

# Apresentação do grupo UFSCar/USP/CPqD/UFES

Workshop REVIR

15/03/2012

# Grupo

- Cesar Augusto C. Marcondes (UFSCar)
- Cesar Augusto C. Teixeira (UFSCar)
- Maria da Graça Pimentel (USP)
- Magnos Martinello (UFES)
- Marcos Salvador / Christian E. Rothenberg (CPqD)

# Roteiro

- Proposta do sub-Projeto
- Estado Atual
  - Testbeds
    - OFELIA
    - RouteFlow
    - Omni / VNext
  - CastFlow
  - Transcodificador
- Integração dos Trabalhos

# Camada do Operador da Rede

- Consórcio: UFScar-USP-CPqD-UFES
- Principal atuação
  - Caso de Uso: uso de **aplicações** sensíveis a parâmetros de QoS sobre a **infraestrutura de redes virtuais**, como a difusão de **vídeo**
- Co-responsabilidade em outras tarefas em conjunto com outras universidades parceiras (**integração**)

# Definição do Caso de Uso

- Assumindo como premissa: uma Internet com *capacidade de virtualização (slices) e programabilidade*
- A Idéia do Caso de Uso
  - **difundir conteúdo multimídia**, ex. vídeos, com **diferentes resoluções em diferentes redes virtuais** com isolamento e QoS
- Imaginemos um cenário em que
  - usuário tipo **A** tenha **TV de alta definição e rede 1Gbps**
  - usuário tipo **B** tenha **notebook e rede 100Mbps**
  - usuário tipo **C** tenha **smartphone e rede 3G a 200 kbps**

**Tipo A, B, C = Grupo A, B, C**

# Soluções da Internet Hoje

- Botão YouTube HD – usuário escolhe a qualidade

## YouTube HD Is Here to Stay!

YouTube

**HD**  
ready

YouTube has just announced that users now have the option of watching some videos in **720p HD**. These streams are available on certain videos, depending on the source file uploaded to YouTube.

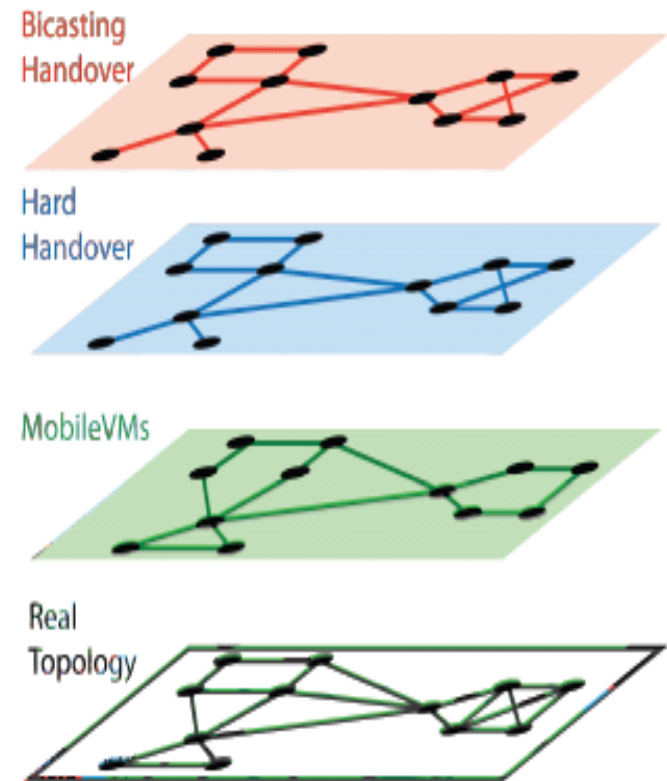
The ability to upload higher quality videos means improved sharpness and visual clarity for some YouTube videos. It's very noticeable in videos that contain text or fast movement and also in full-screen mode.

