Uso de Bridges Linux no Controle de Tráfego entre Sub-Redes em Uma Mesma Rede Lógica

Ricardo Kléber M. Galvão
(rk@ufrn.br)

http://naris.info.ufrn.br

Núcleo de Atendimento e
Resposta a Incidentes de Segurança

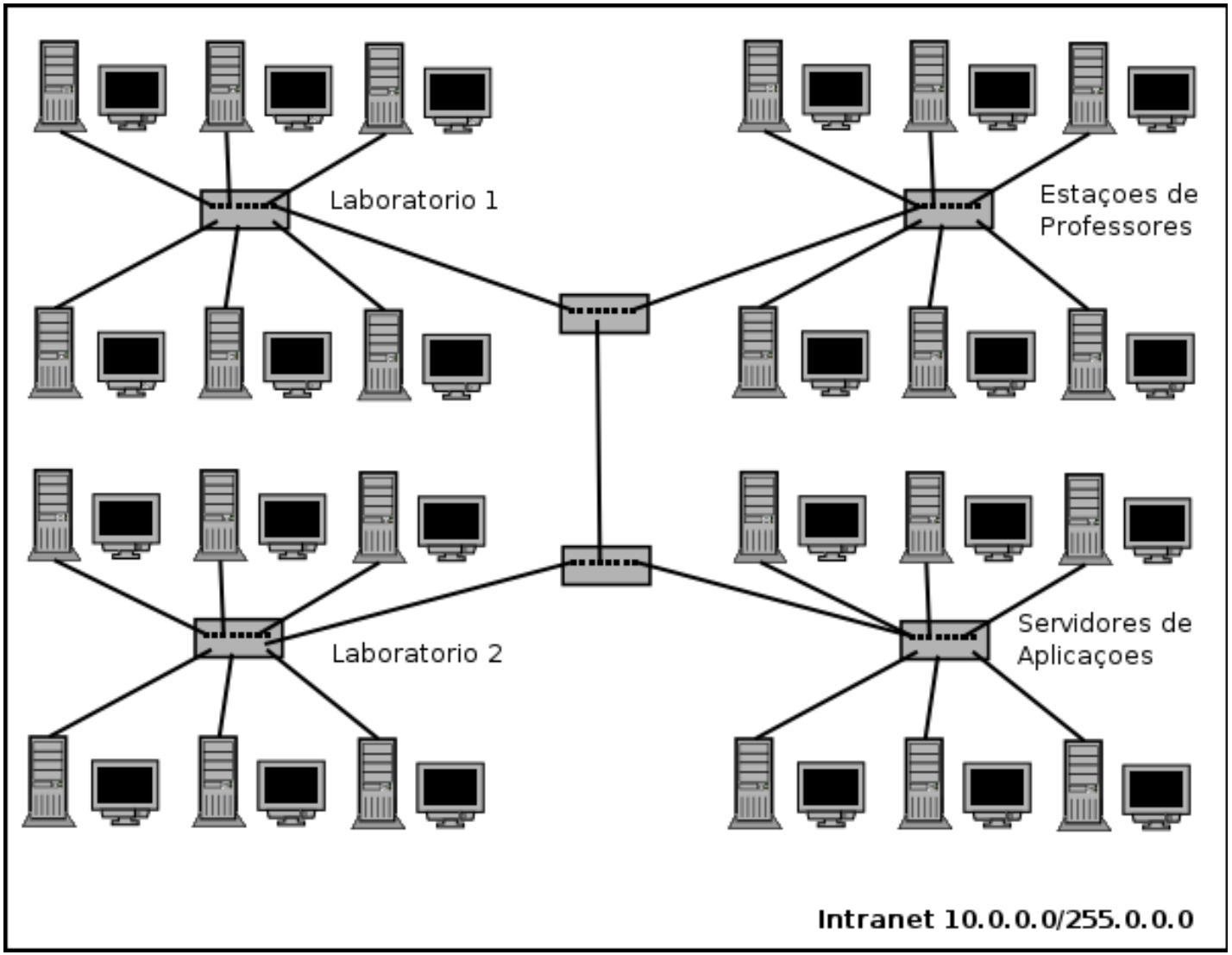
Resumo da Apresentação

Apresentação de detalhes de implementação de uma solução (utilizando software livre) para controle de tráfego entre sub-redes (e/ou máquinas) de uma mesma rede lógica com o uso de bridges (Segmentação física sem reconfiguração lógica)

Visão Geral

- Padrão de projetos de endereçamento de Intranets: rede lógica única
  - Facilidade de
    - implementação
    - manutenção
    - expansão (atribuição de endereços a novas máquinas)
    - acesso entre estações e servidores
      - compartilhamento de arquivos
      - atribuição dinâmica de endereços usando broadcast (DHCP)

