

# ANÁLISE DOS PADRÕES DE QOS EM TRÁFEGO DE VOZ SOBRE IP EM LINUX EMBARCADO



Daniel Bristot de Oliveira; Ramicés dos Santos Silva  
Curso de Ciência da Computação – CTTMAR  
Campus São José – Universidade do Vale do Itajaí – São José – SC – Brasil.  
daniel@bristot.eti.br ramicés@univali.br

# Agenda

- Introdução
- Justificativa
- Objetivos
- Fundamentação
  - QoS em VoIP
  - Padrões de QoS
- Desenvolvimento
  - QoS no Kenel Linux
  - Experimentos
  - Resultados
- Conclusões
- Contribuições
- Trabalhos Futuros

# Introdução

- VoIP
- Vantagens do VoIP
- PSNT vs IP
- Protocolos de qualidade de serviço
- Dispositivos VoIP

# Justificativa

- Necessidade de mercado
- Definição dos parâmetros do kernel
- Definição das ferramentas necessárias
- Instalação como sistema embarcado
- Análise de desempenho

# Objetivo Geral

Estudar o suporte do kernel Linux as técnicas de qualidade de serviço aplicadas a dispositivos compatíveis com as tecnologias VoIP e avaliar o desempenho da utilização destes protocolos de qualidade de serviço em tráfego de voz sobre IP em Linux embarcado.

# Objetivos específicos

- estudar e definir as tecnologias utilizadas para implementação de QoS em plataformas com limitações de recursos;
- definir fatores relevantes (parâmetros) envolvidos na solução para que possam ser avaliados quanto a contribuição para a QoS de VoIP e as métricas que definem a qualidade desse serviço;

# Objetivos específicos

- montar e configurar a estrutura de testes com as tecnologias definidas;
- definir, através da execução de testes experimentais, a influência de cada fator nas métricas de qualidade de serviço; e
- avaliar os resultados, ajustar os parâmetros e repetir os testes buscando validar a otimização da escolha dos parâmetros.

# Qualidade de serviço em VoIP

- Largura de banda
- Perda
- Delay
- Jitter

# Padrões de QoS

## Layer 2:

- IEEE 802.1p
- IEEE 802.1q

## Layer 3:

- ToS
- DiffServ

## Traffic Shapping:

- Garantia de banda
- Priorização

# Experimentos

- Análise de desempenho
- Projeto de experimentos
- Metodologia  $2^k$
- Definição dos fatores e níveis
- Grau de contribuição de cada fator

# Desenvolvimento

## Buildroot

### Ambiente de Desenvolvimento:

- Compilador/Linker...
- Bibliotecas

### Instalação do Linux como sistema embarcado:

- Sistema básico
- Ferramentas de suporte a QoS

# Desenvolvimento

## O Kernel Linux

- Configuração do kernel mínimo
- Suporte as técnicas e padrões de QoS
  - Interface VLAN: IEEE 802.1p e 802.1q
  - Netfilter: ToS e DiffServ
  - Qdisc: *Traffic Shapping*
- Kernels utilizados nos experimentos
- Testes e validação

# Desenvolvimento

## Parâmetros

### Layer 2:

CONFIG\_VLAN\_8021Q=y

### Layer 3:

CONFIG\_NETFILTER=y  
CONFIG\_NETFILTER\_ADVANCED=y  
CONFIG\_NETFILTER\_NETLINK=y  
CONFIG\_NETFILTER\_NETLINK\_QUEUE=y  
CONFIG\_NETFILTER\_XTABLES=y  
CONFIG\_NETFILTER\_XT\_TARGET\_CLASSIFY=y  
CONFIG\_NETFILTER\_XT\_TARGET\_DSCP=y  
CONFIG\_NETFILTER\_XT\_MATCH\_DSCP=y  
CONFIG\_NETFILTER\_XT\_MATCH\_MULTIPORT=y  
CONFIG\_IP\_NF\_IPTABLES=y  
CONFIG\_IP\_NF\_MANGLE=y

### Shapping:

CONFIG\_NET\_SCHED=y  
CONFIG\_NET\_SCH\_CBQ=y  
CONFIG\_NET\_SCH\_HTB=y  
CONFIG\_NET\_SCH\_HFSC=y  
CONFIG\_NET\_CLS=y  
CONFIG\_NET\_CLS\_U32=y  
CONFIG\_CLS\_U32\_PERF=y  
CONFIG\_NET\_CLS\_ACT=y  
CONFIG\_NET\_SCH\_FIFO=y  
CONFIG\_IFB=y

# Desenvolvimento

## Configuração do ambiente:

### - Embarcado

Técnicas de QoS

Ferramentas de benchmark (VoIP e Iperf)