- **Youtube: Vídeos de diferentes formatos e codificações** são previamente re-codificados em “flash vídeo” em formato e resoluções fixas e armazenados
  - Processo é feito em background e posterior oferece links HD e normal
- **YouTube: Distribuição** usando uma **rede CDN**
  - Distribuição de carga entre os continentes, mas cada site ainda usa modelo Cliente/Servidor na última milha (N\*clientes)

# Solução com Virtualização da Rede

- **Operador da Rede (ex.: Youtube)** requisita **fatias de rede** para o provedor de redes virtuais, de modo a acomodar **árvores de disseminação de vídeo de diferentes qualidades (normal, HD)**
- Distribuir **eficientemente o vídeo** entre **os switches virtualizados** usando uma árvore multicast

*Exemplo de Slices controlados pelo FlowVisor/Openflow*

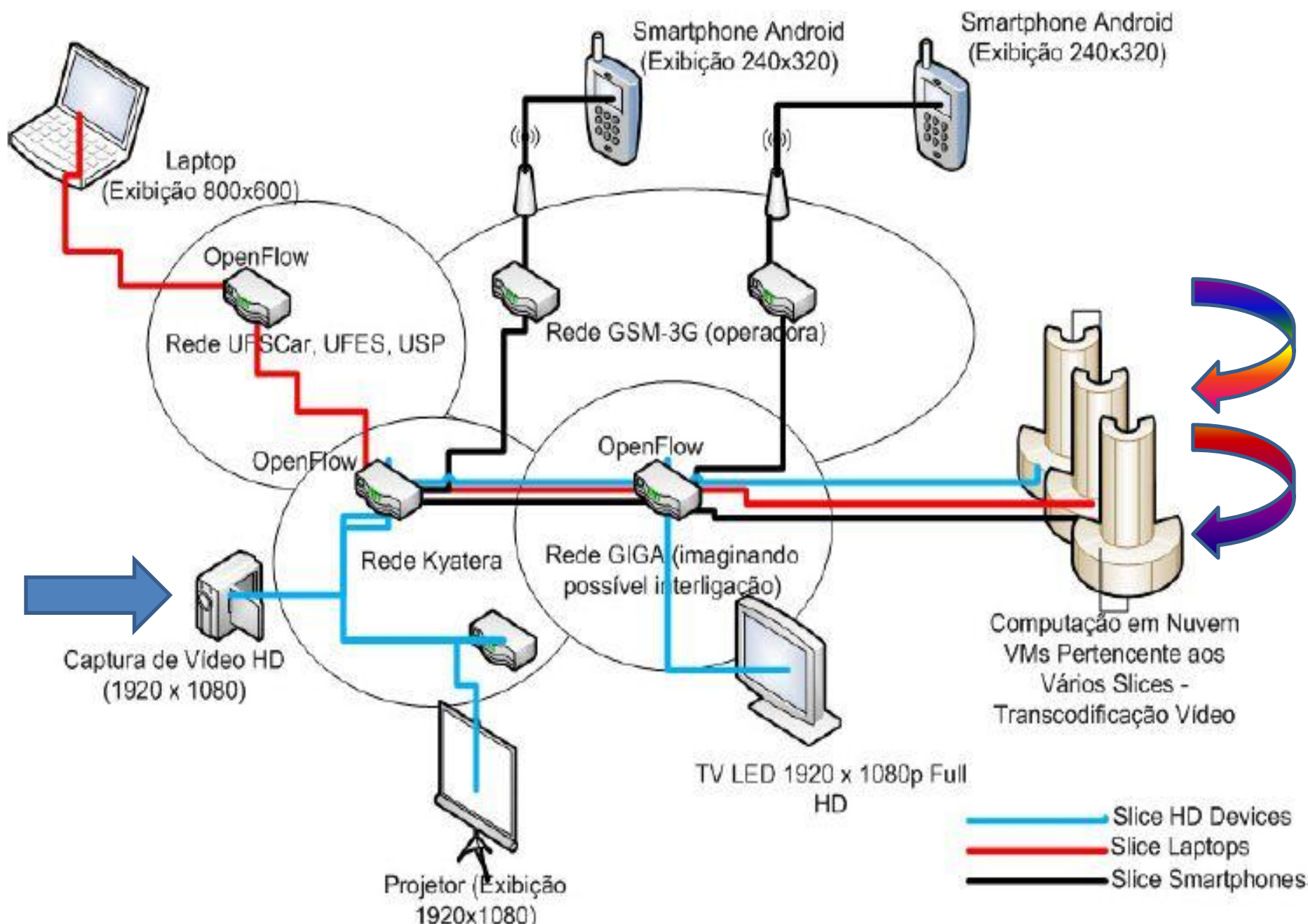


## Detalhamento do Caso de Uso

1. **Switches** com capacidades de virtualização **openflow, xen, híbrido** de acordo com desenvolvimento do projeto REVIR
2. **Isolamento dos slices, QoS, segurança**, a partir de desenvolvimentos do projeto REVIR
3. Diferentes **‘fatias de rede’** podem **interoperar** (como se fosse roteamento entre fatias) e nesse caso, usar transcodificação baseado na nuvem
4. Desenvolvimento de Software para **Criação da Árvore de Disseminação** (Multicast) onde cada **Árvore** esta associada a uma qualidade de **Vídeo**
