



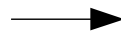
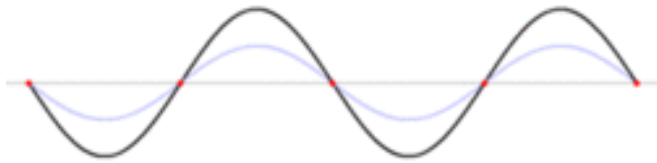
**VoIP e mitos: por que a voz
picota, atrasa... QoS e seus desafios**

Agenda

- **Conceitos de voz digital**
- **Gargalos de transmissão - perda de pacotes**
- **Conceito de QoS**
- **Conceito de largura de banda, latência, jitter...**
- **DiffServ**
- **Dicas de implementação**
- **Perguntas e respostas**

Conceito de voz digital

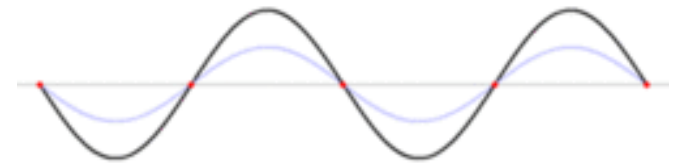
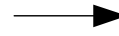
Codificação



0010111011011110011011110100111

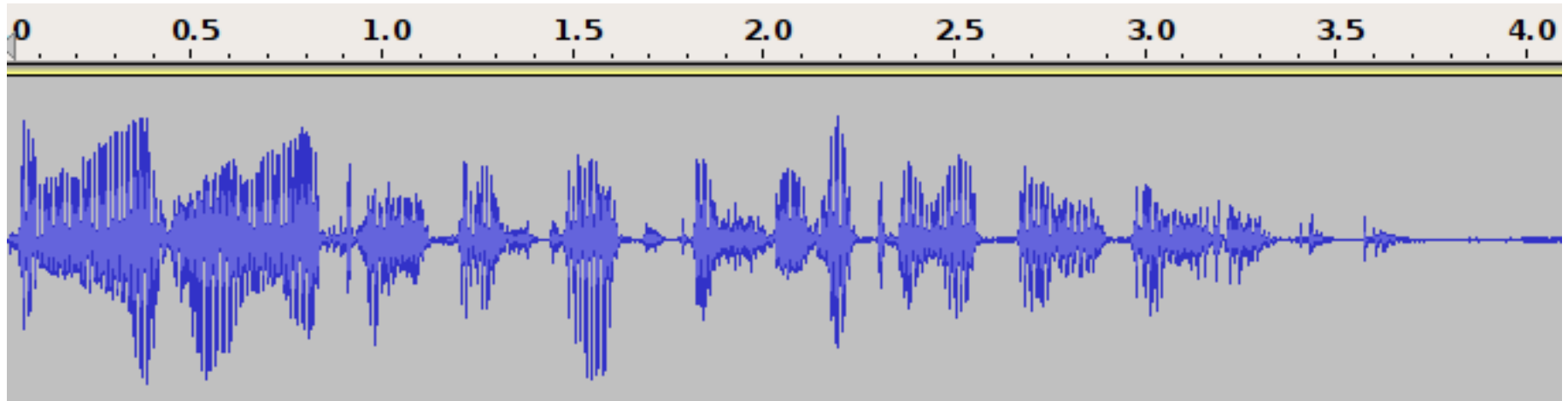
Decodificação

0010111011011110011011110100111



Codec - processo que converte a voz em dados, e vice-versa

Pneumoultramicroscopicossilicovulcanoconiótico



GSM



```
0000100100000000111000000101000000  
0000000000000000001100000000000000
```

54384 bits / 6798 bytes

54384 bits / 4,12 segundos = 13200 bits/segundo (± 13 Kbps)