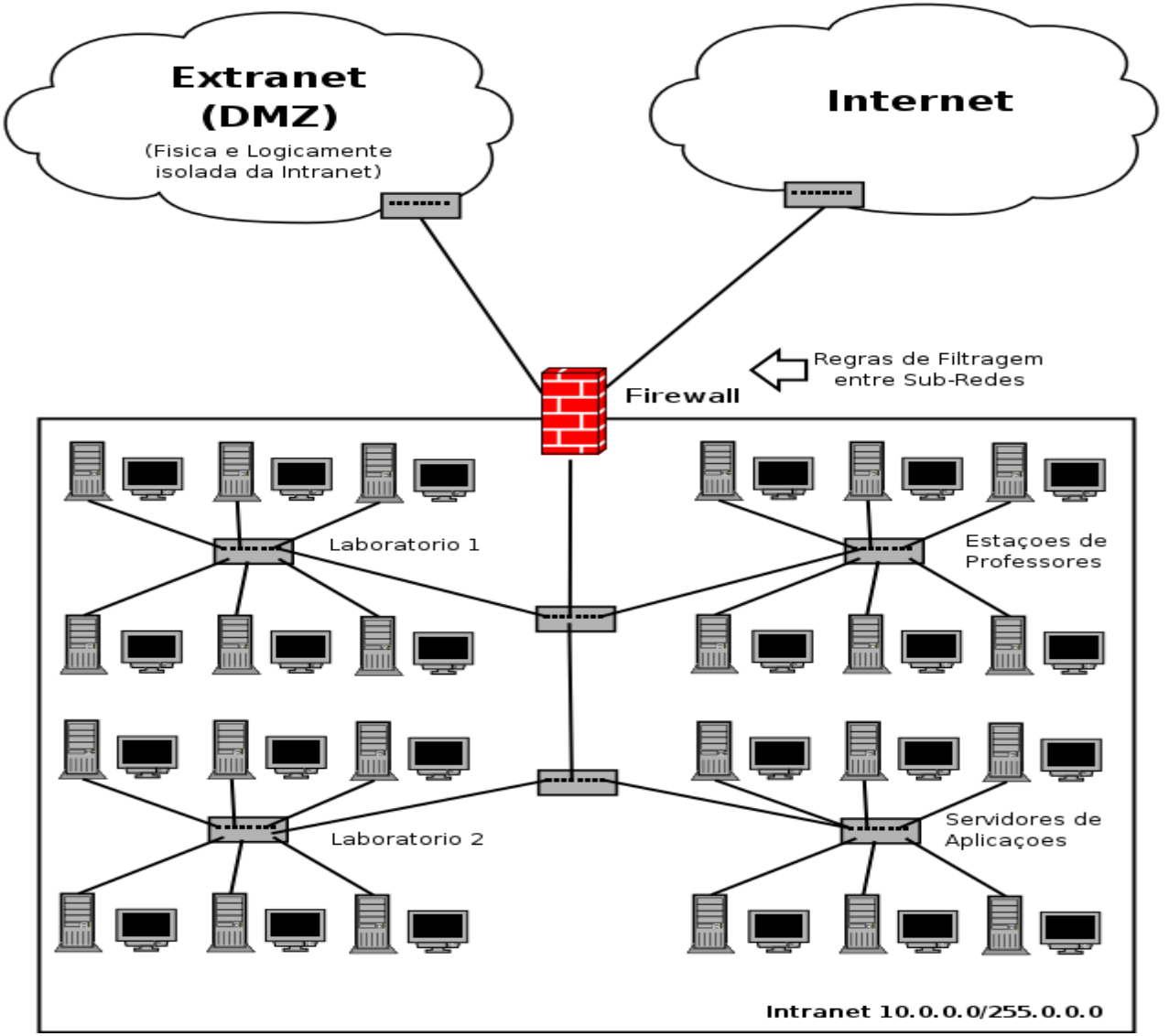
Rede Lógica Única



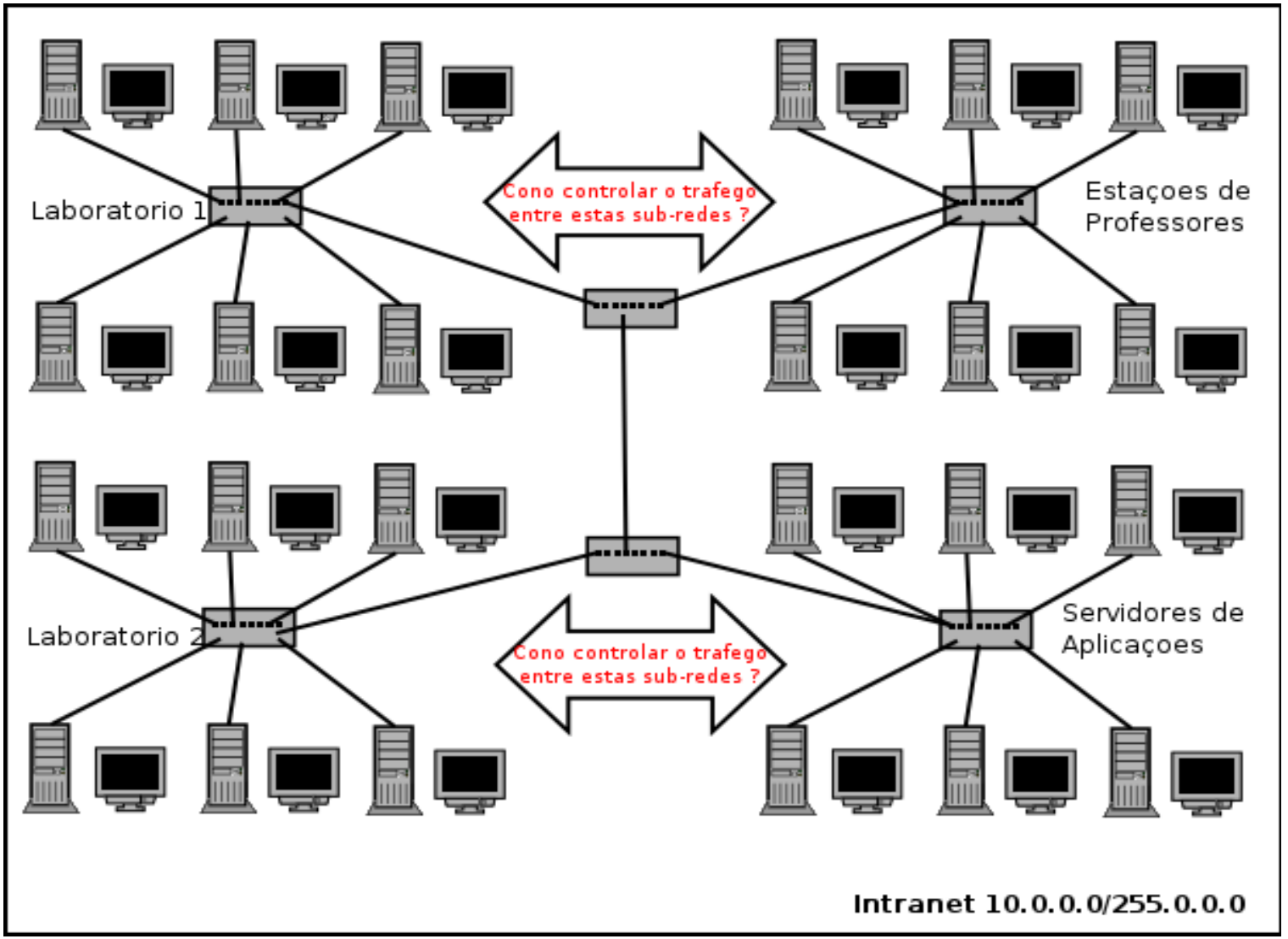
Rede Lógica Única
Ponto de Controle de Tráfego (Filtragem)