5. **Testbed contempla Switches Programáveis** equipados com **placas netFPGA** (hardware reprogramável) fazendo aceleração da virtualização da rede a nível de hardware , interconectando as instituições do sub-grupo



# Exemplo da Testbed com Aplicação Proposta em Execução



- Estado Atual
  - **Testbeds**
    - **Omni / VNext**
    - **OFELIA**
    - **RouteFlow**
  - CastFlow
  - Transcodificador

# Testbeds

- Requisitos:
  - Cubos netFPGA
  - Servidor de Virtualização
- Idéia: Interconectar as instituições e experimentar ferramentas de controle e monitoramento de testbed (CMF)
- Direcionamento Inicial:
  - OMNI / VNNext (UFES)
  - OFELIA CF (UFSCar)
  - CPqD (openflow + flowvisor + routeflow)

# Testbed – Resultados (1/3)

- OMNI / VNNext
  - Sergio Charpinel (UFES – Pop-ES RNP)
  - Ricardo Guessuato (UFSCar)
- Instalação concluída: UFSCar
- Experimentos ainda em testes
- Tem VPN + as ferramentas em 1 máquina (não é netFPGA): UFES já operacional

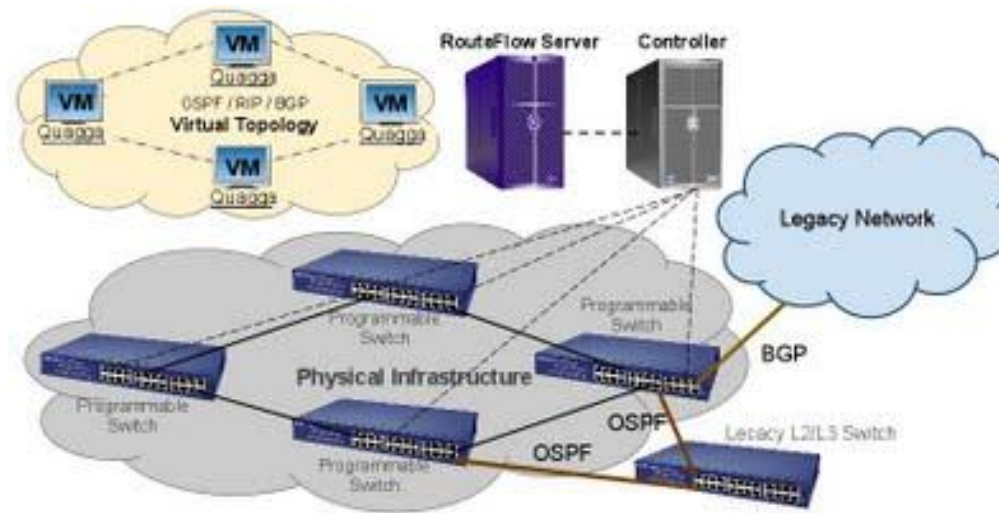
# Testbed – Resultados (2/3)

- OFELIA CF
  - Baseado em projeto FP7
  - Software OCF instalado, configurado e testado
  - Localmente e na testbed OFELIA
- Suporte:
  - Flowvisor para isolamento das fatias
  - VM baseada em Xen nas pontas como *hosts*
- Mini-curso SBRC 2012
- VM RouteFlow e VM Transcodificador

