Avaliação de Desempenho de Fluxo Multimídia em Redes Definidas por Software (SDN)

JACKSON NUNES DA SILVA

Orientador: Prof. Eduardo Tavares

Abril de 2015

Roteiro da Apresentação

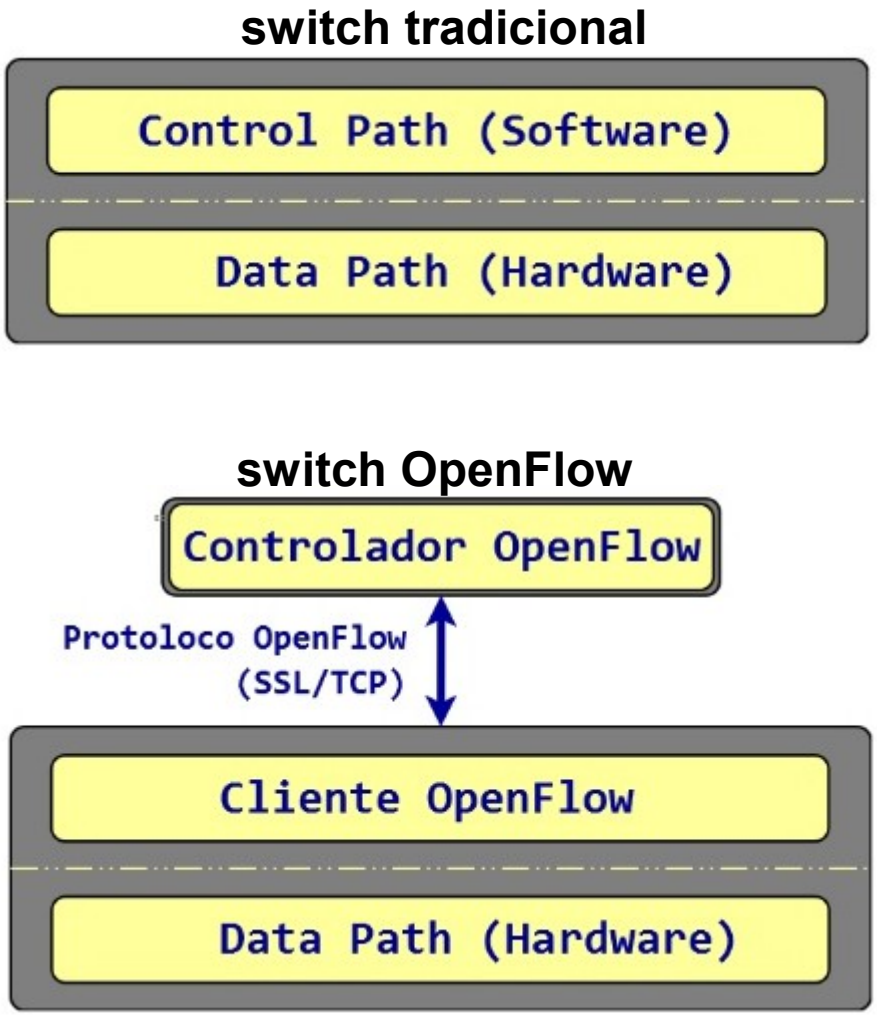
- Introdução
- Software Defined Network (SDN) e OpenFlow
- Objetivos
- Metodologia
- Arquiteturas Utilizadas nos Testes
- Resultados Preliminares
- Resultados Esperados

Jackson Nunes – jns@cin.ufpe.br

Introdução

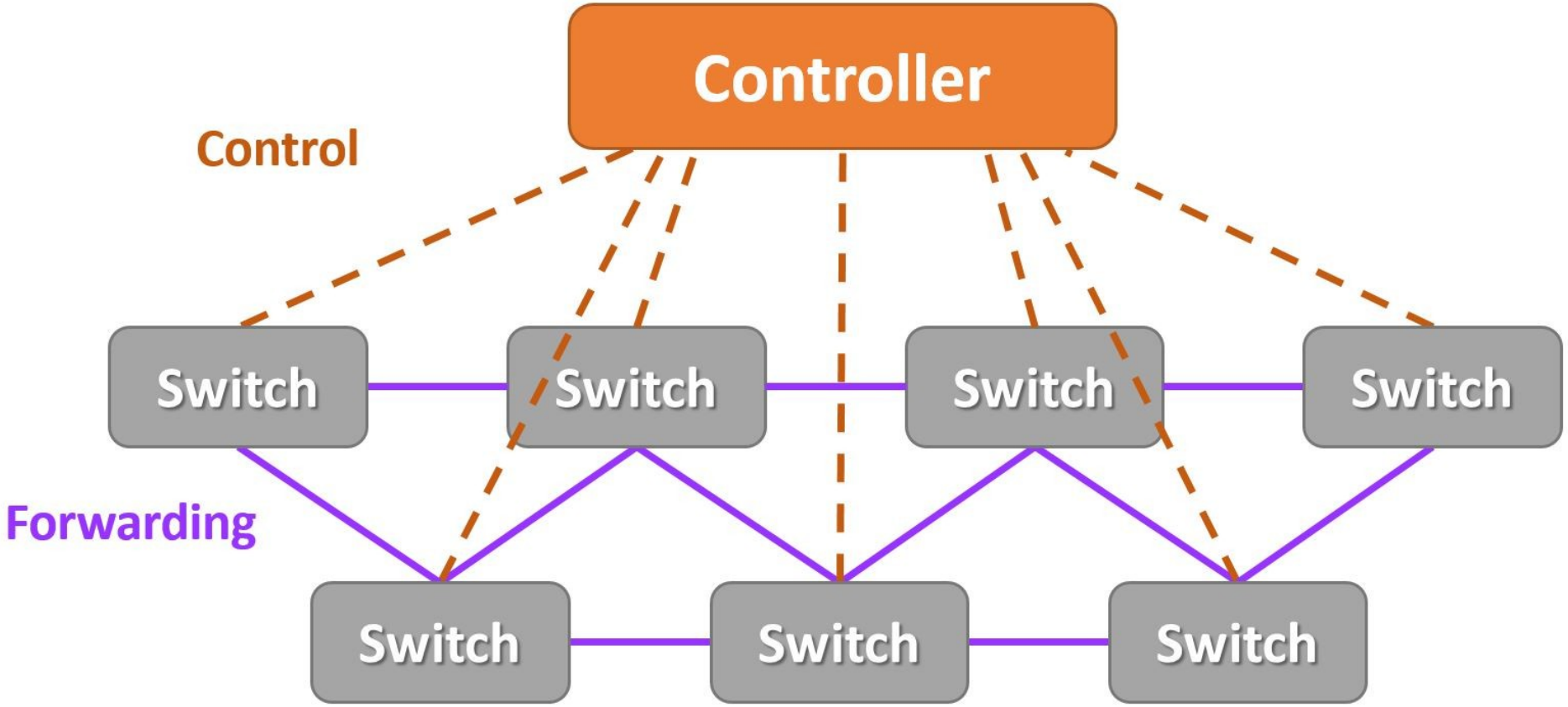
- Impacto da qualidade da rede em aplicações multimídia;
- Métricas que influenciam na Qualidade do Serviço para sistemas de tempo real (atraso, jitter, etc);
- Redução do degradação da qualidade do serviço de mídias de tempo real (RTP), em redes convergentes.
- Redes “Programáveis”;
- SDN/OpenFlow;