- A filtragem entre redes física e logicamente isoladas é possível
  - Intranet
  - Extranet
  - Internet
- E o controle de tráfego dentro da Intranet?
  - Ataques internos
  - Bloqueio de compartilhamentos
  - Controle de "alastramento" de vírus na rede interna
  - Filtragem de portas específicas

Rede Lógica Única
Ponto de Controle de Tráfego (Filtragem)



Rede Lógica Única
Ataques Internos e Outros Problemas



Rede Lógica Única
Ataques Internos e Outros Problemas

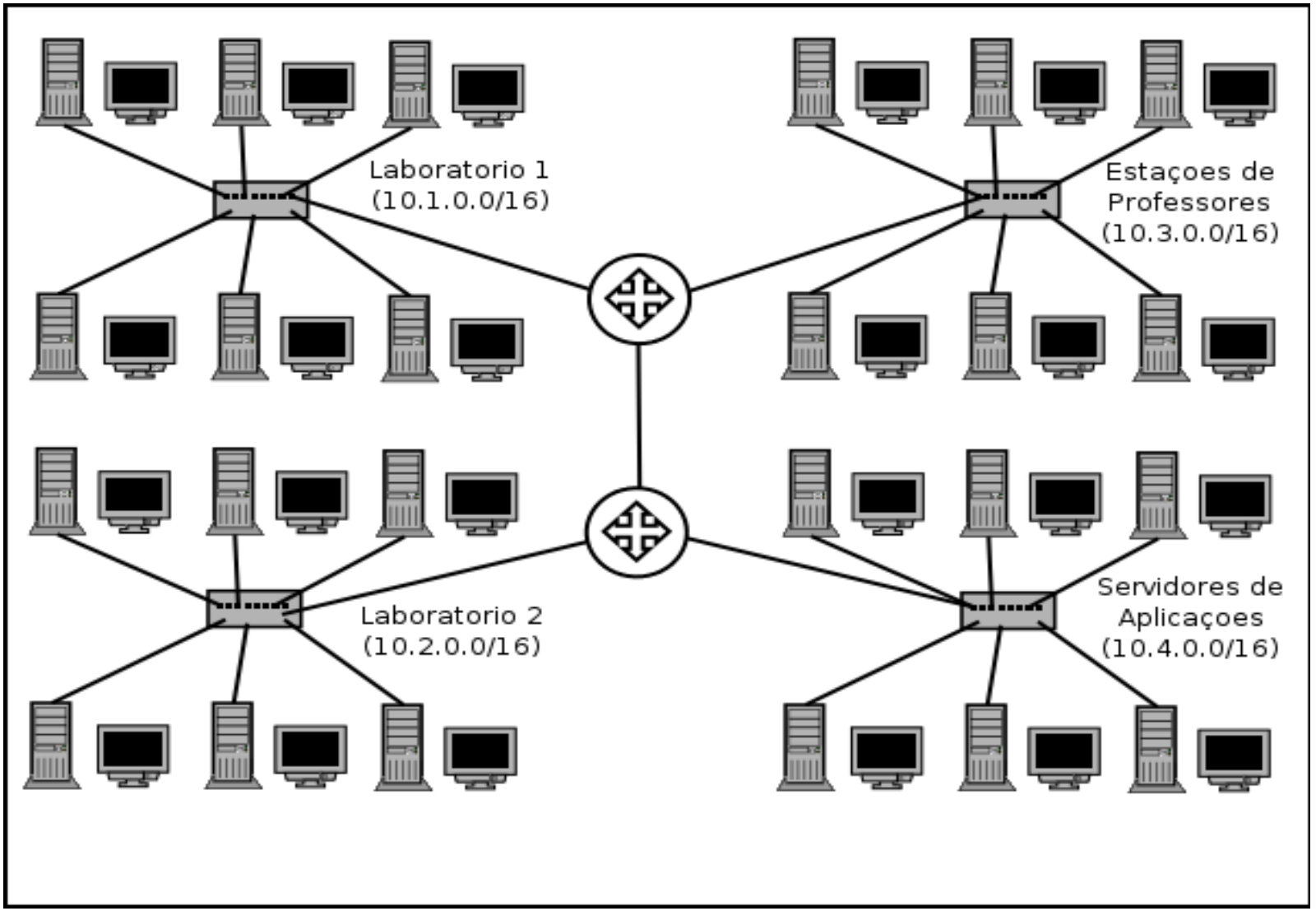
- **Hubs/Switches não realizam controle de tráfego**
- **A identificação até poderia ser possível com a utilização de IDS**