# Testbed – Resultados (3/3)

- RouteFlow
  - Interesse principal do CPqD
- Foram feitos testes com **RouteFlow + FlowVisor** (UFSCar / CPqD)
  - FlowVisor isola (FlowSpace)
  - RouteFlow – roteamento legado (Quagga)

- Figura Ilustrativa:



# Testbed

- Montada Testbed Local UFSCar (OFELIA)
- Alocados Servidores para OMNI/Vnext (UFES, UFSCar)
- Equipamentos netFPGA chegaram em Fevereiro (foram entregues em cada instituição parceira) – máquinas para testbed

- Estado Atual
  - Testbeds
    - Omni / VNext
    - OFELIA
    - RouteFlow
  - **CastFlow**
  - Transcodificador



# CastFlow

- Repensar **modelo Multicast** e prover distribuição eficiente de dados baseado em OpenFlow
- Estudo de Caso: IP TV
  - Cada **emissora seria tratada como fonte de dados**.
    - Emissora tem programação normal e faz chamada para uma Transmissão Ao Vivo direto depois volta para Emissora
  - **Receptores no mesmo canal** pertencem ao mesmo grupo multicast.
    - Endereço IP (privados ou **público com semantica redefinida**) alocado para Grupo
  - A **troca de canal** no receptor implicaria na **troca para outro grupo**.
  - **Troca constante de canal** das rotas nos grupos.
    - Abaixo de 1 seg