O que é SDN e OpenFlow?



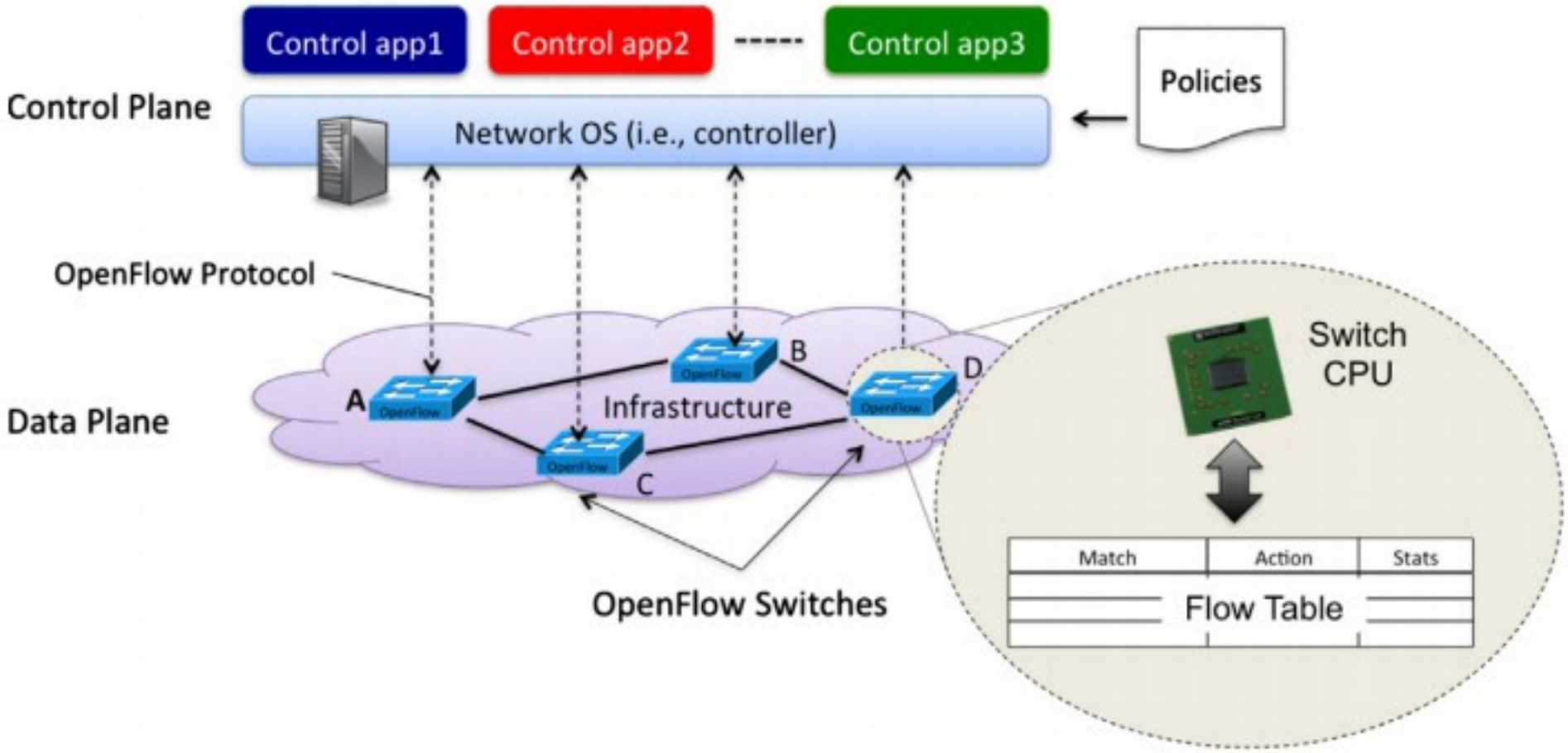
Jackson Nunes – jns@cin.ufpe.br

Controladora OpenFlow



Jackson Nunes – jns@cin.ufpe.br

SDN - Arquitetura



Jackson Nunes – jns@cin.ufpe.br

Objetivo Geral

Realizar a avaliação de desempenho de tráfego de dados multimídia, em Redes Definidas por Software (SDN), visando a melhoria do QoS, através da aplicação de modificações na controladora *OpenFlow*.

Jackson Nunes – jns@cin.ufpe.br

Objetivos Específicos

- Implantar infraestrutura SDN para o ambiente proposto;
- Implementar sistema de tempo real para a comunicação entre *hosts*;
- Executar uma aplicação de um comutador simples na controladora SDN;
- Avaliar o fluxo multimídia no switch simples (L2) e em cenários distintos de carga na rede;
- Explorar alterações na controladora SDN, modificando o switch e os serviços disponibilizados, visando melhorar o QoS;
- Refazer os testes e a avaliação de desempenho, com metodologia e fluxo de dados equivalente;
- Criar modelos para a infraestrutura de rede utilizada e aplicações;
- Testar e validar os modelos.