  (porém, de forma passiva)

Conclusão

- **Não existem elementos de controle de tráfego entre sub-redes de uma mesma rede lógica (!!??)**

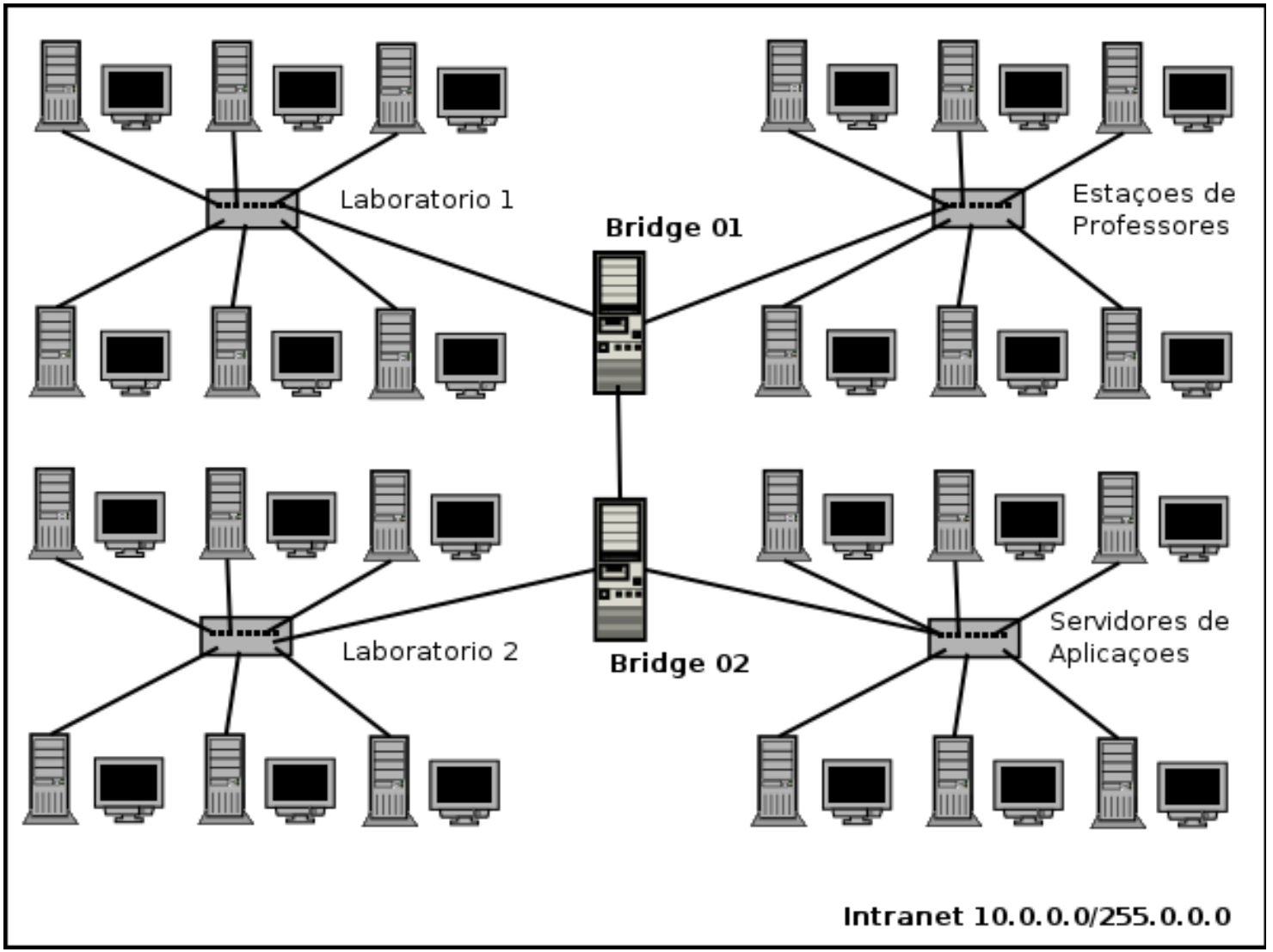
Utilizando Roteadores
Refazendo o Projeto Lógico



Utilizando Roteadores
Refazendo o Projeto Lógico

- **Vantagens Imediatas:**
  - Possibilidade de filtragens entre sub-redes
  - Diminuição do tráfego em broadcast
- **Impactos:**
  - Mudanças no projeto lógico
    - → Reconfiguração da rede (e das estações)
    - → Impacto nos usuários
  - Substituição de hubs/switches (entre sub-redes) por roteadores

Bridges Isolando Sub-redes em uma Rede Lógica Única



Bridges Isolando Sub-redes em uma Rede Lógica Única

- **Vantagens Imediatas:**
  - Possibilidade de filtragens entre sub-redes
  - Diminuição do tráfego em broadcast
  - Mesmo projeto lógico (não altera endereçamento)
  - Transparência para o usuário
- **Impactos:**
  - Substituição de hubs/switches (entre sub-redes) por bridges

    (OBS: Neste caso, as bridges são computadores com Linux fazendo esta função)

Montagem de uma Bridge Linux

Funcionamento

- Todo o tráfego entre as sub-redes atravessa fisicamente uma bridge;
- A bridge captura pacotes em nível de enlace, porém qualquer tipo de filtragem pode ser realizado antes do repasse (até mesmo em nível de aplicação);
- Possibilidade de:
  - Filtragem por origem ou por destino;
  - Filtragem por porta/serviço;
  - Filtragem por informações do payload (carga) de cada pacote

