de *link*
 2. Escalabilidade
 - Número suficientemente grande de nós

Objetivos/Escopo

3. Baixo Custo

- Poderá ser executada em *laptop/desktop* de uso comum

4. Realismo [Handigol et al 2012]

- (i) Funcional
- (ii) Temporal
- (iii) Tráfego

5. Facilidade de Uso

- GUIs, topologias pré-definidas, configuração ágil e simples

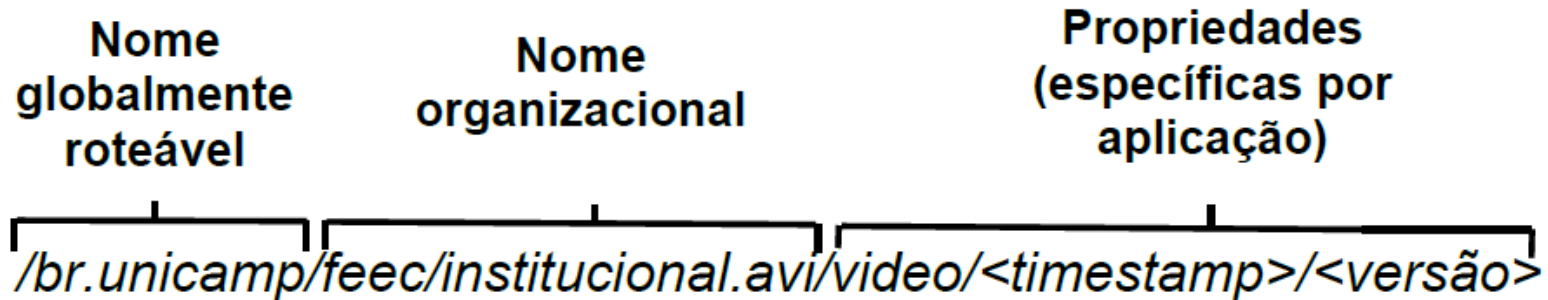
- Escopo

- Modelo de CCN, Linux, código aberto (GitHub)

*Mini-CCNx: uma Plataforma de Prototipagem
Rápida para Redes Orientadas a Conteúdo
Carlos Cabral – RA:059595*

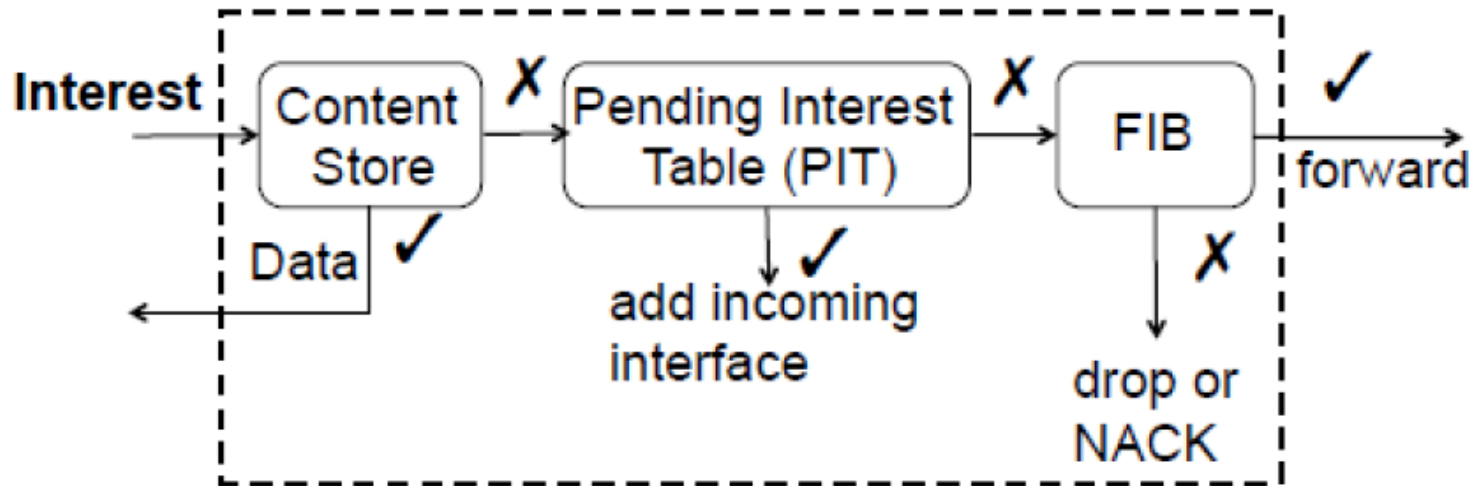
REVISÃO BIBLIOGRÁFICA

Modelo CCN - Nomeação



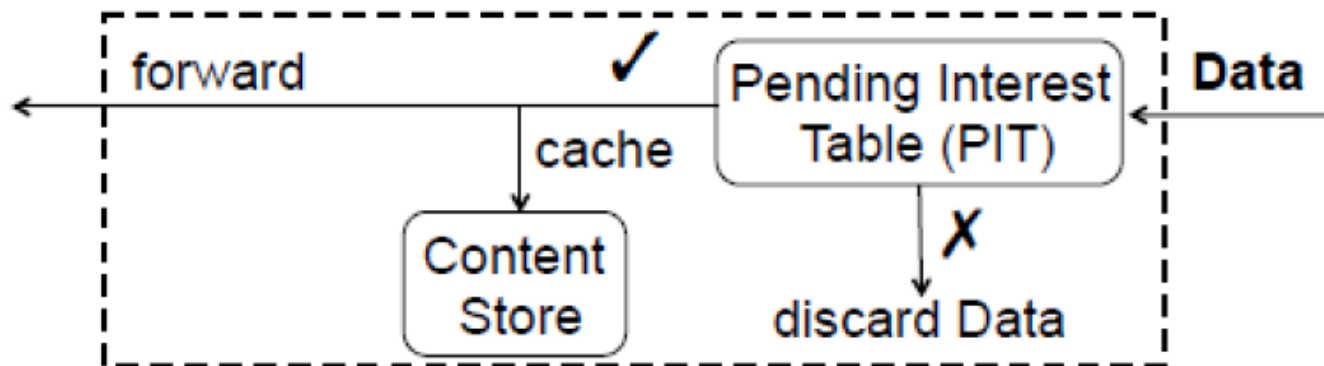
- Sequências de caracteres e números
- Nomes hierárquicos
- Característica semântica

Modelo CCN – *Caching* e encaminhamento



Downstream

Upstream



X lookup miss

✓ lookup hit

Modelo CCN – Implementação e Maturidade

- Projeto NDN (*Named Data Networking*)
 - Universidades e institutos americanos
 - Especificação, implementação, *testbed*, aplicações, etc.
- Implementação de referência: CCNx
 - Componente principal: *ccnd* (responsável por *caching* e encaminhamento)
 - Aplicação no espaço de usuário
 - Implementado majoritariamente em C (APIs C e JAVA)
 - Atualizações constantes, lista ativa

Modelo CCN – Desafios

- Escalabilidade
- Nomeação
- *Caching*
- Aplicações
- Outros
 - Rico ambiente de pesquisa e novas propostas

Plataformas Experimentais para o modelo CCN

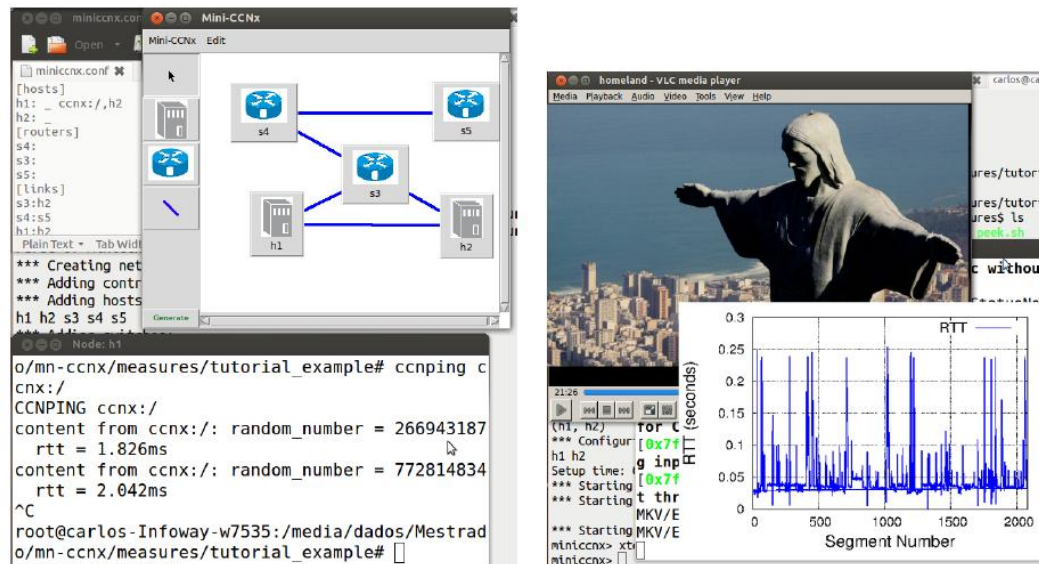
	Simuladores	Testbeds	Emuladores
<i>Ex:</i>	<i>ndnSIM/ccnSim</i>	<i>Testbed NDN</i>	-
Flexibilidade	Alta	Baixa	Alta
Escalabilidade	Alta	Baixa	Média
Custo	Baixa	Alta	Baixa
Realismo	Não	Sim	Sim
Facilidade de Config.	Média	Baixa	Alta
Config. de links	Sim	Com restrições	Sim

*Mini-CCNx: uma Plataforma de Prototipagem
Rápida para Redes Orientadas a Conteúdo
Carlos Cabral – RA:059595*

PROJETO E IMPLEMENTAÇÃO

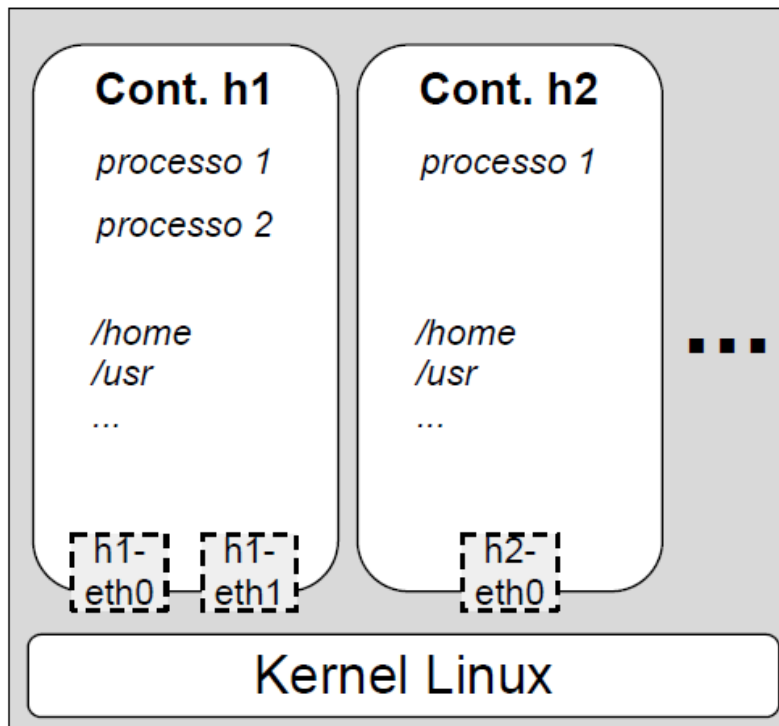
Abordagem

- Novo emulador focado no modelo CCN – **Mini-CCNx**
- Somente na forma de um emulador seria possível atingir o melhor balanço entre os requisitos propostos