Conceito de voz digital - Transmissão

000001001000000001110000001010000000
000000000000000000011000000000000000

54384 bits / 6798 bytes

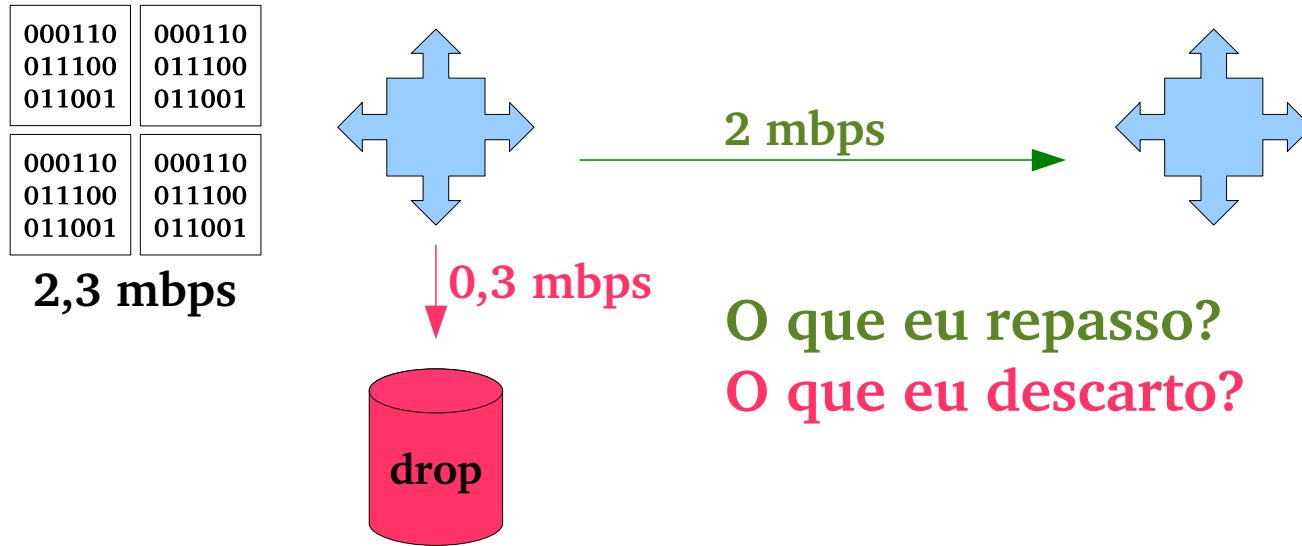
$54384 \text{ bits} / 4,12 \text{ segundos} = 13200 \text{ bits/segundo } (\pm 13 \text{ Kbps})$

Pacotes com 20 ms de áudio
4,12 segundos = 206 pacotes de 264 bits cada

000110	000110	000110	000110	000110	000110	000110	000110	000110	000110	000110	000110	000110	000110	000110	000110	000110
011100	011100	011100	011100	011100	011100	011100	011100	011100	011100	011100	011100	011100	011100	011100	011100	011100
011001	011001	011001	011001	011001	011001	011001	011001	011001	011001	011001	011001	011001	011001	011001	011001	011001



Gargalos de transmissão



QoS = Quality of Service

Trabalhar na escassez
Prioridades diferentes para serviços diferentes

Se não há escassez, esqueça QoS!

QoS - Conceito

QoS /kju/ou/es/: ato de ferrar com a vida de alguns pacotes para que alguns outros tenham uma vida boa.

QoS não ajuda em nada, só atrapalha!

**A questão está em: quem vamos atrapalhar hoje?
E qual será nosso grau de malvadeza? :-D**

**Só existe QoS para o que você envia.
O que você recebe já foi recebido...**

Conceitos de rede

000110
011100
011001



leva 1 pacote em 1 segundo
leva 1 pacote por vez
logo, 5 pacotes = 5 segundos

000110	000110	000110	000110	000110
011100	011100	011100	011100	011100
011001	011001	011001	011001	011001



leva 1 pacote em 3 segundos
leva até 5 pacotes por vez
logo, 5 pacotes = 3 segundos

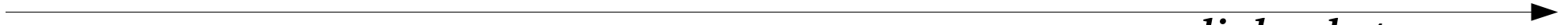
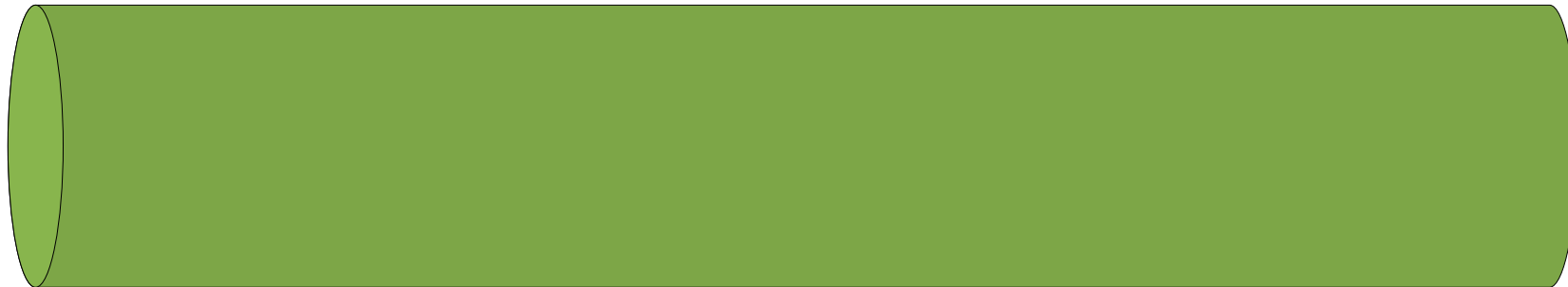
Conceitos de rede [2]