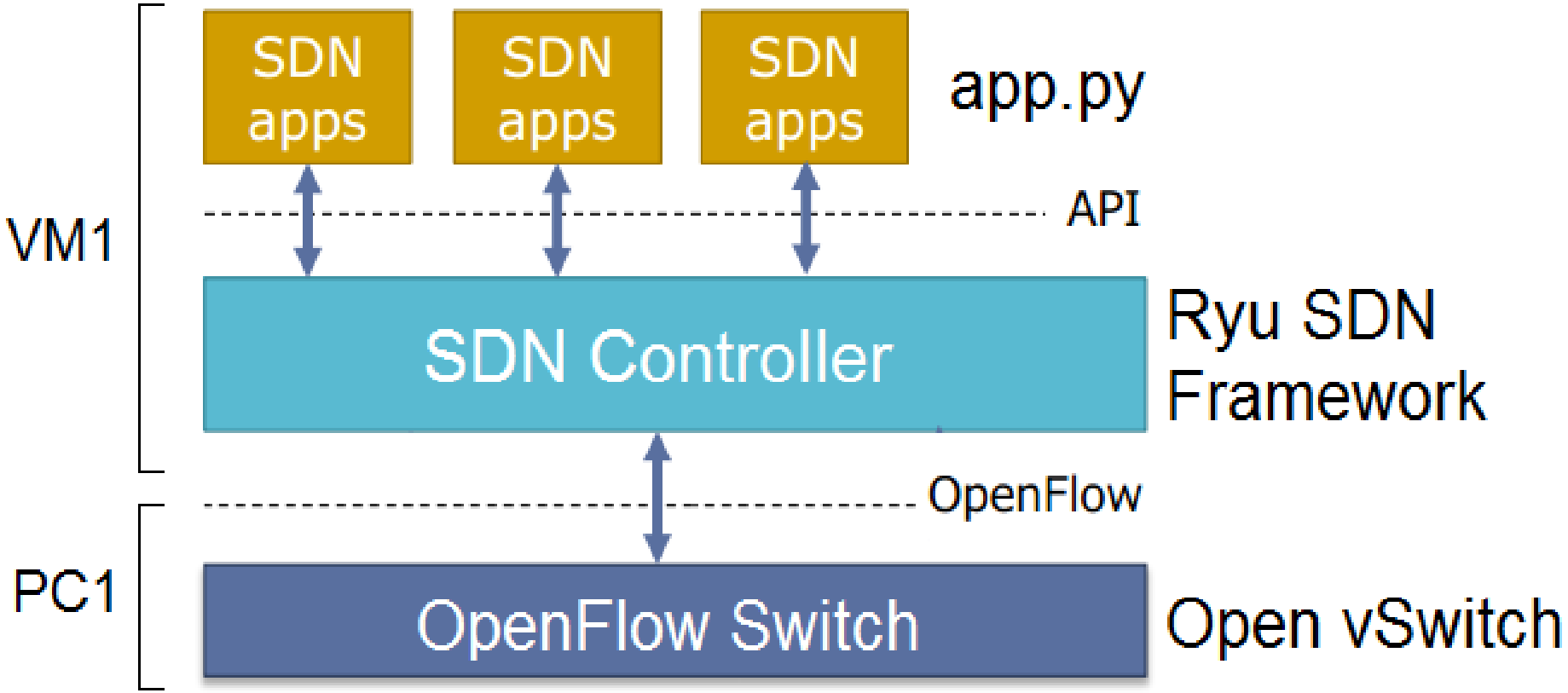
Metodologia

- Infraestrutura de rede para o ambiente proposto, utilizando switchs e hosts virtualizados (VirtualBox, Open vSwitch, Ryu SDN Framework, etc);
- Sistema de comunicação VoIP, implantado através de centrais telefônicas Asterisk;
- Scripts em Python para adicionar os serviços à Controladora (simple_switch.py, qos.py, isolation.py, etc);
- *Software* para a geração de tráfego concorrente (Iperf, etc);
- Wireshark para a captura de pacotes e análise do *Streaming* (protocolo RTP) das chamadas;
- Técnica de modelagem (rede de Petri, etc) para representação da arquitetura de rede e das aplicações vinculadas à controladora;

Objetivos Específicos

- Implantar infraestrutura SDN para o ambiente proposto;
- Implementar sistema de tempo real para a comunicação entre *hosts*;
- Executar uma aplicação de um comutador simples na controladora SDN;
- Avaliar o fluxo multimídia no switch simples (L2), em cenários distintos de carga na rede;
- Explorar alterações na controladora SDN, modificando o switch e os serviços disponibilizados, visando melhorar o QoS;
- Refazer os testes e a avaliação de desempenho, com metodologia e fluxo de dados equivalente;
- Criar modelos para a infraestrutura de rede utilizada e aplicações;
- Testar e validar os modelos.