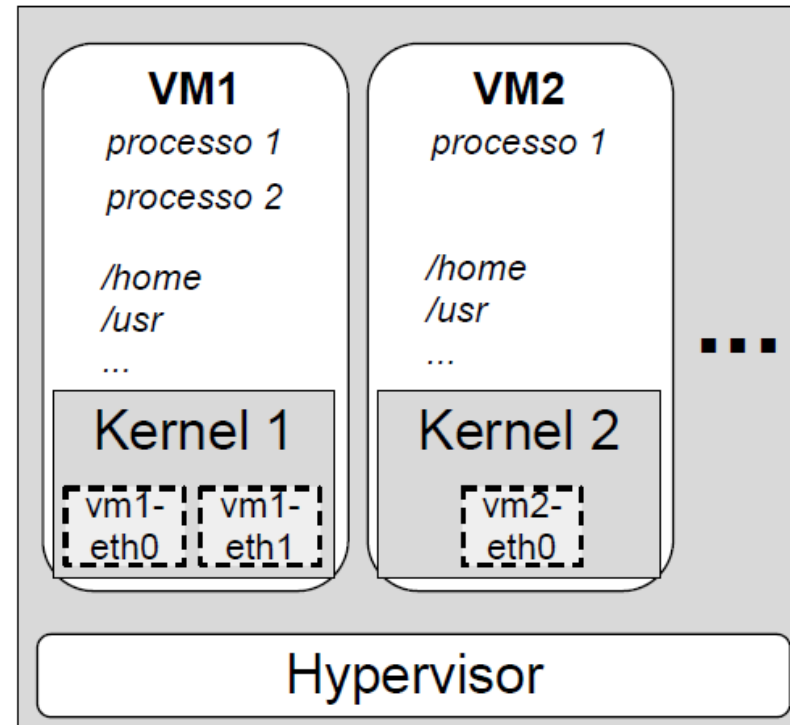


Abordagem

- Emulação Baseada em Contêineres (EBC)
 - “Virtualização leve”
 - Menor *overhead* se comparado com sistemas virtualizados
 - Único *kernel* -> questões sobre isolamento



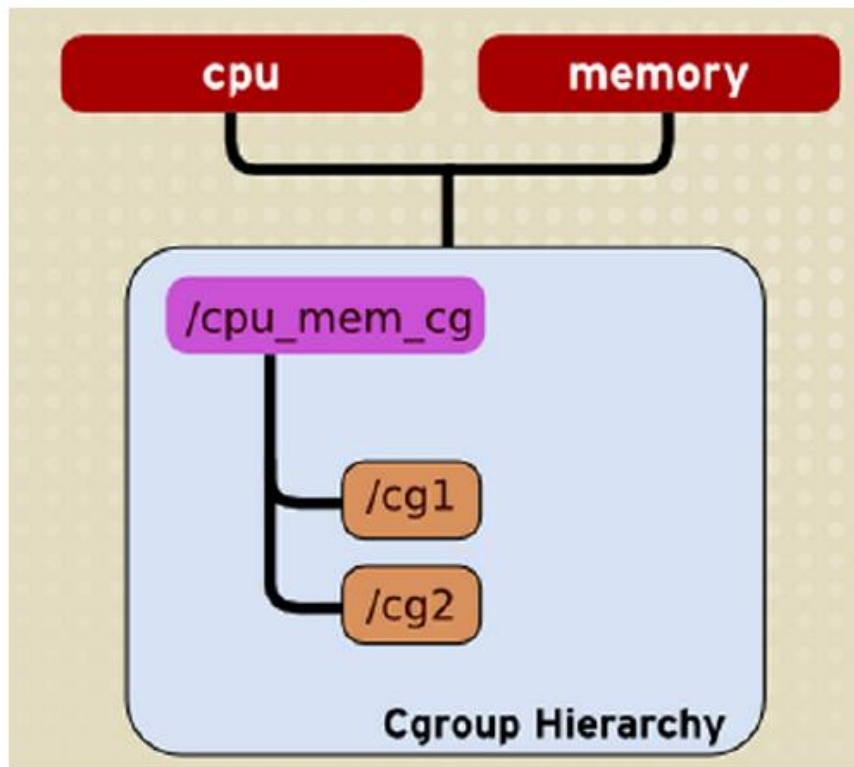
(a) Contêineres Linux



(b) Virtualização baseada em *hypervisor*

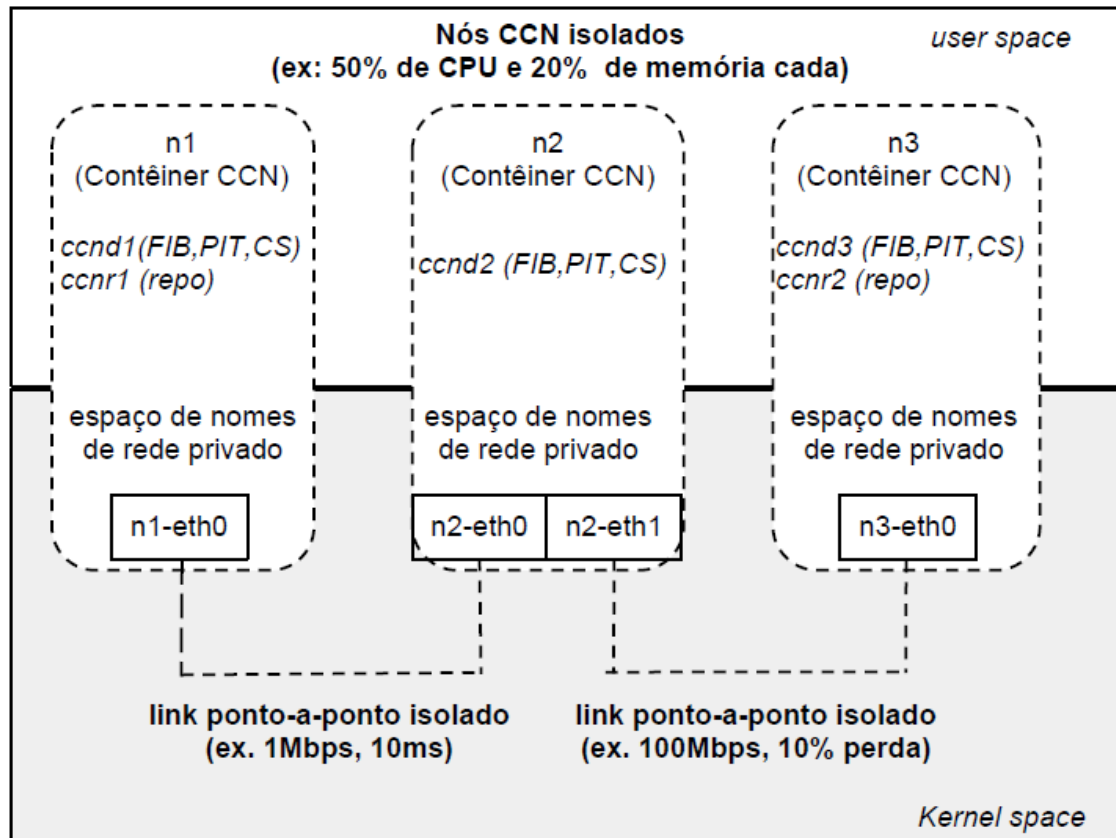
Abordagem

- Linux *control groups* (*cgroups*)
 - Permite alocar limites para a utilização de recursos (CPU, memória, etc) para um certo grupo de processos
 - Essencial para o isolamento de desempenho



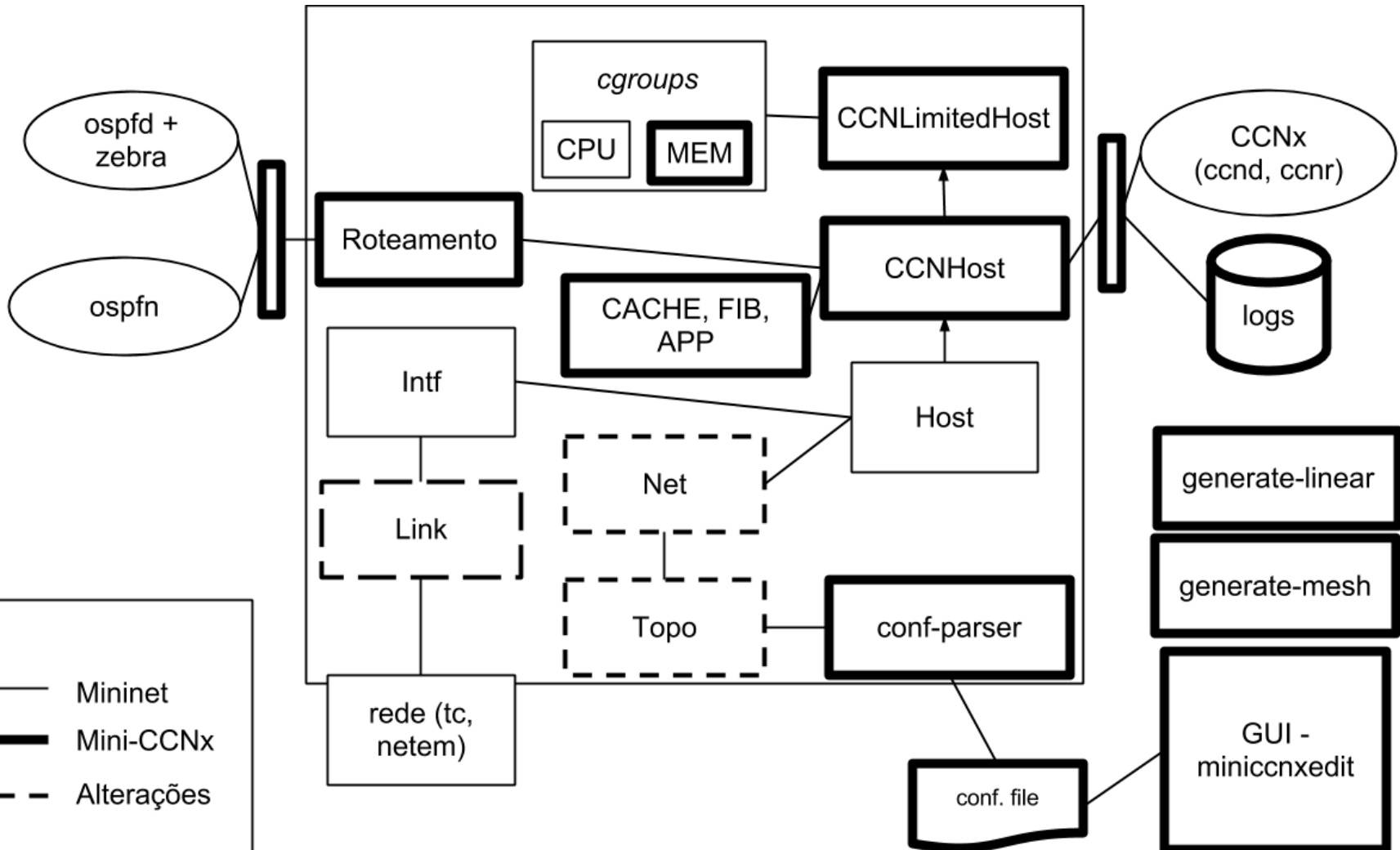
Visão Geral

- Cada contêiner é um nó CCN com seu espaço de nomes de rede privado



Implementação

- Fork do emulador Mininet (Redes OpenFlow)



Implementação

- Documentação, código, tutoriais e VM no GitHub

Tutorial

[New Page](#)[Edit Page](#)[Page History](#)

[Home](#) | [Installing](#) | [Tutorial](#) | [GUI](#) | [Node Classes](#) | [Topology Generators](#) | [Publications](#) | [Related Projects](#)

This will be a step-by-step guide of how to use **Mini-CCNx**. This tutorial assumes you already have **Mini-CCNx** installed. If that's not the case, please see [Installing](#). **Mini-CCNx** is a fork of *Mininet*. It would be nice to get familiar with it first. See the [Mininet Walkthrough](#) for more information on *Mininet*.