Montagem de uma Bridge Linux

- Para a configuração de uma bridge utilizando GNU/Linux é necessário que:
  - o kernel tenha o suporte a bridge habilitado;
  - o aplicativo `brctl` esteja instalado.
- As versões mais recentes do Debian GNU/Linux já trazem suporte nativo no kernel à implementação de bridges;
- Para que o aplicativo `brctl` esteja disponível para a configuração basta somente que o pacote `bridge-utils` esteja instalado

  (o pacote para distribuições baseadas no Linux RedHat tem o mesmo nome)

Montagem de uma Bridge Linux

- Uma vez preparada a máquina é necessário que todas as interfaces de rede da máquina utilizadas na bridge sejam ativadas sem endereçamento IP:

```
ifconfig eth0 0.0.0.0
ifconfig eth1 0.0.0.0
ifconfig eth2 0.0.0.0
...
```

- Atribui-se uma denominação lógica à bridge (`br0`):

```
brctl addbr br0
```

Montagem de uma Bridge Linux

- Em seguida devem ser adicionadas à bridge `br0` todas as interfaces que serão utilizadas para interconectar as porções de rede:

```
brctl addif br0 eth0
brctl addif br0 eth1
brctl addif br0 eth2
...
```

Montagem de uma Bridge Linux

Se esta intranet é interligada a outras redes e/ou à Internet, deve-se adicionar a rota padrão (default) apontando o roteador que encaminhará pacotes desta rede para máquinas fora de seu domínio:

```
route add default gw [endereço ip do gateway]
```

Montagem de uma Bridge Linux

Uma bridge não necessita de nenhum endereço IP associado;

Porém, se for necessária a administração remota da própria bridge, pode-se atribuir-lhe um endereço IP da rede lógica a que pertence;

Este endereço poderá ser acessado a partir de qualquer porção de rede ligada fisicamente à bridge (a menos que sejam inseridas regras de bloqueio de acesso nos filtros da bridge).

```
ifconfig br0 [endereço IP] netmask [máscara da rede] up
```

Montagem de uma Bridge Linux
Resumo (Exemplo)

```
ifconfig eth0 0.0.0.0 promisc up
ifconfig eth1 0.0.0.0 promisc up
ifconfig eth2 0.0.0.0 promisc up

brctl addbr br0
brctl addif br0 eth0
brctl addif br0 eth1
brctl addif br0 eth2

ifconfig br0 10.1.1.2 netmask 255.255.0.0 up

route add default gw 10.1.1.1 dev br0
```

Montagem de uma Bridge Linux
Resumo (Exemplo)

Usando o /etc/network/interfaces (Debian):

```
auto br0
iface br0 inet static
   address 10.1.1.2
   netmask 255.255.0.0
   network 10.1.0.0
   broadcast 10.1.255.255
   gateway 10.1.1.1
   bridge_ports eth0 eth1 eth2
```

```
# ifdown -a
# ifup br0
```

Filtragem de Pacotes na Bridge

- A Bridge é uma solução que captura/repassa pacotes em nível 2 (OSI)
- Porém, uma vez capturado o pacote, pode-se utilizar:
  - Netfilter/Iptables
  - Ebtables
- ... para a filtragem em TODAS as camadas (inclusive aplicação)...
- ... antes de proceder o repasse (ou bloqueio) dos dados que cruzam a bridge.

Aumentando o Nível de Controle

Conclusão

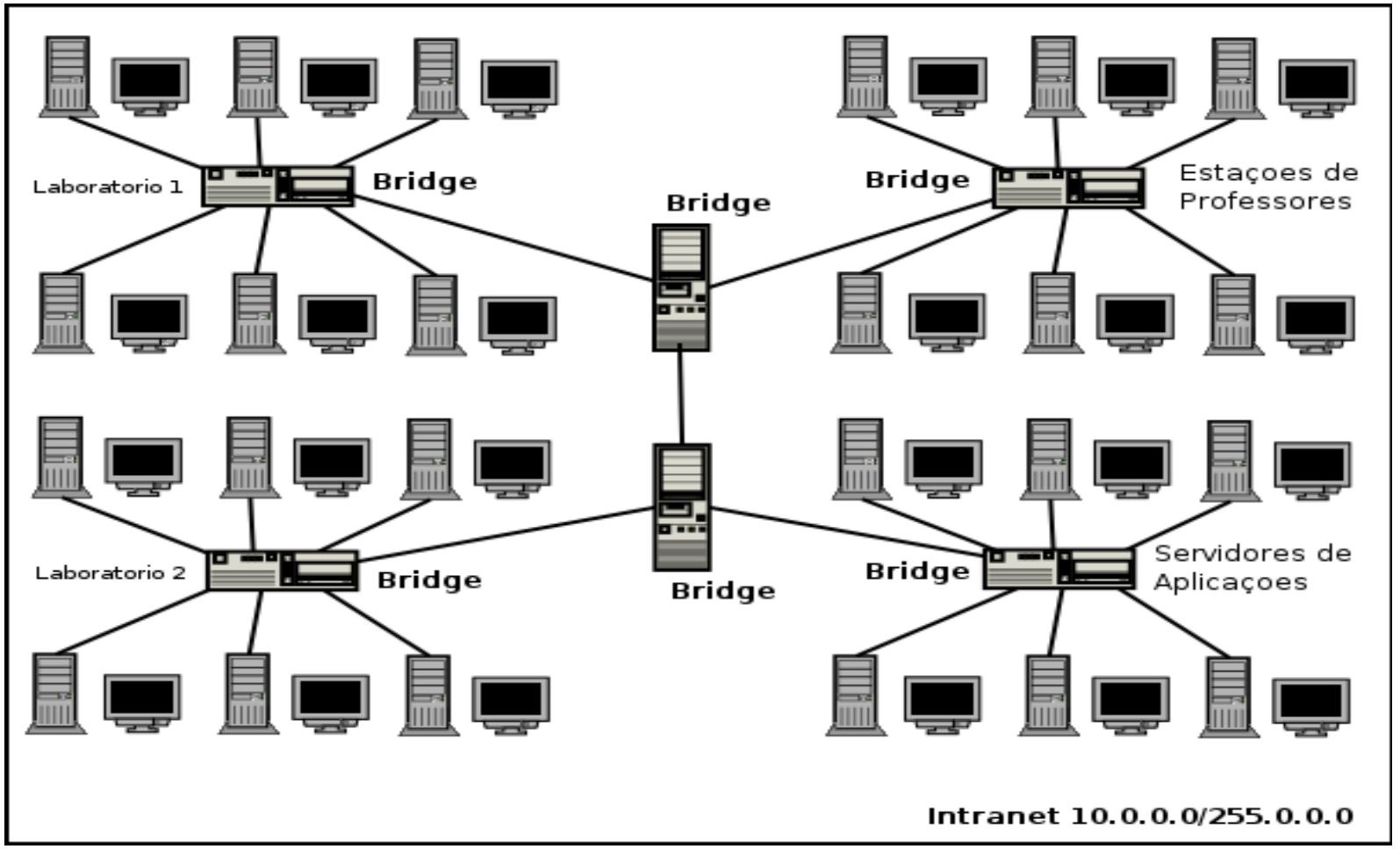
- Com bridges é possível filtrar tráfego entre sub-redes físicas em uma mesma rede lógica