Testes Preliminares - Arquitetura SDN

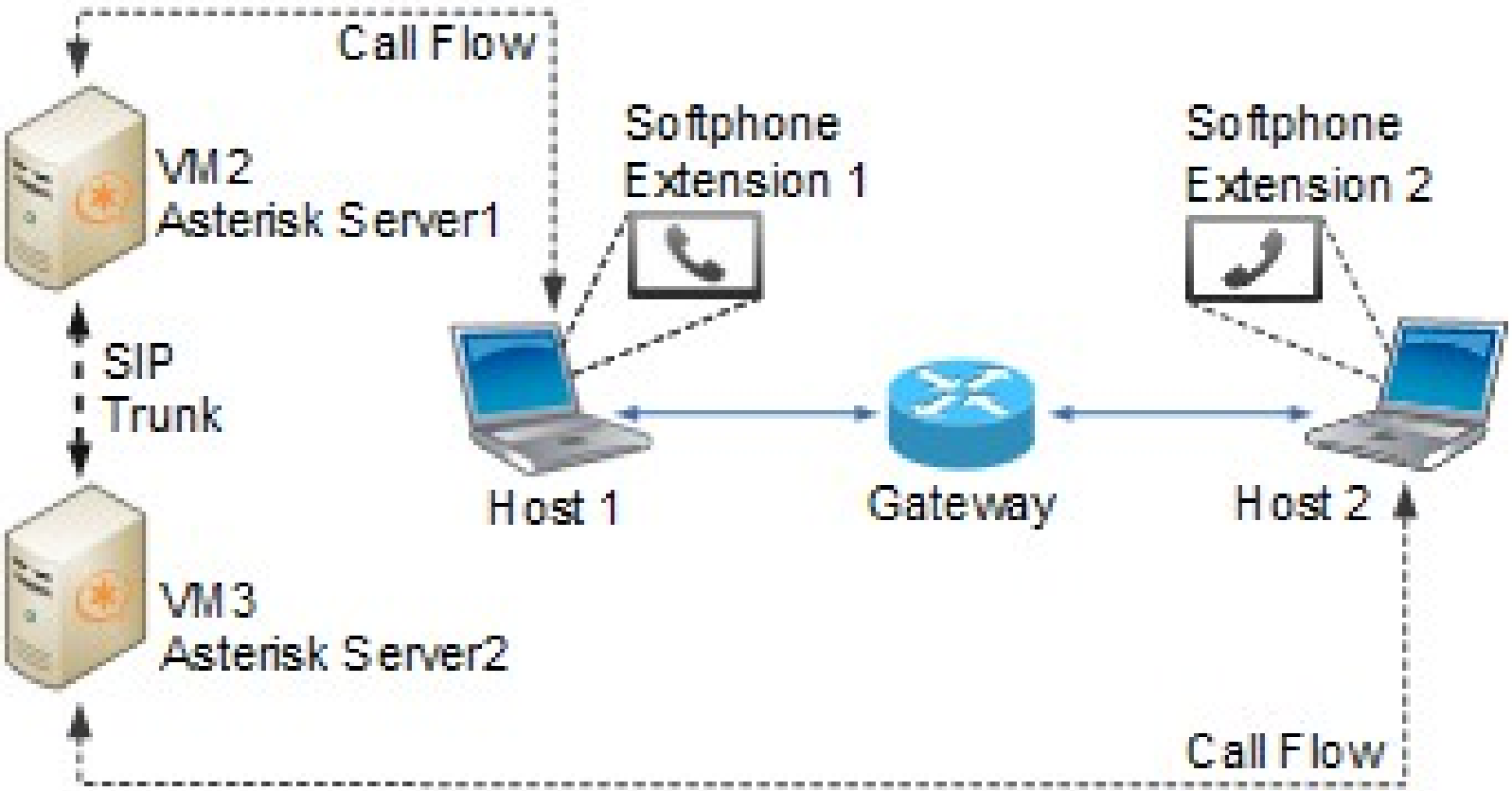


Jackson Nunes – jns@cin.ufpe.br

Objetivos Específicos

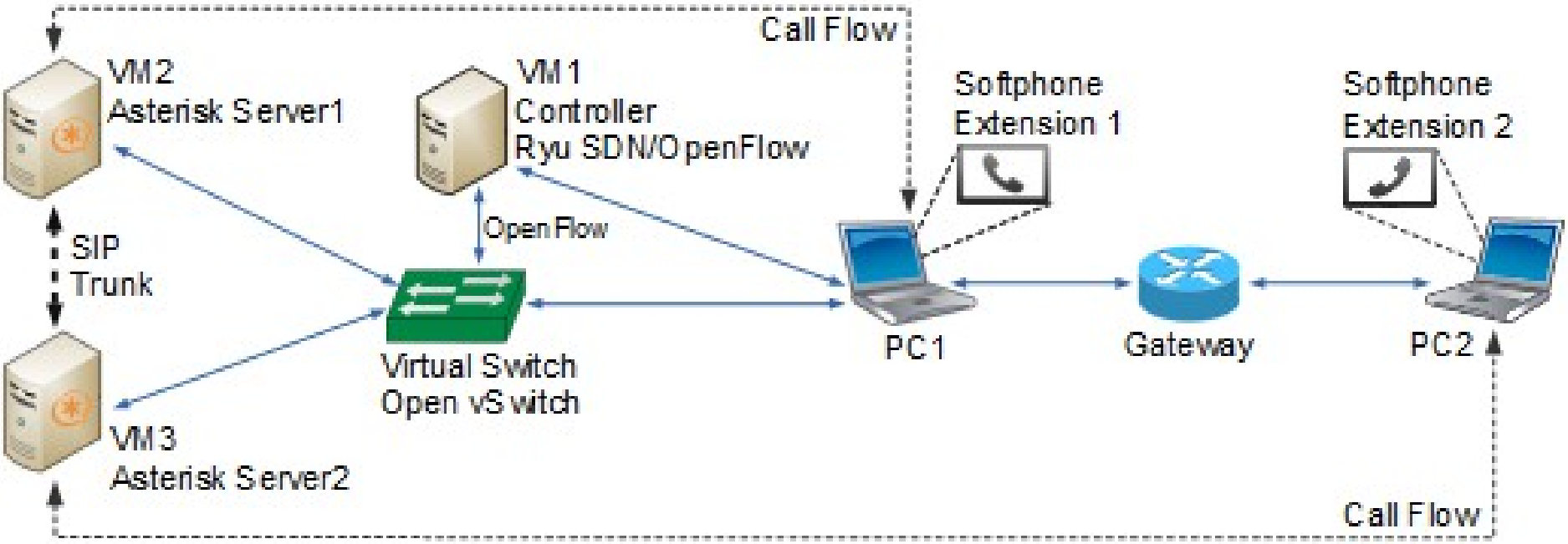
- Implantar infraestrutura SDN para o ambiente proposto;
- Implementar sistema de tempo real para a comunicação entre *hosts*;
- Executar uma aplicação de um comutador simples na controladora SDN;
- Avaliar o fluxo multimídia no switch simples (L2), em cenários distintos de carga na rede;
- Explorar alterações na controladora SDN, modificando o switch e os serviços disponibilizados, visando melhorar o QoS;
- Refazer os testes e a avaliação de desempenho, com metodologia e fluxo de dados equivalente;
- Criar modelos para a infraestrutura de rede utilizada e aplicações;
- Testar e validar os modelos.