Mini-CCNx Configuration File

The main executable, **miniccnx**, needs one argument, that is the **Mini-CCNx** configuration file. In this configuration file, you'll specify a lot of things, such as the hosts, routers, the links between them, the apps that will be executed on the hosts, FIB entries and CPU limit for each host/router. This configuration file is a simple text file, so it can be generated with your favorite editor or by your scripts.

Every **Mini-CCNx** configuration file has three sections, namely, *[hosts]*, *[routers]* and *[links]*. Let's suppose we want to build a topology with 2 hosts (**h1** and **h2**), and a central CCN router (**s1**) between them. That would be something like this:

```
h1 ----- s1 ----- h2
```

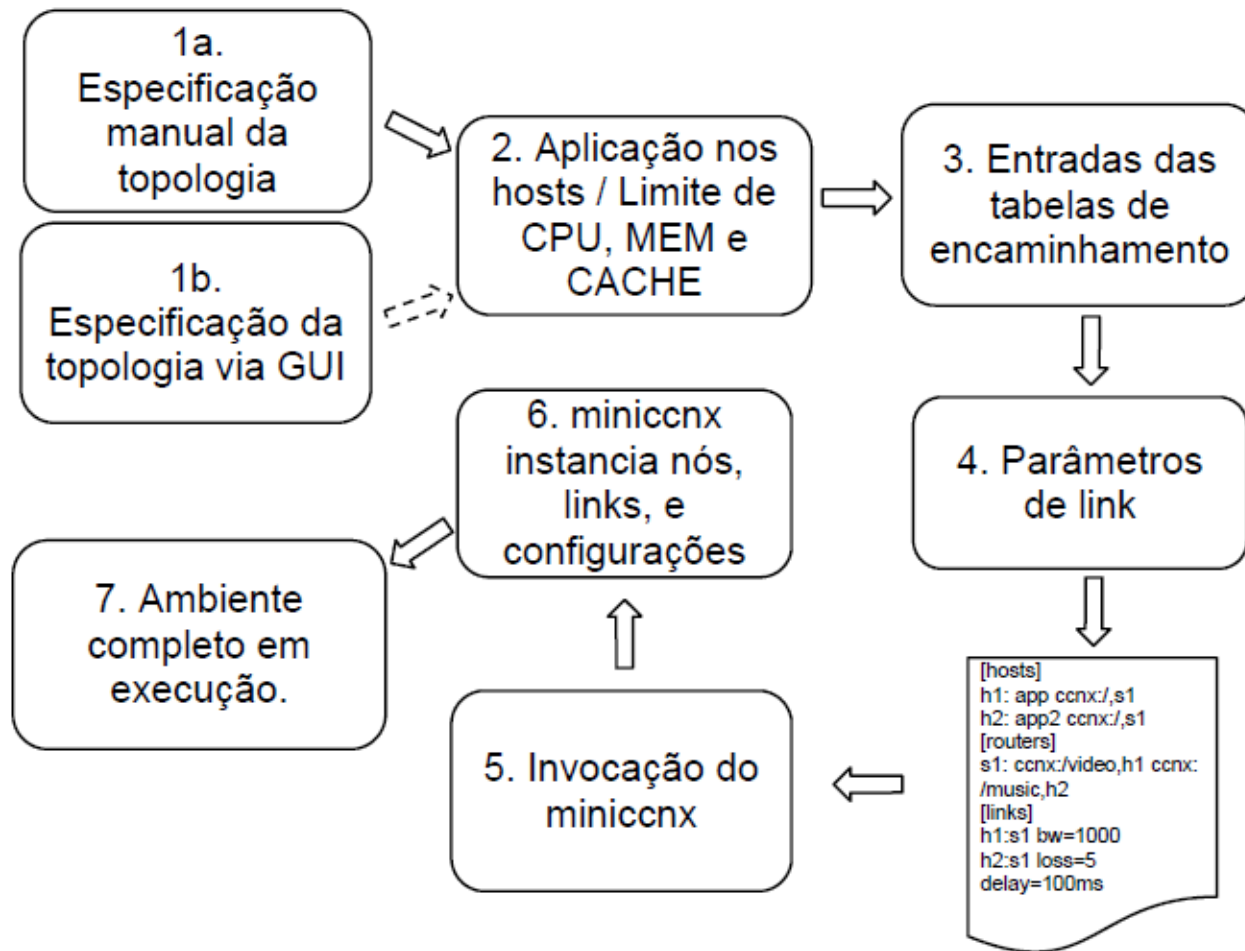
Below there's an example of a **Mini-CCNx** configuration file that represents such topology, plus another extra configuration parameters.

```
[hosts]
```

```
h1: app1 ccnx://,s1
```

```
h2: _ cpu=0.1
```

Implementação



*Mini-CCNx: uma Plataforma de Prototipagem
Rápida para Redes Orientadas a Conteúdo
Carlos Cabral – RA:059595*

METODOLOGIA E RESULTADOS

Metodologia

- Diversos experimentos para avaliar:
 - (i) Escalabilidade
 - (ii) Coerência
 - (iii) Capacidade de isolamento
 - (iv) Fidelidade ante experimentos reais
- Devem abranger a maior quantidade possível de aspectos das ROCs
 - *Caching*, roteamento, distribuição de conteúdos, encaminhamento adaptativo, aplicações...
- Recursos
 - *Latpop* mediano (Core i5 + 4GB RAM)
 - *Desktops* LCA
 - *ccnx* versão 0.7.0

Metodologia

- Medidas
 - *bwm-ng* (banda), *ping*, *tcpdump*
 - *ndndump*, *ccnping*, *ccngetfile*, *ccnputfile*, *ccndstatus*
 - Inspeção e *parse* de *logs* gerados durante os experimentos
 - 95% de confiança onde aplicável + GnuPlot
- Reprodução de Resultados da Literatura
 - Realismo e fidelidade do Mini-CCNx
- Testes abertos

Resultados

- Escalabilidade

Tabela 4.2: Escalabilidade. Topologia *full mesh*

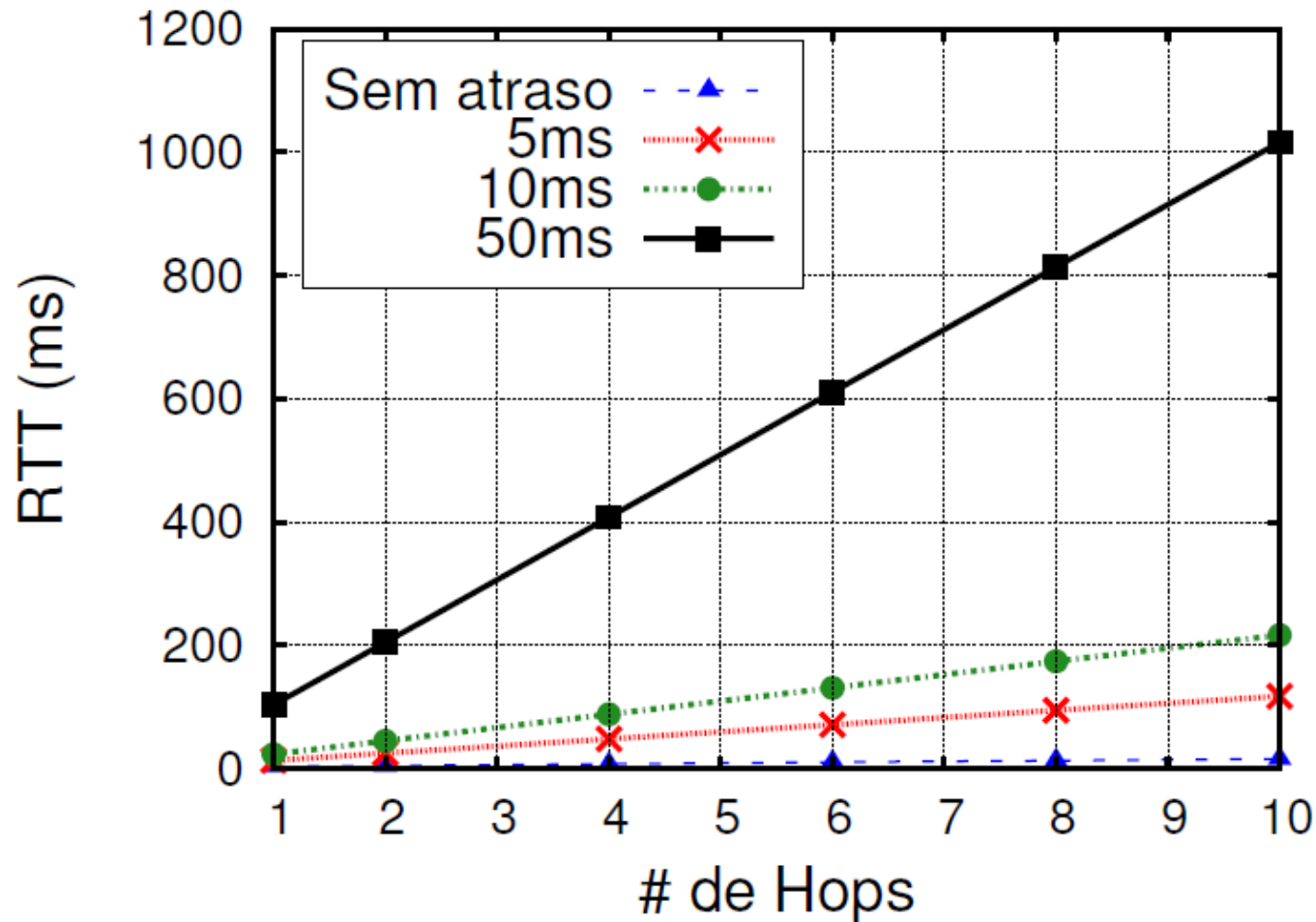
Nº de Nós	Mem. Mini-CCNx (MB)	Mem. <i>ccnd</i> (MB)	Mem. Total (MB)	Nº de <i>Links</i>	Tempo de inic. (s)
4	15.1	7.2	22.3	6	<1
8	15.3	13.7	29.0	28	3
16	15.7	28.9	44.6	120	11
32	18.2	57.1	75.3	496	48
64	26.0	114.5	140.6	2016	118
128	62.0	229.8	291.8	8128	753