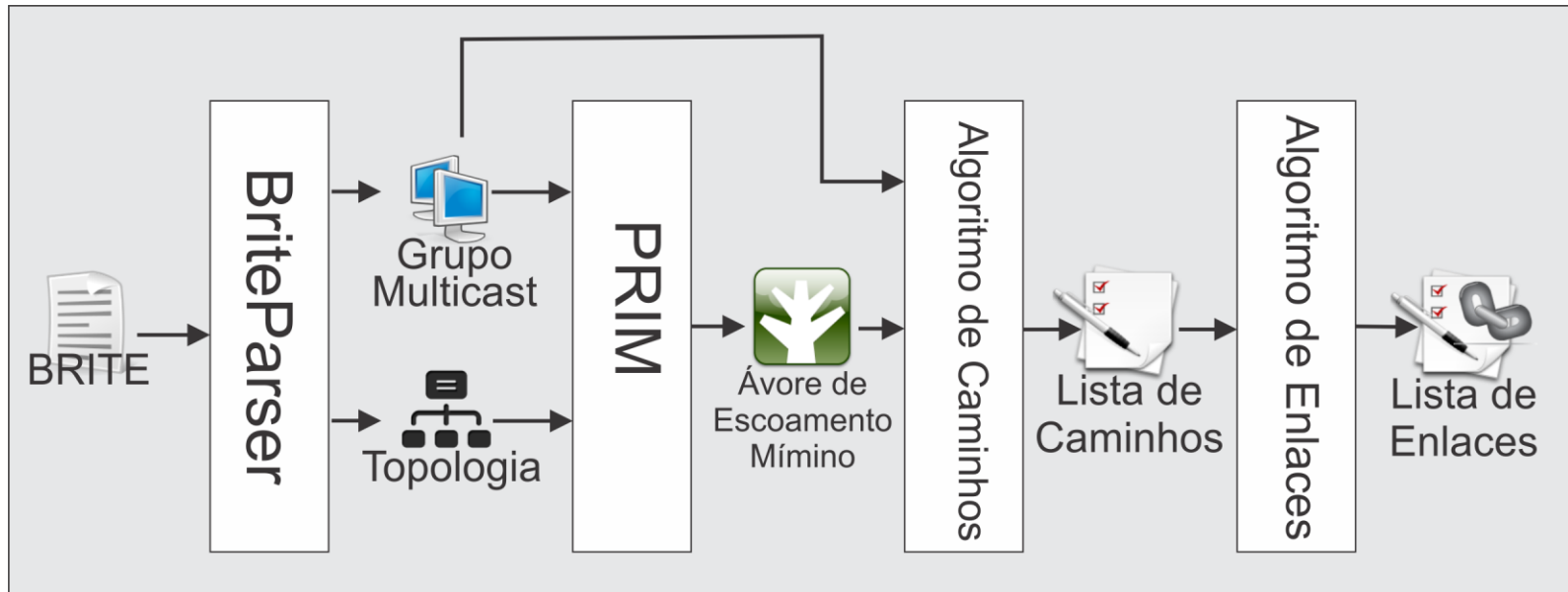
# Proposta CastFlow

- Idéias:
  - **Antecipar o cálculo das rotas** de distribuição multicast com o objetivo de **diminuir o tempo de processamento dos eventos de grupo** (entrada, saída e mudança de fonte).
    - Video Streaming – tem **alto churn**
  - **Centralização** da arquitetura multicast.
  - **Diminuir o número de mensagens de controle** (evitar inundações de mensagens na rede)
    - Usuários entram e saem do sistema (Packet\_IN) específico para o IP alocado para o grupo multicast, sem join/leave (IGMP)

# Funcionamento

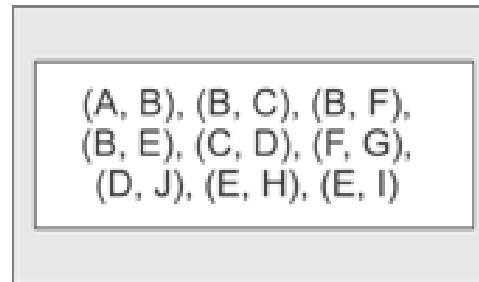
- 1ª Etapa: setup inicial
  - A topologia é obtida (*Datapath Join*)
  - As árvores são calculadas utilizando o algoritmo PRIM
  - As rotas são definidas (*Cálculo*)
- 2ª Etapa: instalação das rotas
  - Controlador distribui as rotas entre os swiches OpenFlow (*Flow Mods*)
- 3ª Etapa: Processamento dos eventos
  - Eventos: entrada, saída e mudança de fonte. (*Packet-Ins*)
  - Alterações mínimas nas árvores multicast (*Diffs*)

# Funcionamento

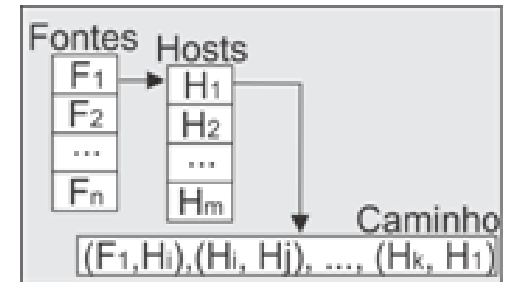


|       |                        |
|-------|------------------------|
| A → D | (A, B), (B, C), (C, D) |
| A → G | (A, B), (B, F), (F, G) |
| A → J | (A, B), (B, F), (F, J) |
| A → H | (A, B), (B, E), (E, H) |
| A → I | (A, B), (B, E), (E, I) |