Arquitetura VoIP



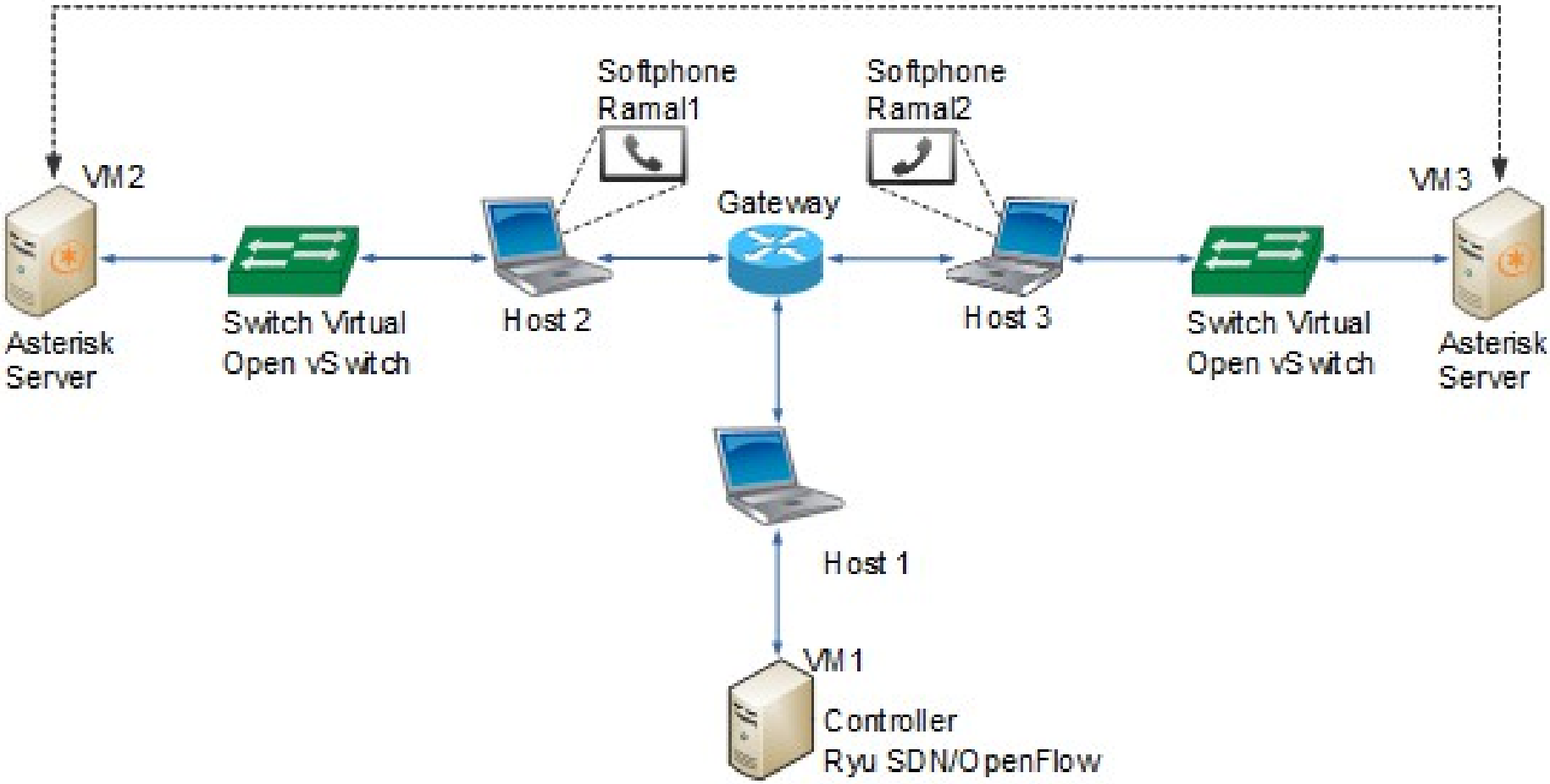
Jackson Nunes – jns@cin.ufpe.br

Testes – Arquitetura 1



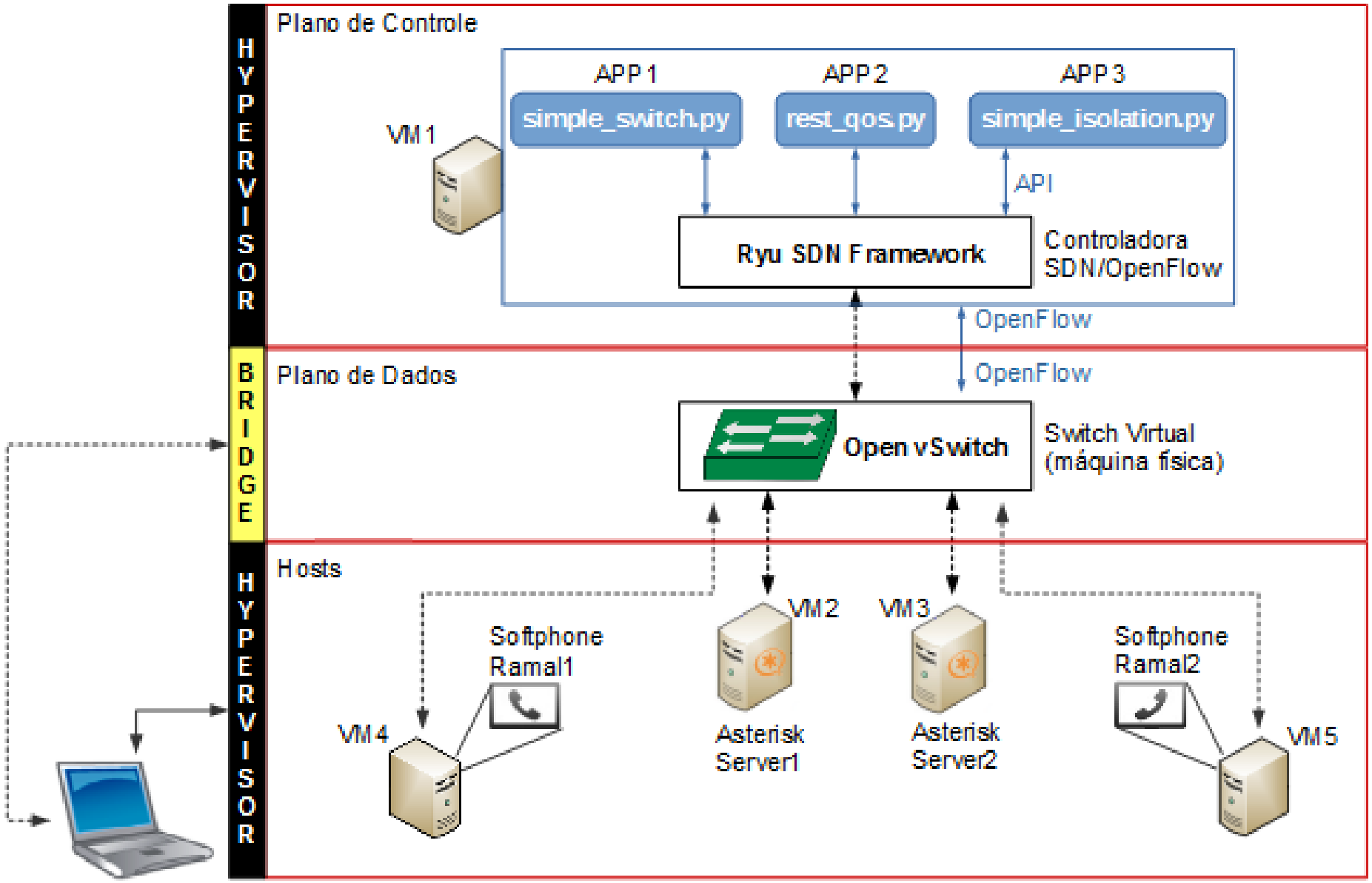
Jackson Nunes – jns@cin.ufpe.br

Testes – Arquitetura 2



Jackson Nunes – jns@cin.ufpe.br

Testes – Arquitetura 3

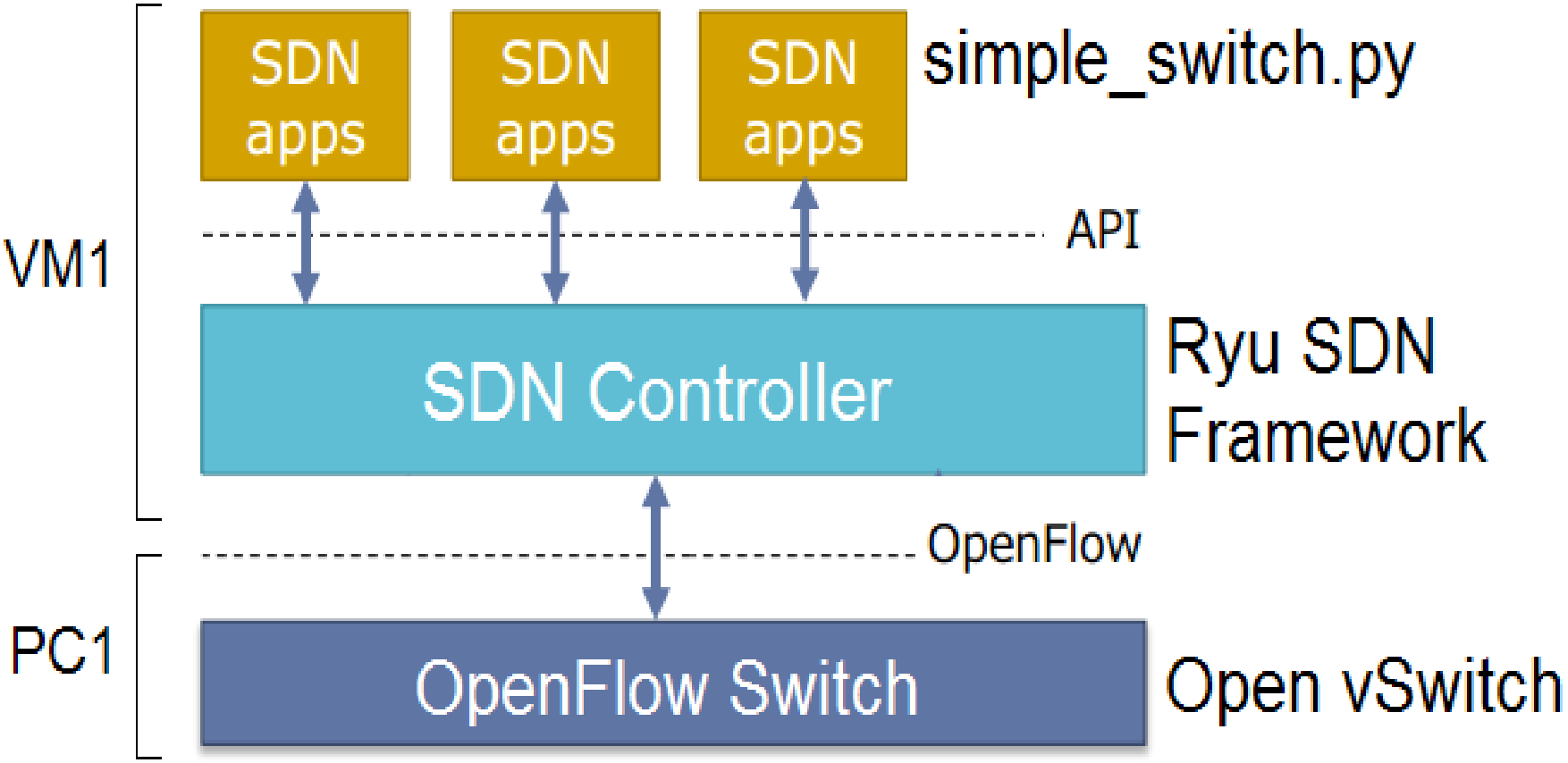


Jackson Nunes – jns@cin.ufpe.br

Objetivos Específicos

- Implantar infraestrutura SDN para o ambiente proposto;
- Implementar sistema de tempo real para a comunicação entre *hosts*;
- Executar uma aplicação de um comutador simples na controladora SDN;
- Avaliar o fluxo multimídia no switch simples (L2), em cenários distintos de carga na rede;
- Explorar alterações na controladora SDN, modificando o switch e os serviços disponibilizados, visando melhorar o QoS;
- Refazer os testes e a avaliação de desempenho, com metodologia e fluxo de dados equivalente;
- Criar modelos para a infraestrutura de rede utilizada e aplicações;
- Testar e validar os modelos.

Arquitetura SDN – app simple_switch.py



Jackson Nunes – jns@cin.ufpe.br

App - simple_switch.py

```
class SimpleSwitch(app_manager.RyuApp):
    OFP_VERSIONS = [ofproto_v1_0.OFP_VERSION]

    def __init__(self, *args, **kwargs):
        super(SimpleSwitch, self).__init__(*args, **kwargs)