Tabela 4.3: Escalabilidade. Topologia linear

Nº de Nós	Mem. Mini-CCNx (MB)	Mem. <i>ccnd</i> (MB)	Mem. Total (MB)	Nº de <i>Links</i>	Tempo de inic. (s)
4	15.0	7.2	22.2	3	<1
16	15.1	28.9	44.0	15	1
64	15.7	113.8	129.5	63	6
256	18.1	458.6	476.7	255	36
512	21.6	918.5	940.1	511	95
1024	27.8	1835.4	1863.2	1023	228
1536	35.3	2754.2	2789.6	1535	320

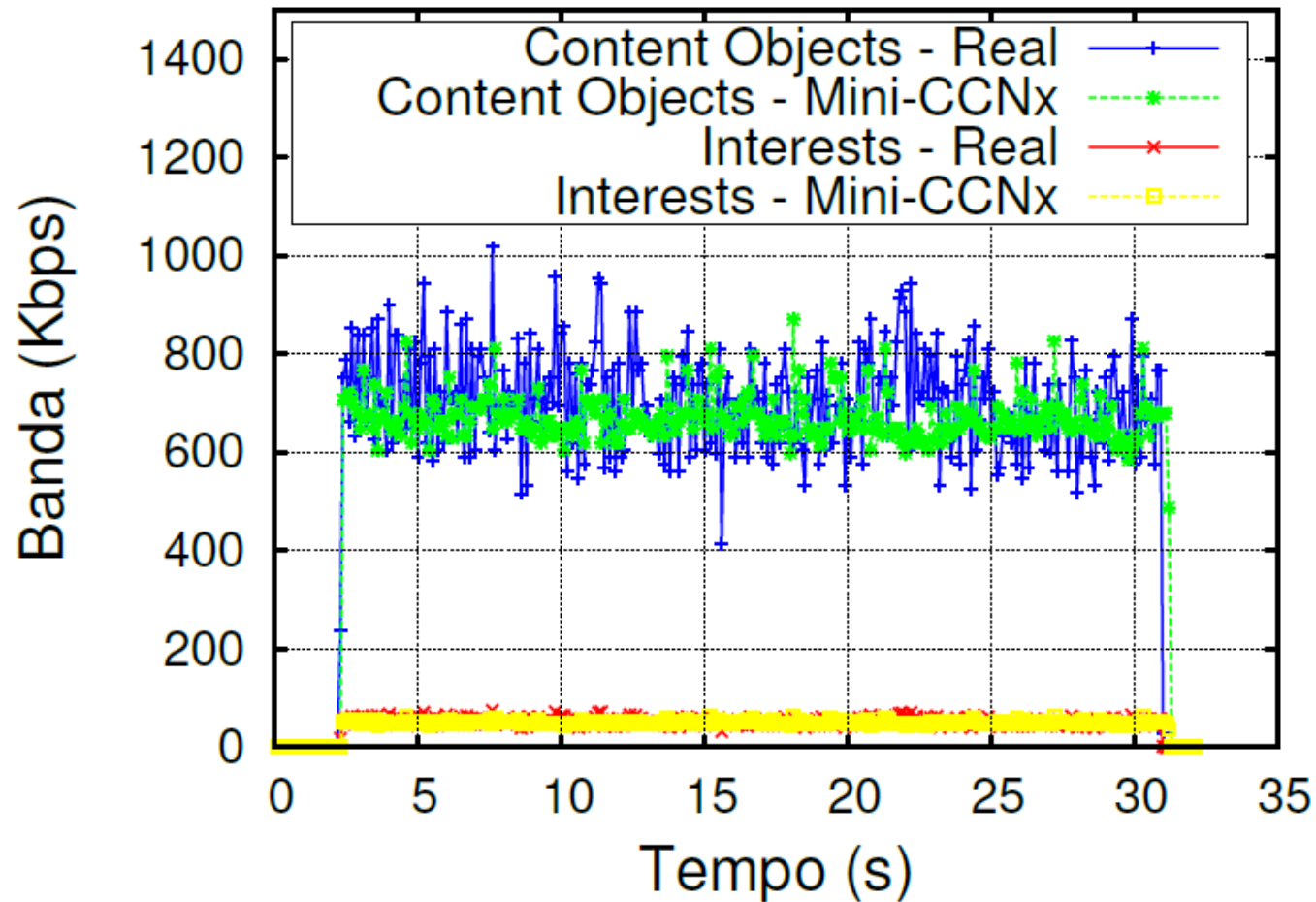
Resultados

- Coerência



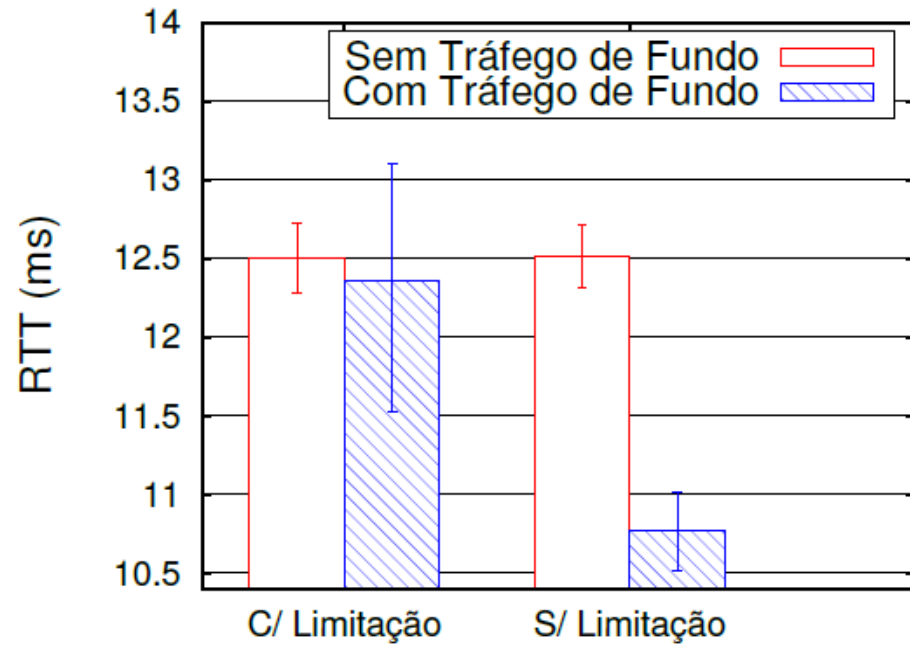
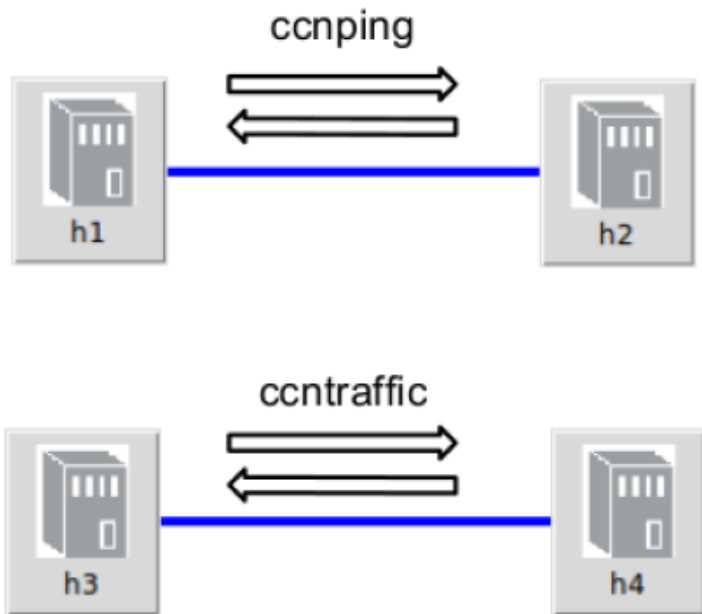
Resultados

- Fidelidade



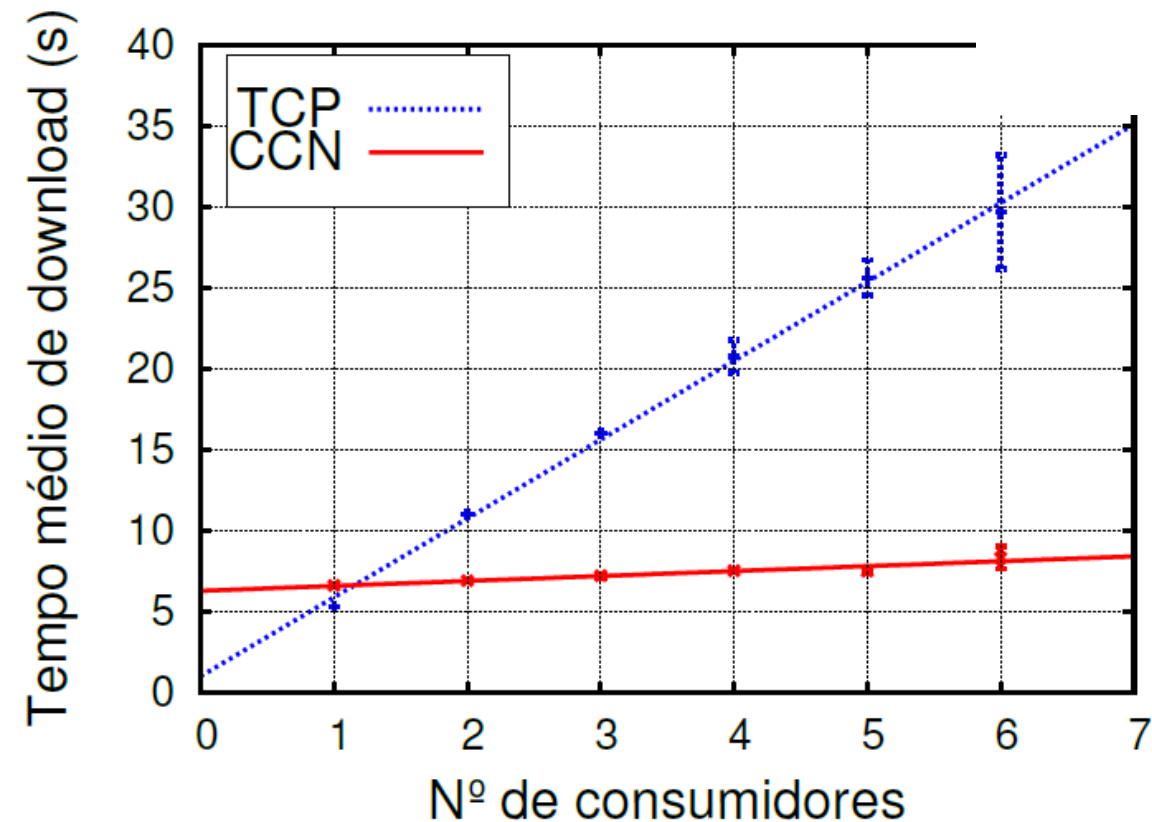
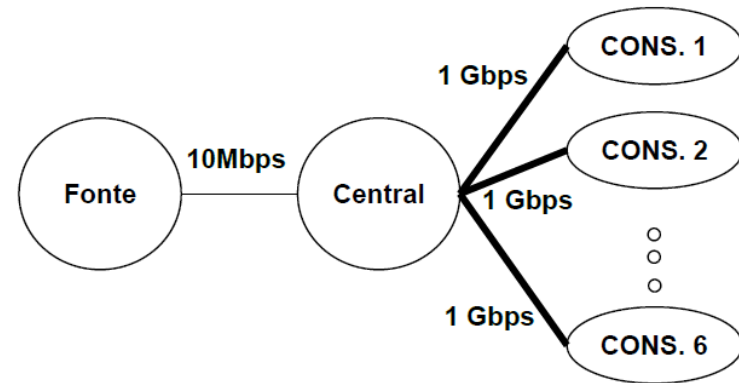
Resultados