Novos Desafios

- E se trocarmos os hubs/switches que ligam as estações por bridges, poderemos filtrar o tráfego entre estações de uma mesma rede lógica?
- Isso é implementável?
- A implementação é viável?

Aumentando o Nível de Controle

Idéia Inicial



Uso de Bridges para Controle de Tráfego entre Máquinas

Requisitos Físicos

- 01 (uma) placa de rede para cada máquina ligada

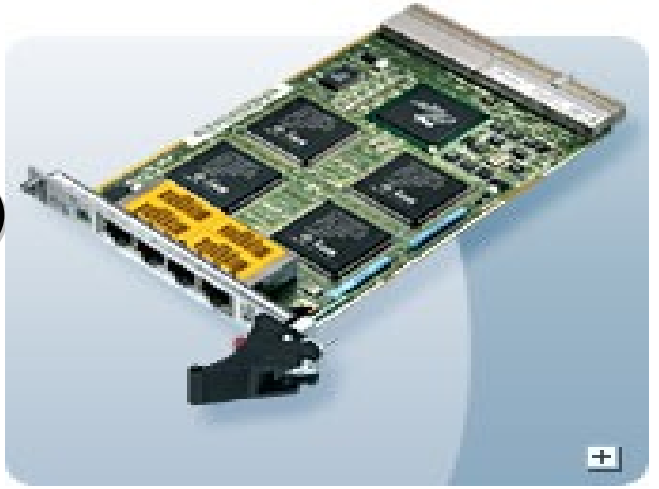
Constatação Inicial

- A troca de um hub/switch de 24 portas por uma bridge Linux implica em:
  - Necessidade de uma máquina com 24 placas de rede !!??

Uso de Bridges para Controle de Tráfego entre Máquinas

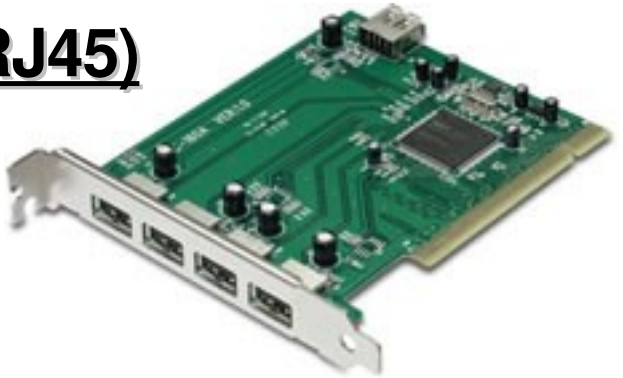
Solução 01

- Uso de placas Quadfast (4 portas por placa)
- Mínimo de 6 slots PCI
  - 6 slots x 4 portas = 24 portas
- Solução fisicamente possível mas...
  - Placas com 6 slots PCI não são comuns
    - ⇒ Custo elevado
  - Placas Quadfast também não
    - ⇒ Custo elevado