        self.mac_to_port = {}

    def add_flow(self, datapath, in_port, dst, actions):
        ofproto = datapath.ofproto

        match = datapath.ofproto_parser.OFPMatch(
            in_port=in_port, dl_dst=haddr_to_bin(dst))

        mod = datapath.ofproto_parser.OFPFlowMod(
            datapath=datapath, match=match, cookie=0,
            command=ofproto.OFPFC_ADD, idle_timeout=0, hard_timeout=0,
            priority=ofproto.OFP_DEFAULT_PRIORITY,
            flags=ofproto.OFPFF_SEND_FLOW_REM, actions=actions)
        datapath.send_msg(mod)
```


Jackson Nunes – jns@cin.ufpe.br

Objetivos Específicos

- Implantar infraestrutura SDN para o ambiente para proposto;
- Implementar sistema de tempo real para a comunicação entre *hosts*;
- Executar uma aplicação de um comutador simples na controladora SDN;
- Avaliar o fluxo multimídia no switch simples (L2), em cenários distintos de carga na rede;
- Explorar alterações na controladora SDN, modificando o switch e os serviços disponibilizados, visando melhorar o QoS;
- Refazer os testes e a avaliação de desempenho, com metodologia e fluxo de dados equivalente;
- Criar modelos para a infraestrutura de rede utilizada e aplicações;
- Testar e validar os modelos.