velocidade - latência - medida em tempo



largura de banda - vazão (*bandwidth*)
bits / segundo

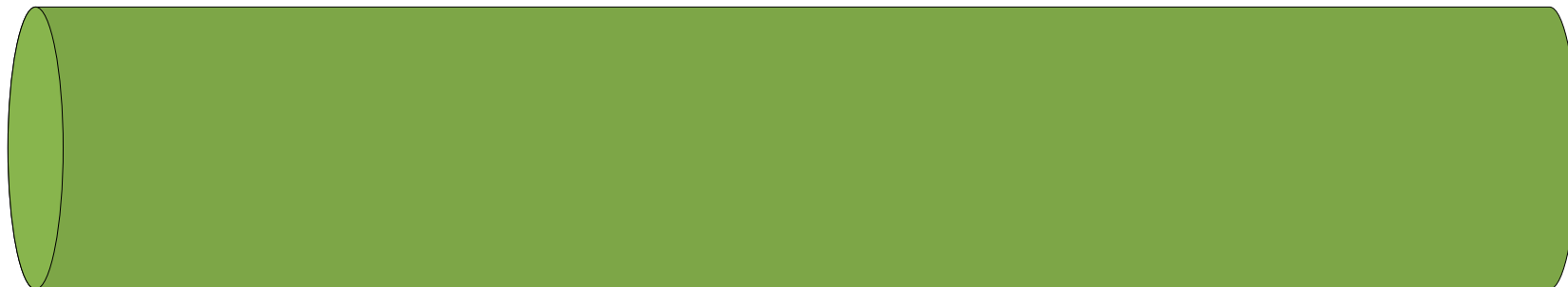


linha de tempo

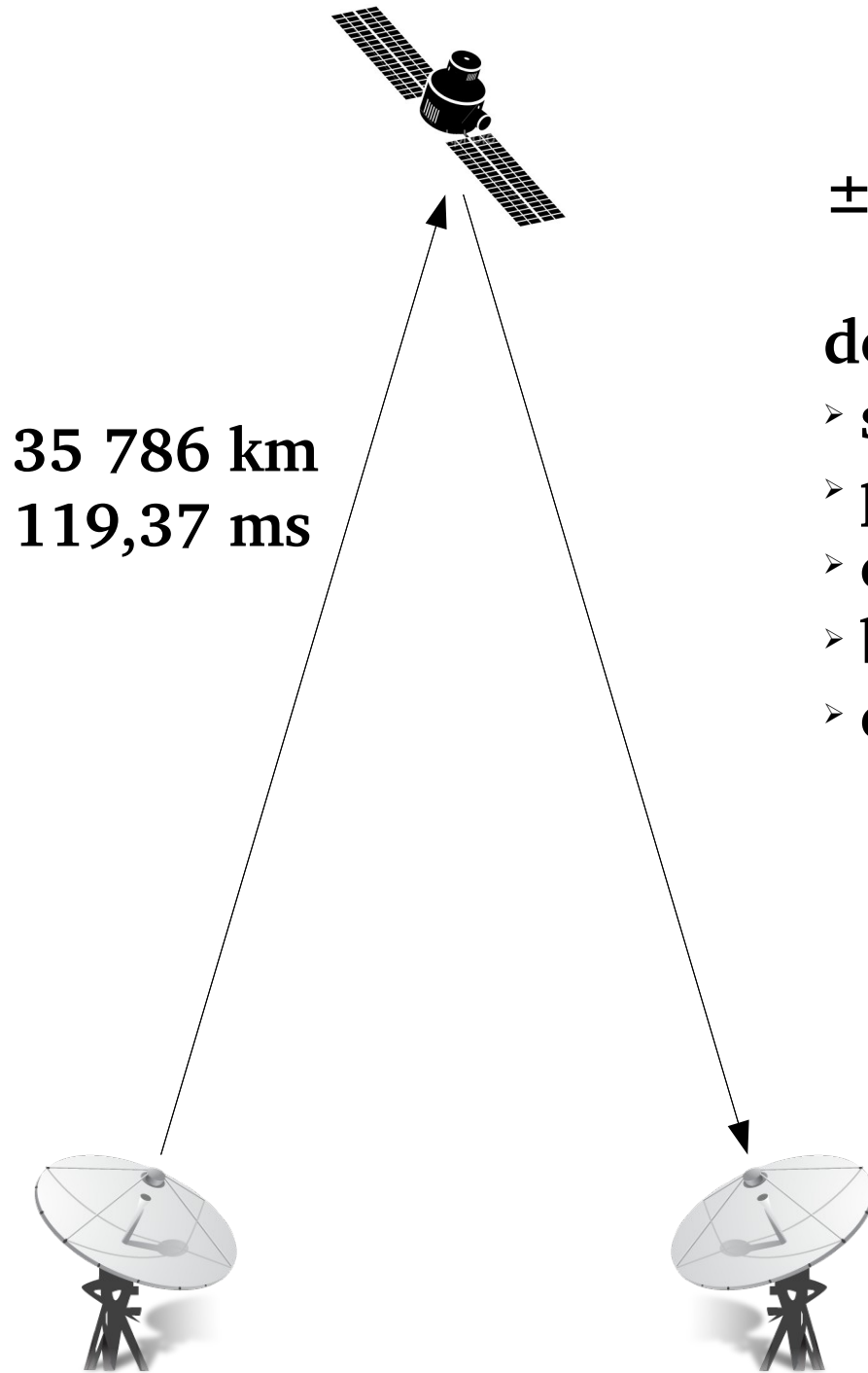


Velocidade ou latência: comprimento do cano

Largura de banda: diâmetro do cano



Conceitos de rede - Exemplo satélite geoestacionário



± 240 ms ida-e-volta

desprezando-se outras latências

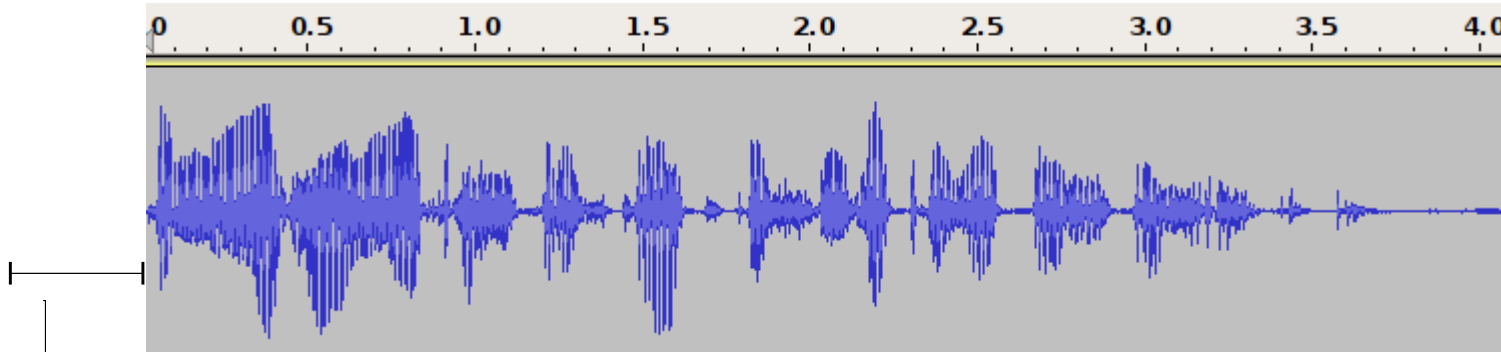
- sinalização
- processamento
- compressão
- buffering
- etc...

**Não confundir as grandes
obras do Mestre Picasso
com o grande picasso do
Mestre de Obras**

-- Alguém espirituoso

Largura de banda não implica velocidade!

Latência e jitter



latência: exemplo 150 ms constante

jitter: latência variando - alguns pacotes atrasam mais que outros

Latência e jitter - Exemplo

```
% ping -nc 10 1.2.3.4
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
64 bytes from 1.2.3.4: icmp_seq=1 ttl=52 time=136 ms
64 bytes from 1.2.3.4: icmp_seq=2 ttl=52 time=238 ms
64 bytes from 1.2.3.4: icmp_seq=3 ttl=52 time=129 ms
64 bytes from 1.2.3.4: icmp_seq=4 ttl=52 time=124 ms
64 bytes from 1.2.3.4: icmp_seq=5 ttl=52 time=160 ms
64 bytes from 1.2.3.4: icmp_seq=6 ttl=52 time=183 ms
64 bytes from 1.2.3.4: icmp_seq=7 ttl=52 time=118 ms
64 bytes from 1.2.3.4: icmp_seq=8 ttl=52 time=253 ms
64 bytes from 1.2.3.4: icmp_seq=9 ttl=52 time=230 ms
64 bytes from 1.2.3.4: icmp_seq=10 ttl=52 time=96.7 ms
```