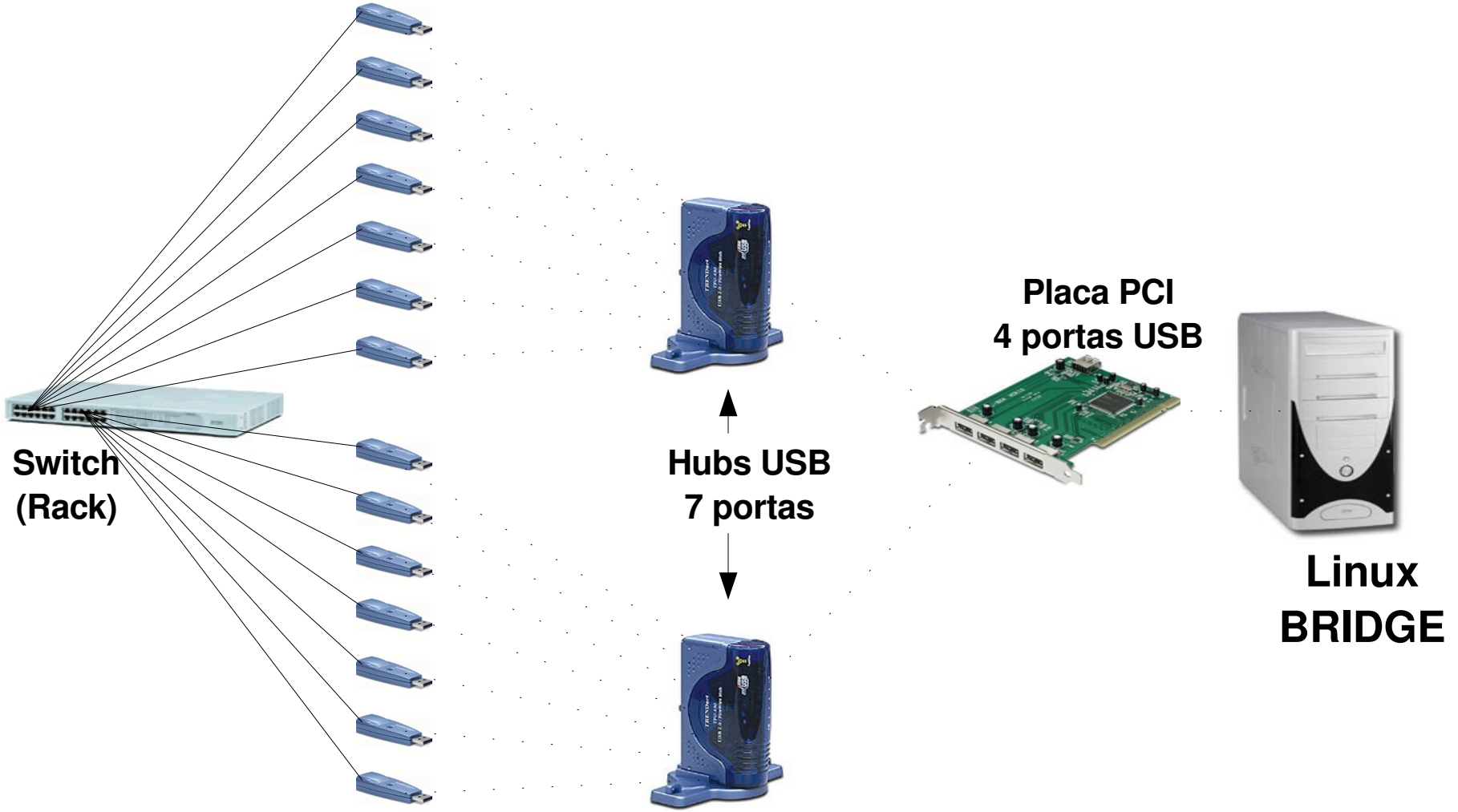
Uso de Bridges para Controle de Tráfego entre Máquinas

Solução 02 :: (Uso de dispositivos USB/RJ45)



- 01 Placa Extensora USB (1 x 4)
- 04 Hubs USB com 07 portas cada
  - Total ⇒ 28 portas USB disponíveis
- 28 Adaptadores USB/RJ45

Uso de Bridges para Controle de Tráfego entre Máquinas



Uso de Bridges para Controle de Tráfego entre Máquinas

Questões Adicionais para Esta Solução

- **Limitação de velocidade USB**
- **Muitos pontos de falha**

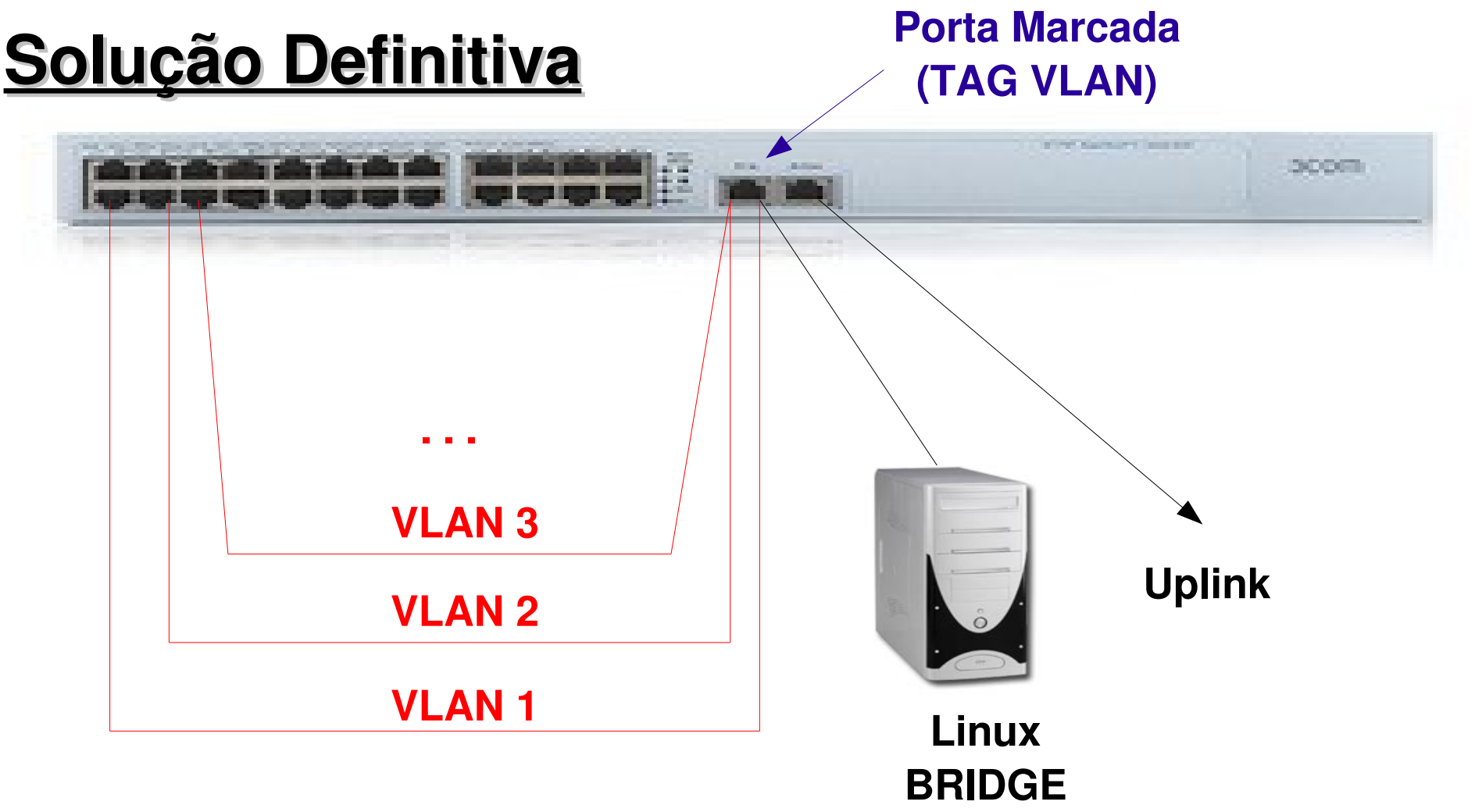
Uso de Bridges para Controle de Tráfego entre Máquinas