Resultados dos testes: simple_switch.py

- Exemplo de Análise de Streaming de voz no Wireshark

Wireshark: RTP Streams

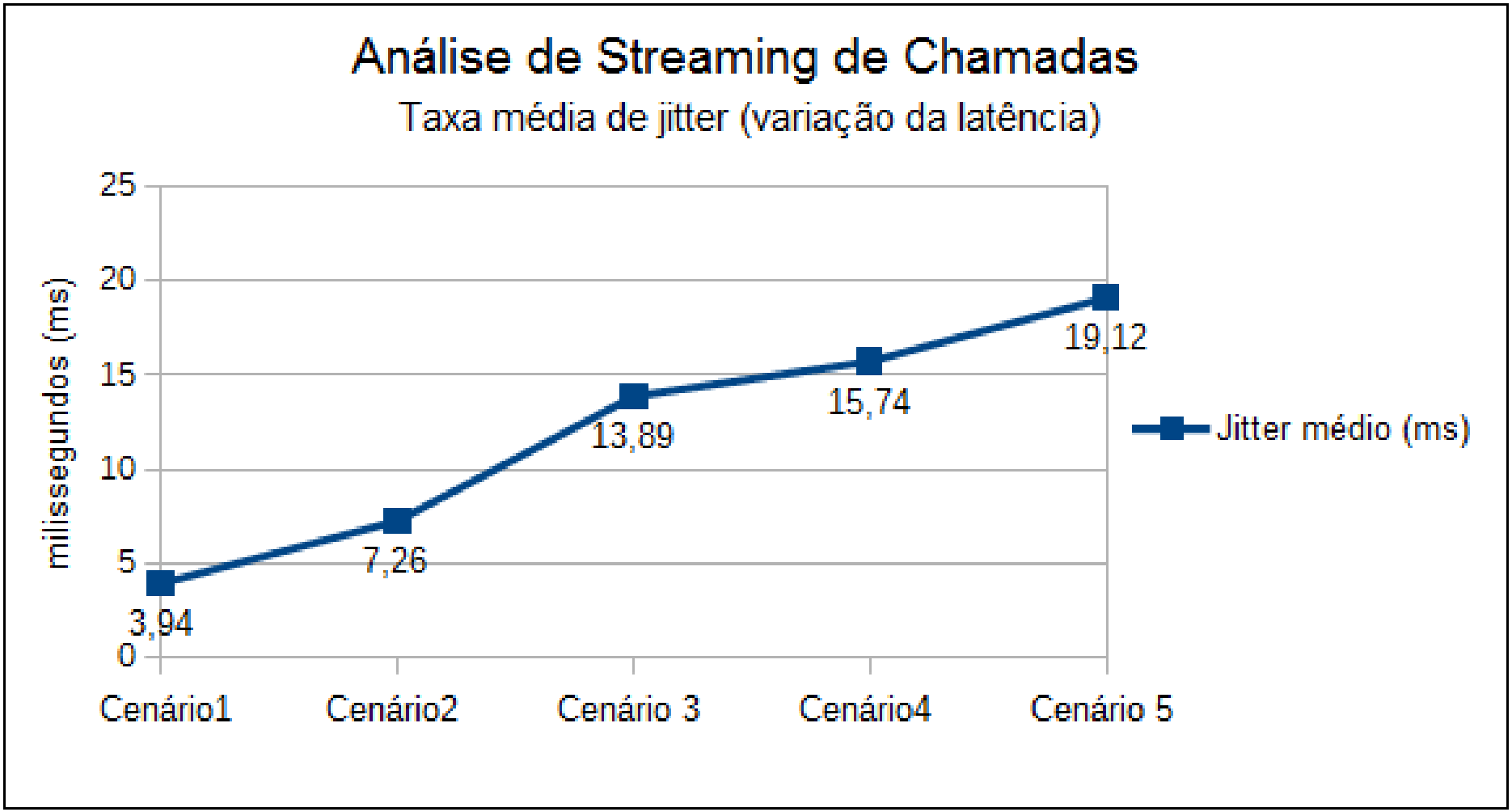
Detected 4 RTP streams. Choose one for forward and reverse direction for analysis

Src addr ▲	Src port	Dst addr	Dst port	SSRC	Payload	Packets	Lost	Max Delta (ms)	Max Jitter (ms)	Mean Jitter (ms)
192.168.1.100	30864	192.168.1.105	15608	0x71EA0D5B	g711U	3002	51 (1.7%)	739.17	63.04	8.01
192.168.1.101	40046	192.168.1.106	10934	0x2833	g711A	2946	55 (1.8%)	501.80	10.78	5.45
192.168.1.105	15608	192.168.1.100	30864	0xF966C30	g711U	2949	52 (1.7%)	514.56	13.41	6.33
192.168.1.106	10934	192.168.1.101	40046	0x183C5C0F	g711A	2953	51 (1.7%)	735.51	62.79	8.27

Forward: 192.168.1.100:30864 -> 192.168.1.105:15608, SSRC=0x71EA0D5B
Reverse: 192.168.1.105:15608 -> 192.168.1.100:30864, SSRC=0xF966C30