--- 1.2.3.4 ping statistics ---

10 packets transmitted, 10 received, 0% packet loss, time 9000ms

rtt min/avg/max/mdev = 96.743/167.153/253.324/53.252 ms

mdev = desvio médio

Latência e jitter - Exemplo mais eficaz

```
% ping -i 0.02 -s 33 -w 10 1.2.3.4
```

```
--- 1.2.3.4 ping statistics ---
```

```
486 packets transmitted, 438 received, 9% packet loss, time 9993ms  
rtt min/avg/max/mdev = 0.134/195.846/504.396/152.369 ms
```

-i 0.02 = 50 pacotes por segundo / 1 pacote a cada 0,02 segundos

-s 33 = 33 bytes de dados em cada pacote (simulando GSM)

-w 10 = encerra o teste em 10 segundos

ATENÇÃO: cuidado com ping. Alguns peers limitam a resposta de ping de propósito, para evitar um DoS

Dica: utilize também o **mtr**, para ver o estado em cada hop

Pacote IPv4:

bit offset	0-3	4-7	8-15	16-18	19-31
0	Version	Header length	Differentiated Services	Total Length	
32	Identification			Flags	Fragment Offset
64	Time to Live	Protocol		Header Checksum	
96	Source Address				
128	Destination Address				
160	Options				
160 or 192+	Data				

8 bits por pacote para classificação de tráfego e notificação de congestionamento (ECN)

Assured Forwarding (AF) Behavior Group

	Class 1	Class 2	Class 3	Class 4
AF: Low Drop	AF11	AF21	AF31	AF41
Med Drop	AF12	AF22	AF32	AF42
High Drop	AF13	AF23	AF33	AF43

EF: expedited forwarding - prioridade máxima

CS: class selector - compatibilidade com o ToS antigo

DiffServ - Marcação

Telefones IP normalmente suportam marcação DiffServ e/ou ToS
(não necessariamente estão habilitados por padrão)

Asterisk: sip.conf

```
tos_sip=cs3           ; Sets TOS for SIP packets.  
tos_audio=ef         ; Sets TOS for RTP audio packets.  
tos_video=af41       ; Sets TOS for RTP video packets.
```

*OBS: no Linux, apenas processos rodando
como root podem marcar pacotes*

```
iptables -t mangle -A OUTPUT -p udp -j DSCP --set-dscp-class EF  
iptables -t mangle -A FORWARD -p udp -j DSCP --set-dscp-class EF
```

Confira com tcpdump:

```
tcpdump -vn ...  
    IP (tos 0xb8, ttl...
```

0xb8 = 10111000 = DiffServ EF

DiffServ - Respeitar marcação

Internet normalmente opera em FIFO, não respeitando marcação
FIFO: first in, first out

Roteadores Linux:

Kernel com advanced routing habilitado: `pfifo_fast`

`pfifo_fast`: ToS antigo (não DiffServ)

Kernel sem advanced routing habilitado: `fifo`

Exemplo de implementação DiffServ no Linux:

<http://hackers.propus.com.br/~marlon/files/qos-diffserv.sh>

Roteadores como Cisco, 3Com, Huawei, Juniper, etc, suportam DiffServ normalmente. A configuração pode ser um pouco complicada de fazer.

Redes MPLS são vendidas com DiffServ fim-a-fim

- **Primeiro de tudo: verifique o estado de seu link sem tráfego**
 - **Se ele não for bom, QoS nenhum no mundo vai ajudar**
- **Esqueça o termo “reserva de banda”**
- **Você precisa frear e baixar a prioridade do resto do tráfego**
 - **Lembre-se: QoS existe para atrapalhar os outros**
- **Evite fila nos roteadores. Se o seu uplink é 400 Kbps, entregue o tráfego a uns 360 Kbps para o roteador**
- **Não confie no que o seu provedor lhe vendeu. Teste você mesmo e defina o seu uplink (use o iperf)**
- **Alguns tipos de circuitos aumentam a latência chegando próximo do limite de banda**
- **Experimente, teste, experimente, teste, experimente...**
- **KISS: keep it short and simple - não invente moda**

Sua cabeça está explodindo? Sim, normal!

?

Perguntas

Marlon Dutra

<marlon@propus.com.br>

<http://hackers.propus.com.br/~marlon/>



<http://www.propus.com.br>