- Isolamento



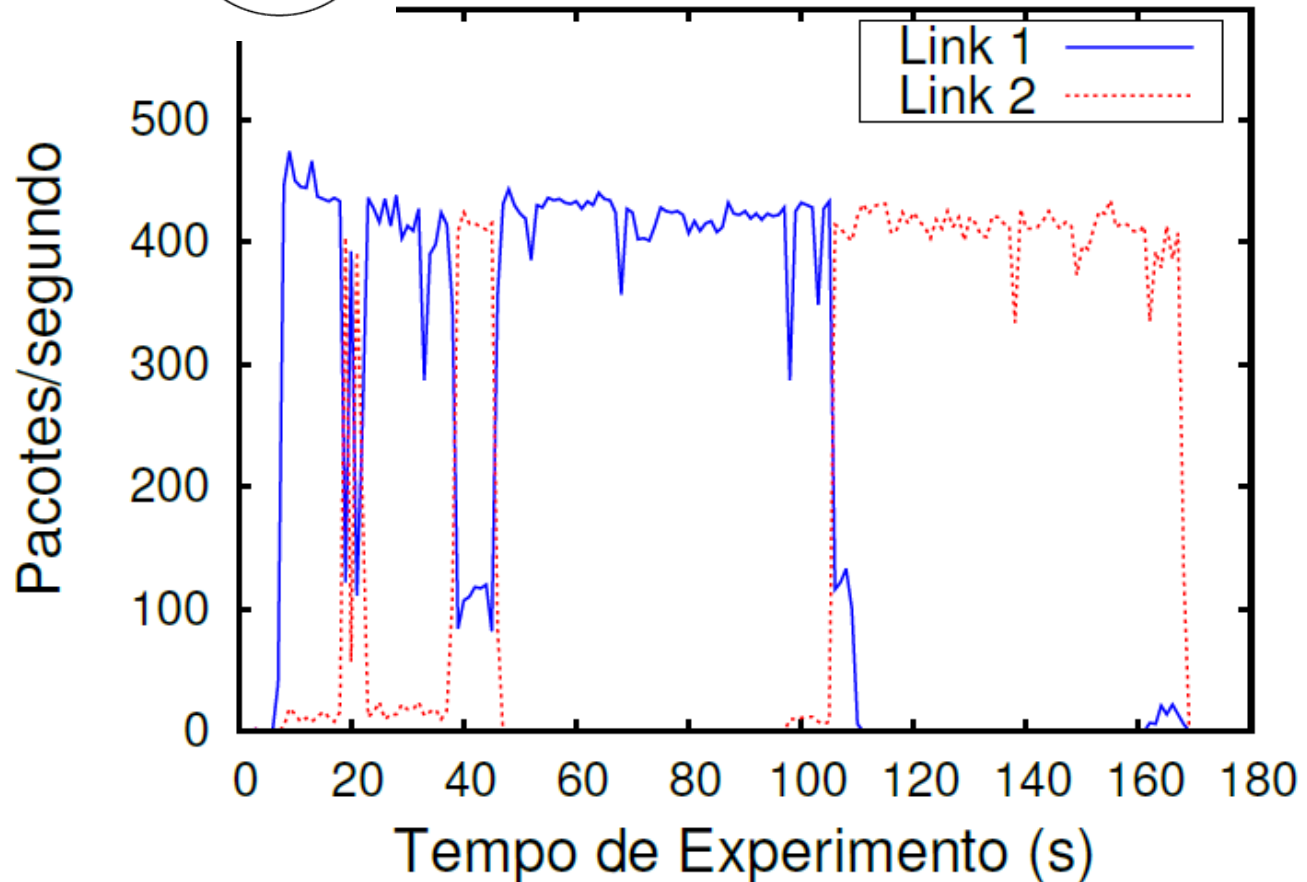
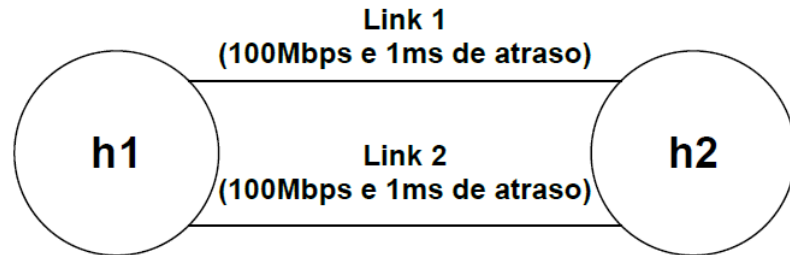
Resultados

- Reprodução de resultados da literatura [Jacobson, 2009]



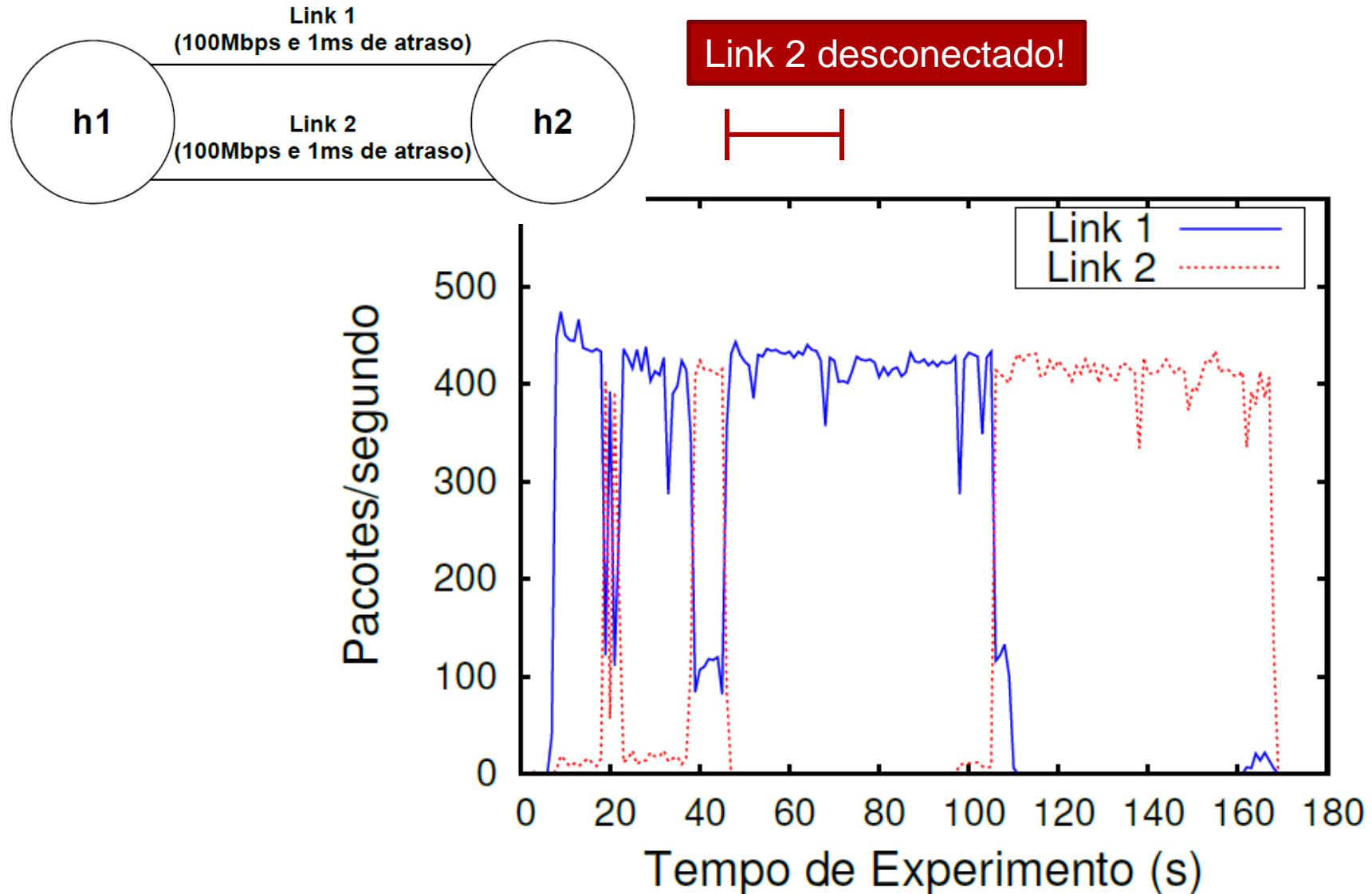
Resultados

- Reprodução de resultados da literatura [Jacobson, 2009]



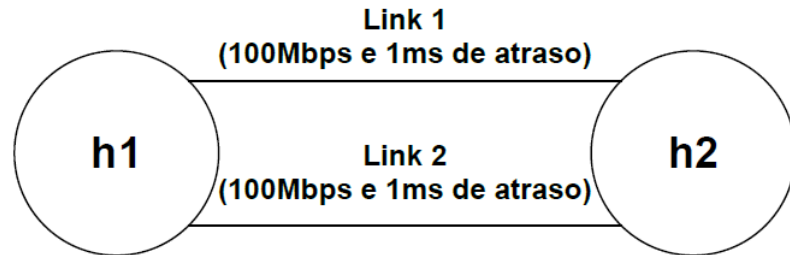
Resultados

- Reprodução de resultados da literatura [Jacobson, 2009]