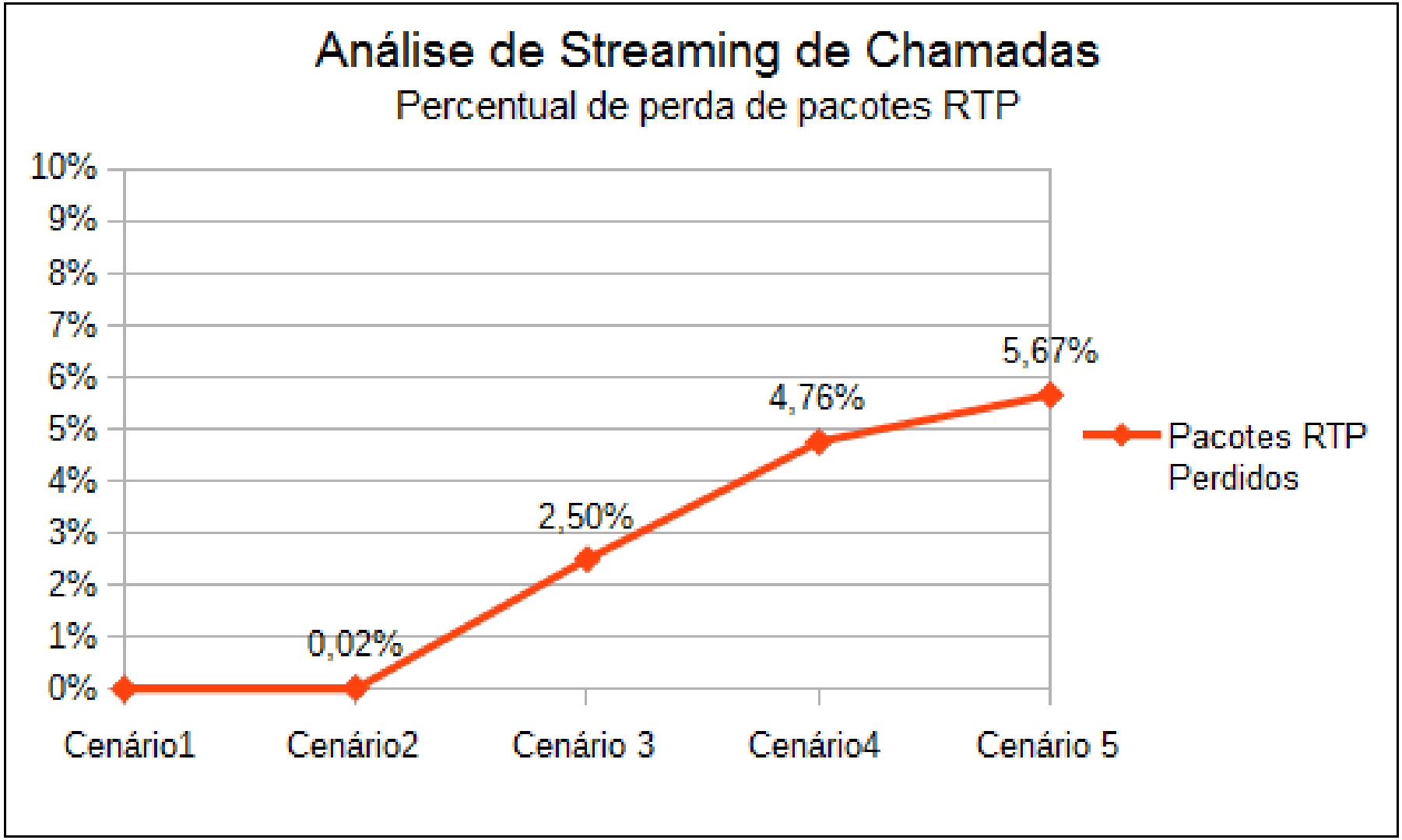
Jackson Nunes – jns@cin.ufpe.br

Resultados dos testes: simple_switch.py



Jackson Nunes – jns@cin.ufpe.br

Resultados dos testes: simple_switch.py



Jackson Nunes – jns@cin.ufpe.br

Objetivos Específicos

- Implantar infraestrutura SDN para o ambiente proposto;
- Implementar sistema de tempo real para a comunicação entre *hosts*;
- Executar uma aplicação de um comutador simples na controladora SDN;
- Avaliar o fluxo multimídia no switch simples (L2), em cenários distintos de carga na rede;
- Explorar alterações na controladora SDN, modificando o switch e os serviços disponibilizados, visando melhorar o QoS;
- Refazer os testes e a avaliação de desempenho, com metodologia e fluxo de dados equivalente;
- Criar modelos para a infraestrutura de rede utilizada e aplicações;
- Testar e validar os modelos.

Apps Ryu SDN

- rest_firewall.py
- rest_nw_id.py
- rest_qos.py
- rest_quantum.py
- rest_router.py
- rest_topology.py
- rest_tunnel.py
- simple_isolation.py
- simple_switch.py

Jackson Nunes – jns@cin.ufpe.br

App - rest_qos.py

```
class RestQoSAPI(app_manager.RyuApp):
    OFP_VERSIONS = [ofproto_v1_0.OFP_VERSION,
                    ofproto_v1_2.OFP_VERSION,
                    ofproto_v1_3.OFP_VERSION]
    _CONTEXTS = {
        'dpset': dpset.DPSet,
        'conf_switch': conf_switch.ConfSwitchSet,
        'wsgi': WSGIApplication}

    def __init__(self, *args, **kwargs):
        super(RestQoSAPI, self).__init__(*args, **kwargs)

        # logger configure
        QoSController.set_logger(self.logger)
        self.cs = kwargs['conf_switch']
        self.dpset = kwargs['dpset']
        wsgi = kwargs['wsgi']
        self.waiters = {}
        self.data = {}
        self.data['dpset'] = self.dpset
        self.data['waiters'] = self.waiters
        wsgi.registory['QoSController'] = self.data
        wsgi.register(QoSController, self.data)
```


Jackson Nunes – jns@cin.ufpe.br

App - simple_isolation.py

```
class SimpleIsolation(app_manager.RyuApp):

    _CONTEXTS = {
        'network': network.Network,
        'dpset': dpset.DPSet,
    }
    def __init__(self, *args, **kwargs):
        super(SimpleIsolation, self).__init__(*args, **kwargs)
        self.nw = kwargs['network']
        self.dpset = kwargs['dpset']
        self.mac2port = mac_to_port.MacToPortTable()
        self.mac2net = mac_to_network.MacToNetwork(self.nw)

    @set_ev_cls(ofp_event.EventOFPSwitchFeatures,
CONFIG_DISPATCHER)
    def switch_features_handler(self, ev):
        msg = ev.msg
        datapath = msg.datapath

        datapath.send_delete_all_flows()
        datapath.send_barrier()
```

Resultados Esperados

- Avaliação do desempenho do fluxo de dados de um sistema de tempo real, implantado em uma infraestrutura de *Software Defined Network* (SDN);
- Avaliação do QoS do sistema, através da realização de testbed, antes e após as modificações realizadas na controladora SDN;
- Diminuir a influência do tráfego concorrente na rede, em relação ao desempenho do sistema de tempo real e, reduzir a perda de pacotes RTP (Real Time Protocol).

Perguntas?

Jackson Nunes – jns@cin.ufpe.br

Obrigado!

Jackson Nunes
jns@cin.ufpe.br