(a) Lista de caminhos



(b) Lista de enlaces

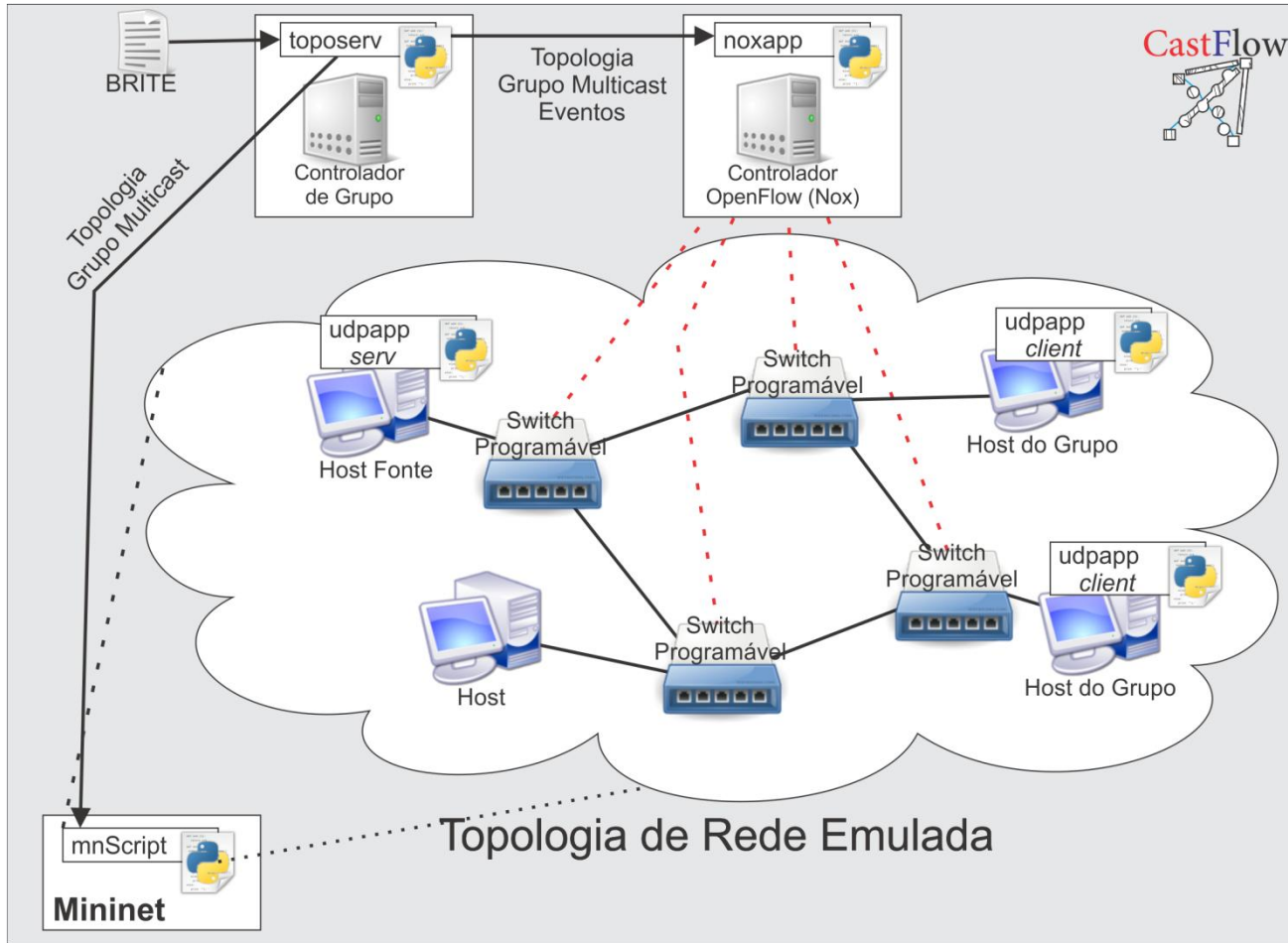


(c) Mapa de rotas

# Componentes do protótipo

- TopologyServer : gerencia a topologia de rede.
- MininetScript: emula máquinas virtuais e switches OpenFlow na rede com a topologia fornecida.
- NoxAPP: controlador OpenFlow com a lógica multicast centralizada.
- UDPAPP: aplicação cliente / servidor de mensagens UDP utilizada para testes de comunicação e distribuição dos dados multicast.

# Arquitetura



**DEMO**

- Estado Atual
  - Testbeds
    - Omni / VNext
    - OFELIA
    - RouteFlow
  - CastFlow
  - **Transcodificador**

# Transcodificador - Resultados

- Software desenvolvido na China (Universidade)
- Transcodificação em Tempo Real
- Código permite associar 2 endereços IP para realizar a transcodificação
- Idéia: Transcodificação na nuvem (baseado em VM Xen)
  - Preliminares: 90% CPU (bare machine)
  - Necessidade VM com **GPU**
- Grupo Cesar Teixeira também esta trabalhando paralelamente com transcodificador





# Transcodificador

- Criação de *imagem Bundle* Transcodificador
- Experimentos com VMs + Estudo com Nuvem (Amazon API)
  - Amazon EC2 Cluster instances (\$2 / hora)
  - JClouds acesso a Amazon via API
- Demonstração no OFELIA CF (usando Xen)