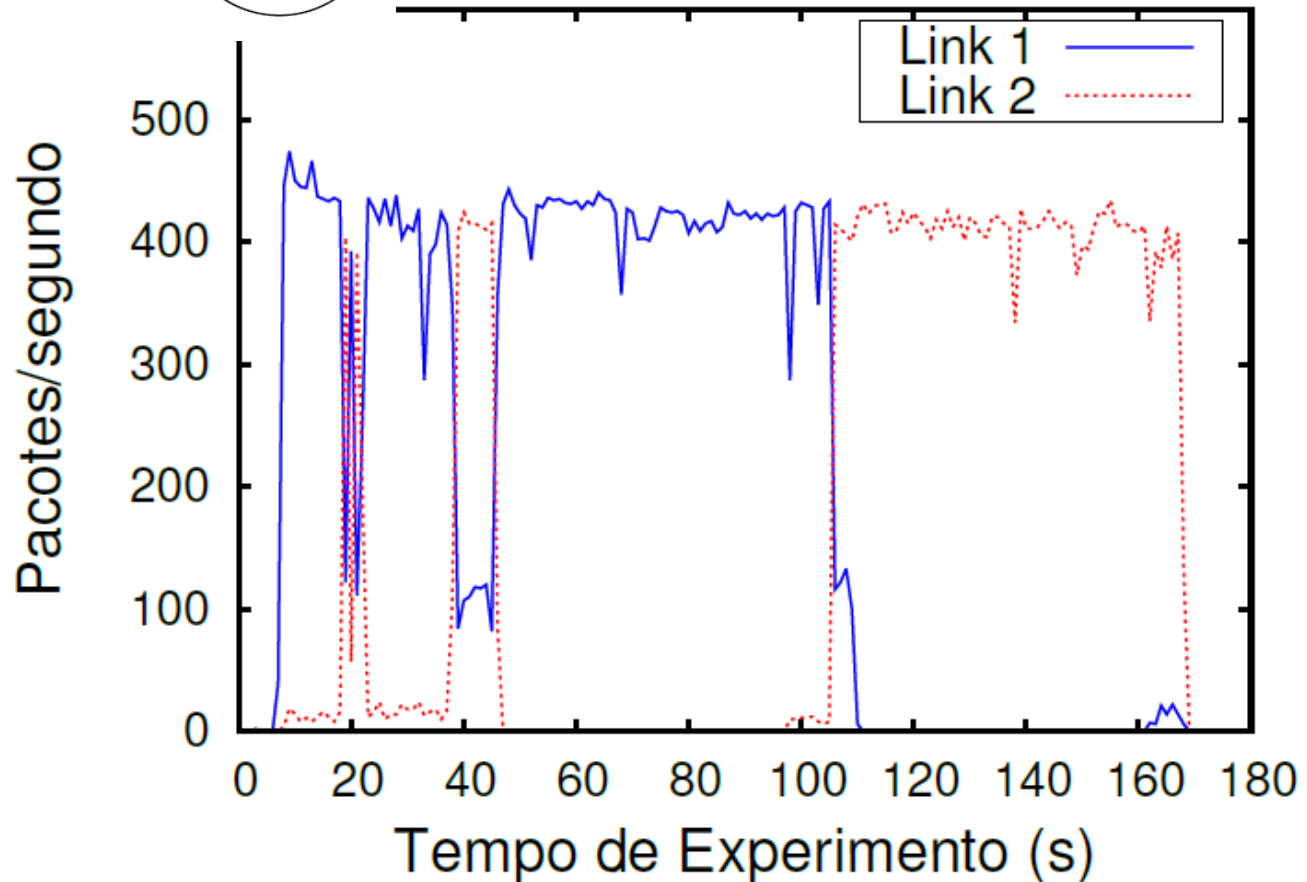


Resultados

- Reprodução de resultados da literatura [Jacobson, 2009]

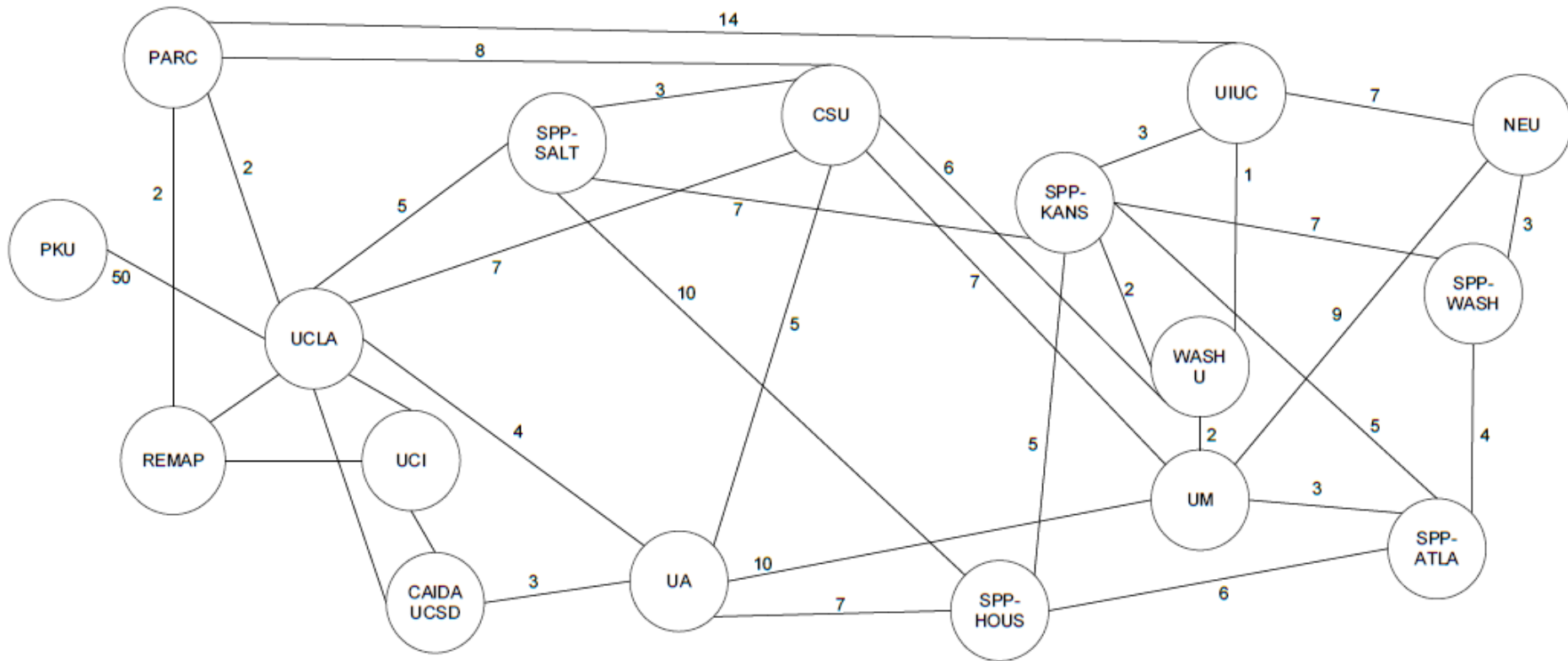


Link 1 desconectado!



Resultados

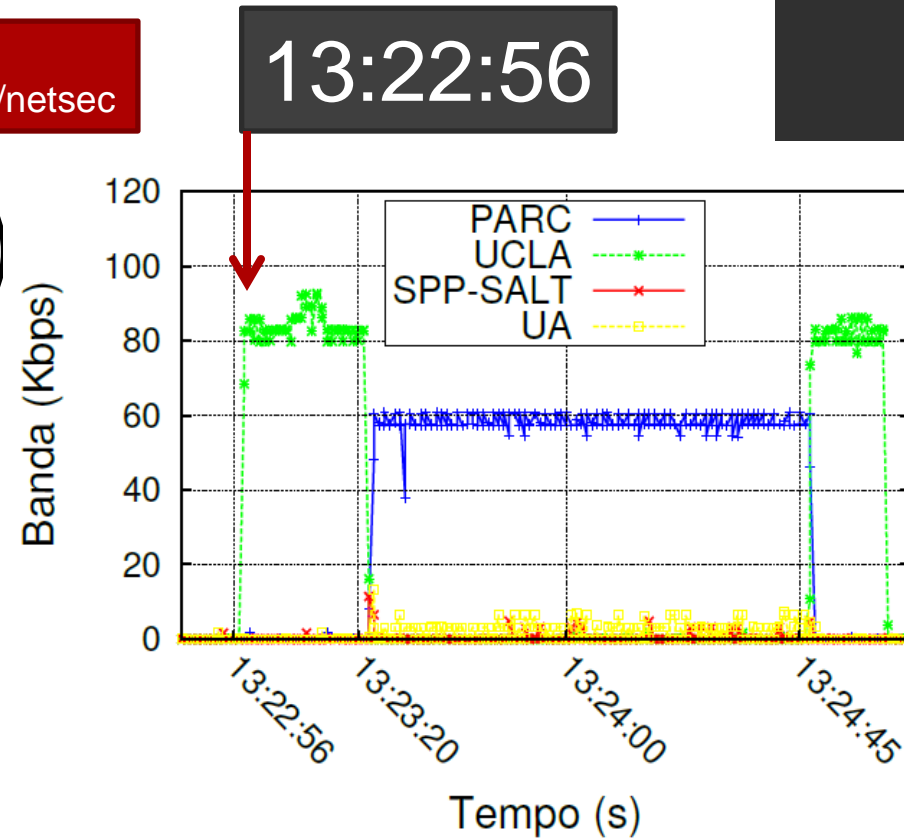
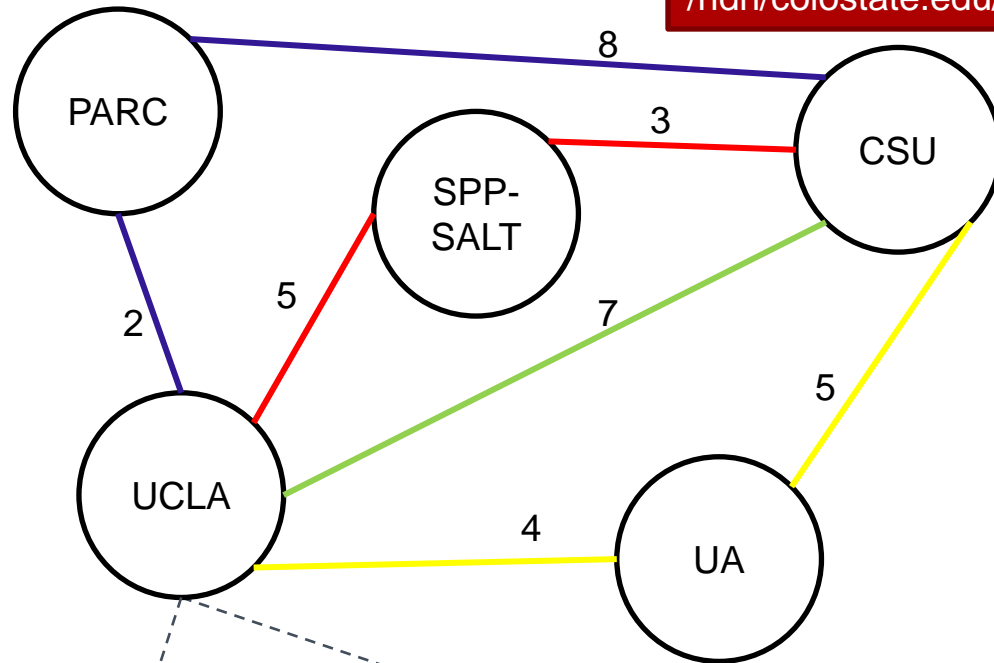
- Emulação do *testbed* NDN