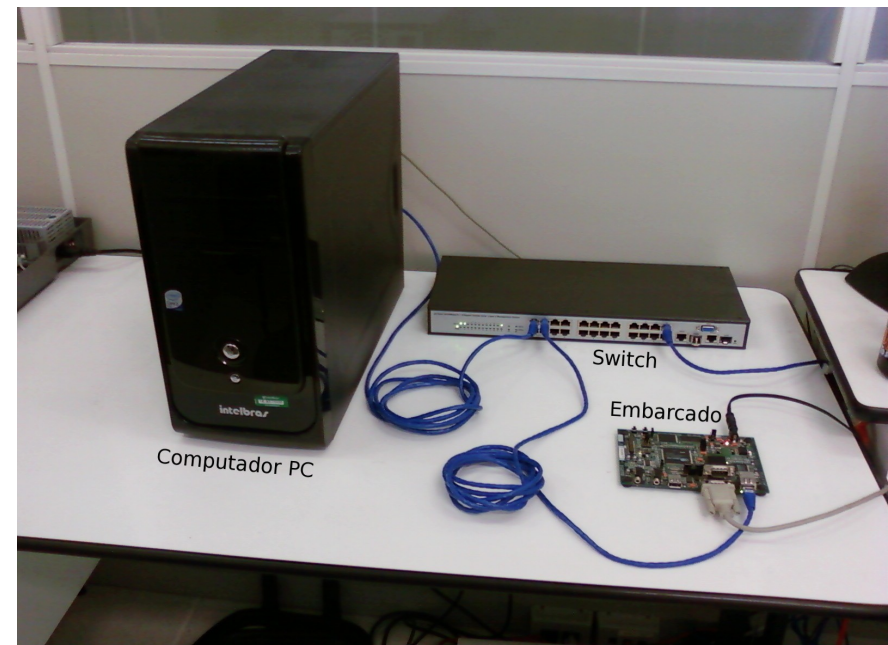
### - Switch

QoS e Limitação de banda

### - PC

Ferramentas de benchmark

Automatização



# Desenvolvimento

## Automatização:

### PC:

```
#!/bin/sh
cd /rundir/
retval=1;
lperf -s
while [ $retval -ne 0 ]; do
    ./client 10.0.1.2 G711
    retval=$?
done
rm estatistica
./client 10.0.1.2 G711
mv estatistica `date +%Y%m%d%H%M` .out
./cliente_reboot 10.0.1.2
sleep 30
reboot
```

### Embarcado:

```
#!/bin/sh
# Executa os scripts que configurarão as regras
# de qualidade de serviço
sh /tc.sh
#inicia o tráfego concorrente
iperf -c 10.0.1.1
# Inicia o servidor da ferramenta de benchmark
./server &
```

# Parâmetros para os experimentos

## Fatores:

- A – QoS em camada 2;
- B – QoS em camada 3;
- C – Garantia de banda; e
- D – Priorização.

## Parâmetros:

- 40 replicações
- 60 segundos warmup
- 60 segundos experimentos
- reinicialização do ambiente a cada iteração

# Resultados dos experimentos

## Fatores analisados

- Jitter
- Perda
- Uso de CPU

## Crítérios

- Margem de erro
- Fatores que influenciaram mais que o erro

# Resultados

## Jitter

| <b>Fator</b> | <b>% de contribuição</b> |
|--------------|--------------------------|
| Shapping – C | 99.36                    |
| Erro         | 0.61                     |

## Perda

| <b>Fator</b> | <b>% de contribuição</b> |
|--------------|--------------------------|
| Shapping – C | 99.85                    |
| Erro         | 0.14                     |

## Cpu

| <b>Fator</b> | <b>% de contribuição</b> |
|--------------|--------------------------|
| Shapping – C | 96.23                    |
| Erro         | 2.48                     |

# Conclusão

Definidos os parâmetros do kernel e aplicações

Testados e validados no sistema embarcado

Garantia de largura de banda = PSTN

Uso de CPU adicional = +1.2%

Utilização de ambas as técnicas

Traffic Shapping

Demais para o auxílio no gerenciamento

# Contribuições

- definição dos fatores e métricas de qualidade de serviço em VoIP;
- definição dos parâmetros de configuração do kernel Linux para o suporte às técnicas e padrões de qualidade de serviço;
- documentação da montagem do ambiente experimental;
- resultados consistentes e confiáveis em função da metodologia utilizada;
- análise da utilização das técnicas e padrões de qualidade que podem servir de apoio a projetos de dispositivos VoIP.

# Trabalhos futuros

- Comparativo de desempenho dos escalonadores HTB, HFSC e CBQ
- identificação de melhorias nos escalonadores de tráfego;
- desenvolvimento de um guia para a utilização da metodologia de projetos de experimentos aplicados a análise quantitativa para a área de redes de computadores
- execução dos experimentos em ambientes diferentes, como em um ambiente real de rede de computadores;
- análise de desempenho das técnicas de qualidade de serviço à outros tráfegos de tempo real;
- análise de desempenho das técnicas de qualidade de serviço utilizando o patch de tempo real do kernel Linux.