Resultados

- Convergência de roteamento OSPFN

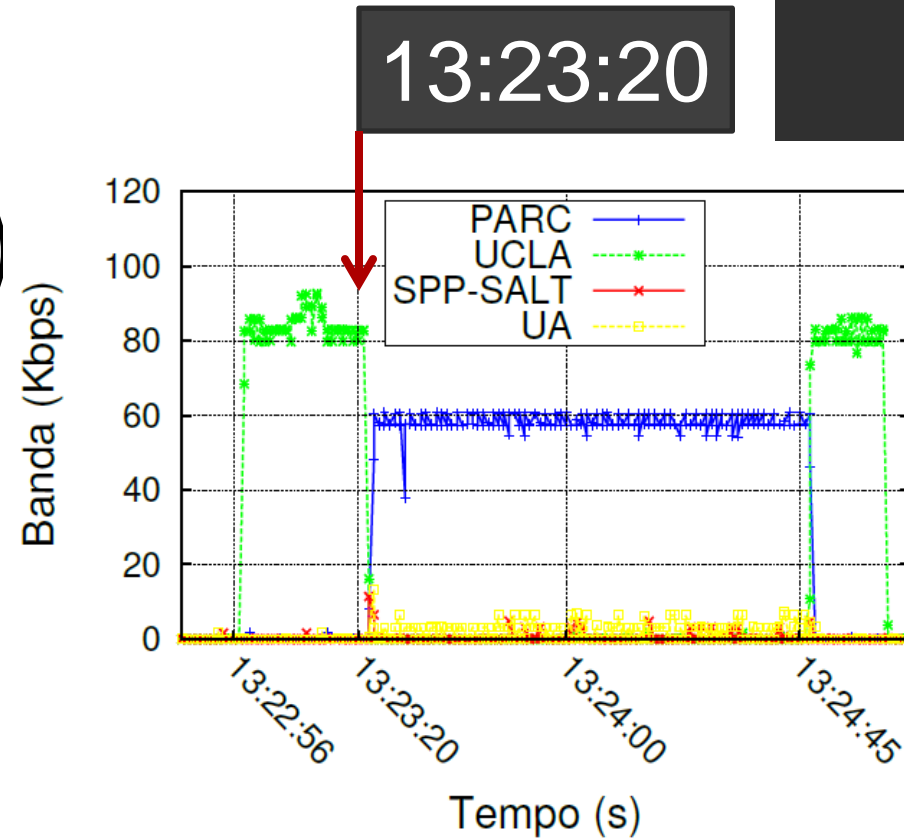
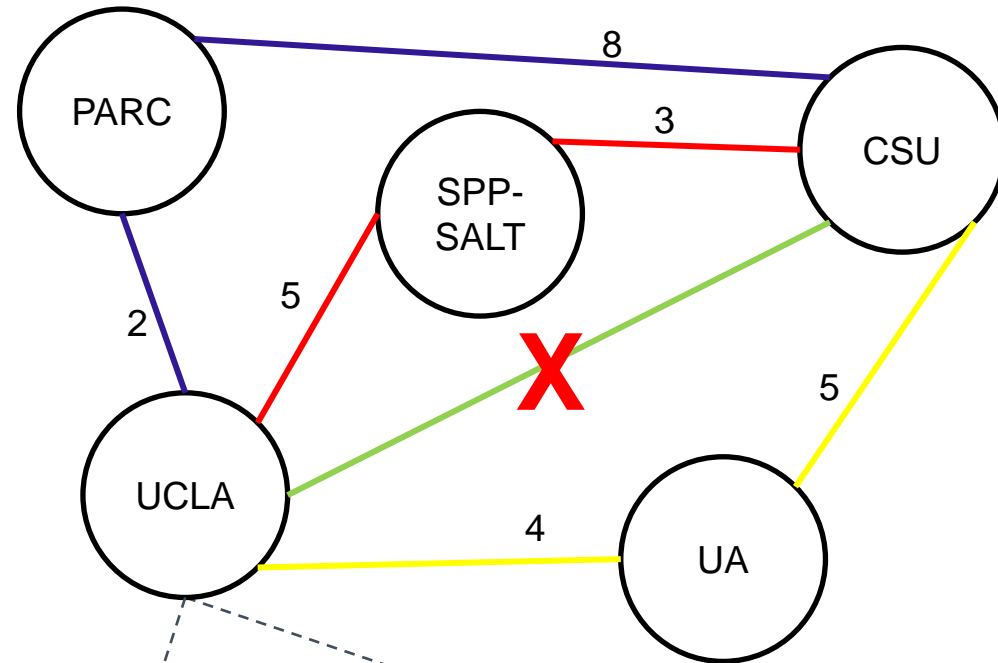
Divulga
[/ndn/colostate.edu/netsec](http://ndn/colostate.edu/netsec)



...
ccnx:/ndn/colostate.edu/netsec face: 8 flags: 0x3 expires: 214748301 (*face 8 = CSU*)
ccnx:/ndn/pku.edu.cn face: 10 flags: 0x3 expires: 2147483012 (*face 10 = PKU*)
ccnx:/ndn/neu.edu/northpole face: 13 flags: 0x3 expires: 2147483012 (*face 13 = UA*)

Resultados

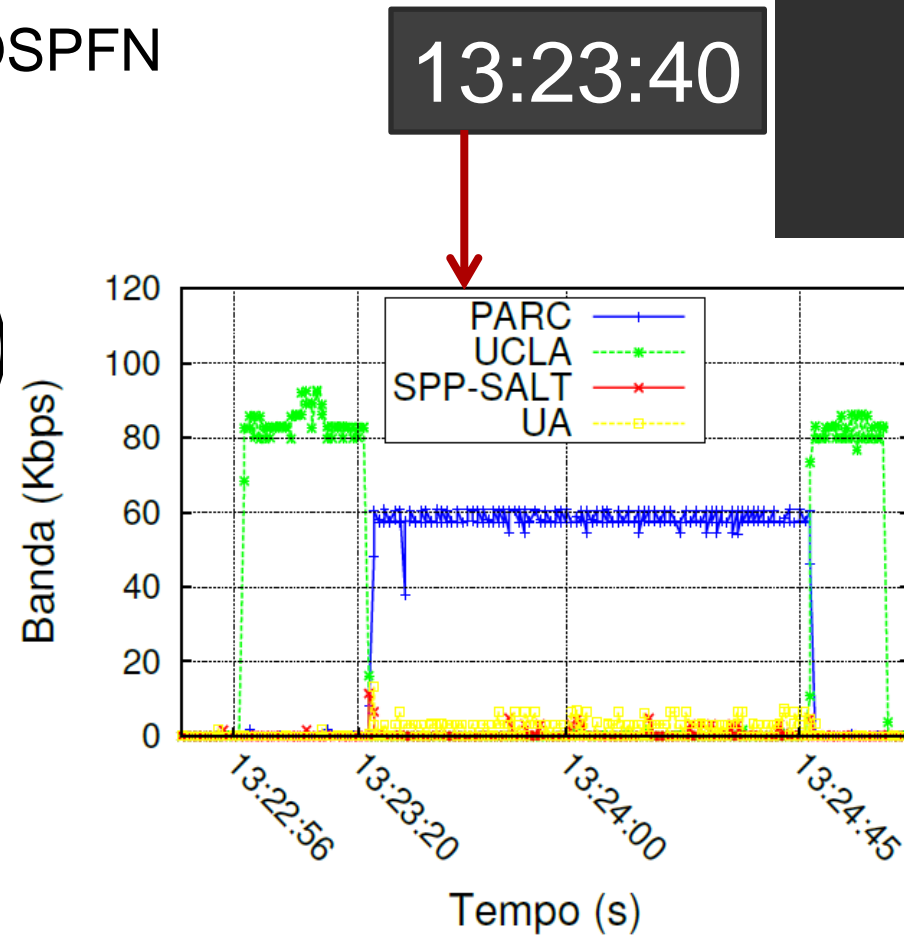
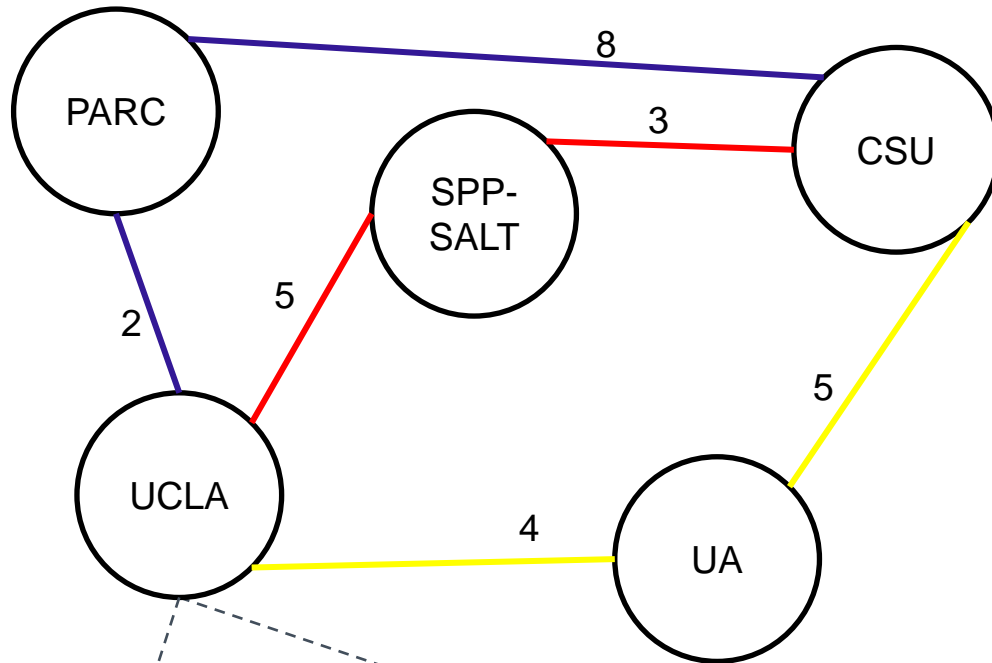
- Convergência de roteamento OSPFN



...
ccnx:/ndn/colostate.edu/netsec face: 13 flags: 0x3 expires: 2147483207 (*face 13 = UA*)
ccnx:/ndn/colostate.edu/netsec face: 12 flags: 0x3 expires: 2147483207 (*face 12 = SPP-SALT*)
ccnx:/ndn/colostate.edu/netsec face: 9 flags: 0x3 expires: 2147483207 (*face 9 = PARC*)

Resultados

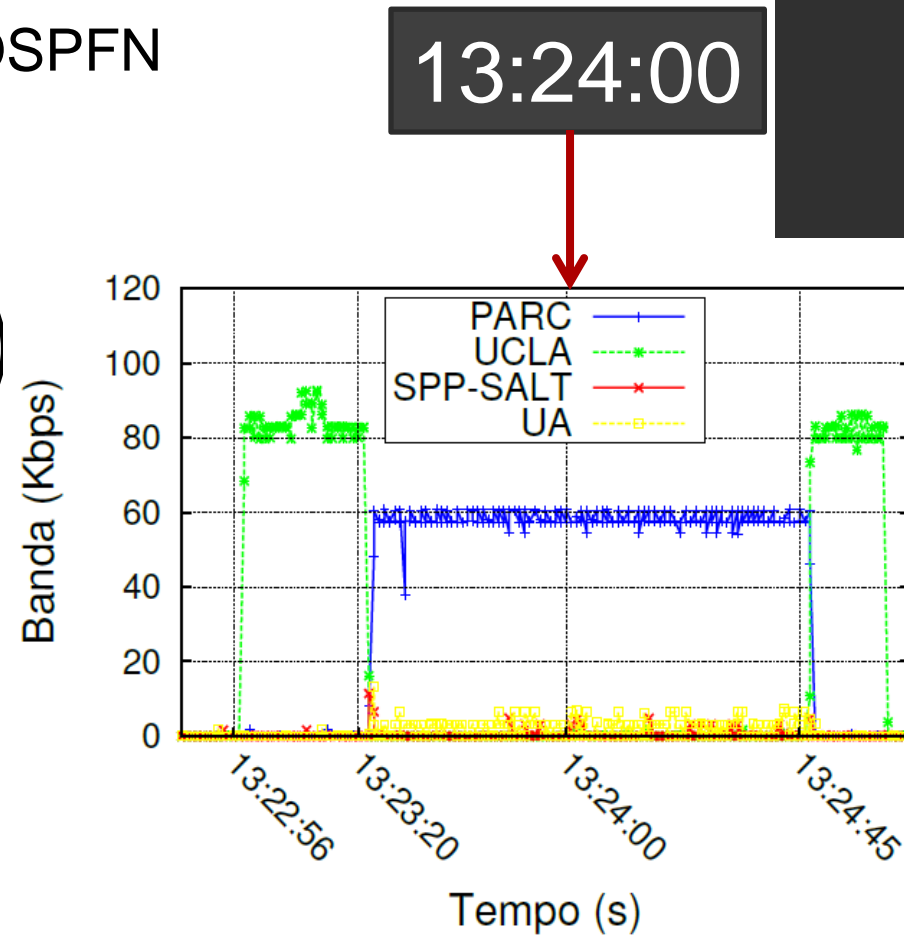
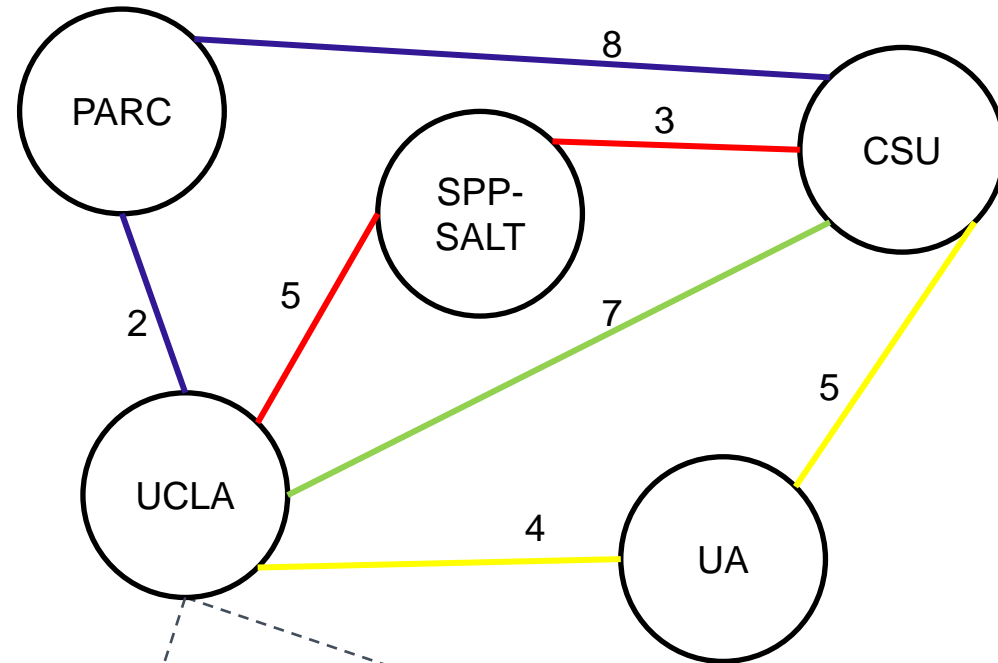
- Convergência de roteamento OSPFN



...
ccnx:/ndn/colostate.edu/netsec face: 13 flags: 0x3 expires: 2147483207 (*face 13 = UA*)
ccnx:/ndn/colostate.edu/netsec face: 12 flags: 0x3 expires: 2147483207 (*face 12 = SPP-SALT*)
ccnx:/ndn/colostate.edu/netsec face: 9 flags: 0x3 expires: 2147483207 (*face 9 = PARC*)

Resultados

- Convergência de roteamento OSPFN

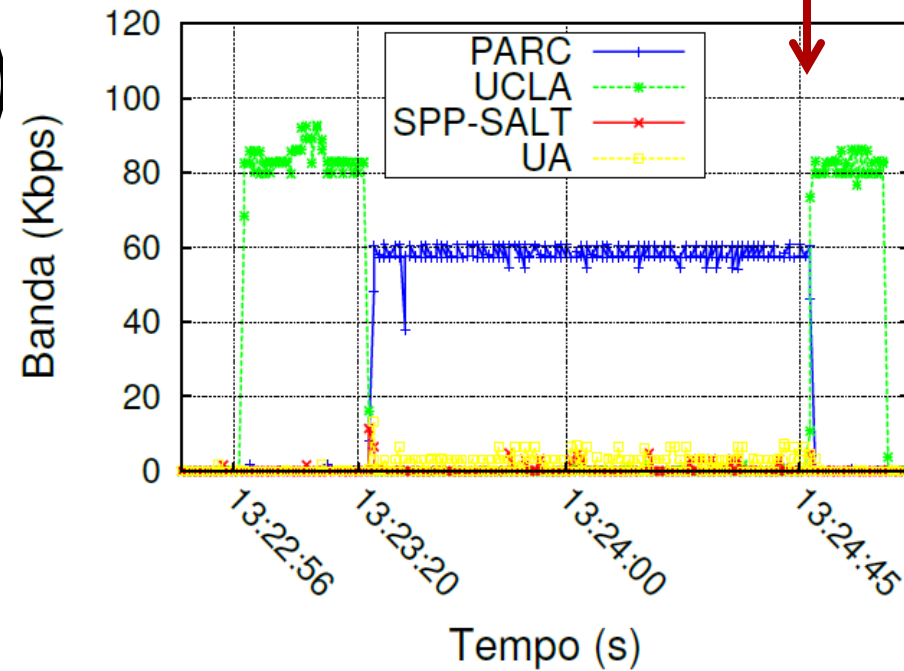
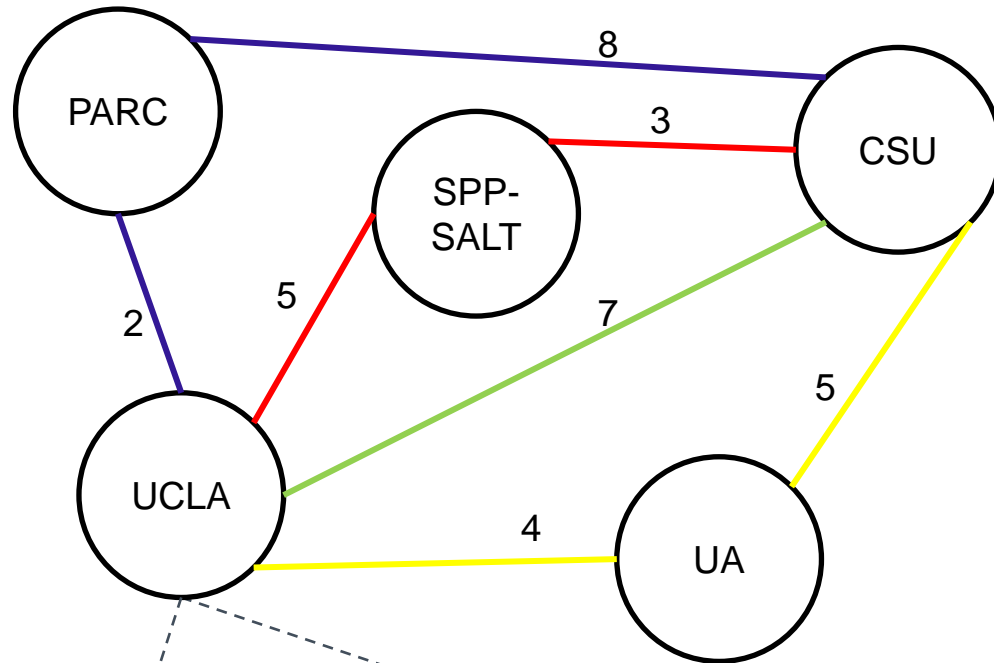


...
ccnx:/ndn/colostate.edu/netsec face: 13 flags: 0x3 expires: 2147483207 (*face 13 = UA*)
ccnx:/ndn/colostate.edu/netsec face: 12 flags: 0x3 expires: 2147483207 (*face 12 = SPP-SALT*)
ccnx:/ndn/colostate.edu/netsec face: 9 flags: 0x3 expires: 2147483207 (*face 9 = PARC*)

Resultados

- Convergência de roteamento OSPFN

13:24:45



...
ccnx:/ndn/colostate.edu/netsec face: 8 flags: 0x3 expires: 214748301 (*face 8 = CSU*)
ccnx:/ndn/pku.edu.cn face: 10 flags: 0x3 expires: 2147483012 (*face 10 = PKU*)

*Mini-CCNx: uma Plataforma de Prototipagem
Rápida para Redes Orientadas a Conteúdo
Carlos Cabral – RA:059595*

CONCLUSÃO

Conclusão

- O Mini-CCNx preenche uma lacuna existente entre as atuais plataformas experimentais para CCN
 - Primeiro emulador genérico para a área
- Código aberto + Documentação
 - Qualquer pesquisador da área pode utilizá-lo e melhorá-lo
- Utilização para ensino

Conclusão

- Trabalhos futuros
 - Redes de sensores sem-fio + CCN
 - Interface gráfica
 - Protocolos de Roteamento
 - Encaminhamento e *Caching*

Publicações

1. ACM SIGCOMM ICN Workshop(ICN 2013-Hong Kong)
 - “Mini-CCNx: Fast Prototyping for Named Data Networking”
 - Visão geral sobre a ferramenta e sua avaliação
2. ACM SIGCOMM ICN Workshop(ICN 2013-Hong Kong)
 - “Reproducing Real NDN Experiments using Mini-CCNx (DEMO SUBMISSION)”
 - Demonstração focada em reproduzir experimentos reais
3. SBRC 2013 – Salão de Ferramentas (Brasília, Maio)
 - “Mini-CCNx: Prototipagem Rápida para Redes Orientadas a Conteúdo baseadas em CCN”

Obrigado



UNICAMP