Solução Definitiva

- Switch 24 portas (10/100) + 2 portas (10/100/1000) [3Com 4226T]
- 24 Estações ligadas às portas do switch (10/100)
- Uplink ligado a uma das portas do switch (10/100/1000)
- Bridge Linux ligada em uma das portas do switch (10/100/1000)
  - 25 VLANs, sendo:
    - 1 VLAN (de duas portas) configurada para cada porta + porta bridge
    - 1 VLAN (de duas portas) configurada para ligação bridge + Uplink

Uso de Bridges para Controle de Tráfego entre Máquinas

Solução Definitiva



Uso de Bridges para Controle de Tráfego entre Máquinas

Configurações

- A bridge deve estar ligada em uma das portas mais velozes do switch;
  - Configuração desta porta (no switch);
    - Habilitar TRUNK
    - Habilitar TAGs 802.1q
- Configuração (switch) das demais portas:
  - Criar uma VLAN para cada par (porta_estação/porta_bridge)
  - Importante: Não habilitar TAGSs 802.1q nestas portas (apenas na da bridge)

Uso de Bridges para Controle de Tráfego entre Máquinas

Configuração da Bridge

- Pacotes instalados:
  - `bridge-utils`
  - `vlan`
- Arquivo /etc/network/interfaces (Debian):

```
iface eth0 inet static
        address 0.0.0.0
        netmask 0.0.0.0
        vlan_raw_device eth0
iface vlan2 inet static
        address 0.0.0.0
        netmask 0.0.0.0
        vlan_raw_device eth0
iface vlan3 inet static
        address 0.0.0.0
        netmask 0.0.0.0
        vlan_raw_device eth0
```

Uso de Bridges para Controle de Tráfego entre Máquinas

- **Reinicializar a rede:**

```
/etc/init.d/networking restart
```

- **Criar bridge:**

```
brctl addbr br0
```

- **Adicionar vlans na bridge:**

```
brctl addif br0 vlan2
brctl addif br0 vlan3
```

- **Atribuir um endereço à bridge:**

```
ifconfig br0 [endereço IP] netmask [máscara da rede] up
```

Uso de Bridges para Controle de Tráfego entre Máquinas

Funcionamento

- Todo o tráfego entre estações ligadas às portas do switch necessita passar na bridge antes de ser repassado (ou filtrado);
- Todo o tráfego entre estações é passível de filtragem em todas as camadas:
  - Bloqueio de compartilhamentos netbios;
  - Bloqueio de acesso a serviços não autorizados;
  - Bloqueio total de acesso entre máquinas (sem desconexão física).

Conclusões

- Facilidade de instalação e configuração
- Maior nível de controle (entre redes e/ou entre máquinas)
- Cooperação com Firewalls de Rede (filtragens específicas)
- Transparência ao usuário (solução mantém projeto lógico)
- Custo/benefício

(Solução pode ser implementada totalmente utilizando Softwares Livres)

Casos de Uso

- **Acadêmico**
  - Controle de laboratórios
  - Segmentação física de sub-redes (acadêmica x administrativa)
- **Comercial**
  - Detecção de tráfego malicioso entre setores da empresa
  - Isolamento e filtragem de setores e/ou máquinas da empresa

O ebtables

- `ebtables [-t table ] -[ADI] chain regra [extensão] target`
- `ebtables [-t table ] -P chain ACCEPT | DROP`
- `ebtables [-t table ] -F [chain]`
- `ebtables [-t table ] -L [chain]`
- `ebtables [-t table ] -E chain-antiga nova-chain`

O ebtables

- **Tabelas (tables):**
  - **filter (Chains: INPUT, OUTPUT e FORWARD)**
  - **nat (Chains: PREROUTING, OUTPUT e POSTROUTING)**
  - **broute (Chain: BROUTE)**

O ebtables

- **Especificações para regras:**

  `-p (protocolo)`

  - **Pode estar em hexadecimal ou nominal:**

    `ARP = 0x0600`

    `IPV4 = 0x0800`

  `-i` **(interface física de entrada, p.ex. eth0)**

  `--logical-in` **(interface lógica de entrada, p.ex. br0)**

  `-o` **(interface física de saída)**

  `--logical-out` **(interface lógica de saída, p.ex. br0)**

O ebtables

- `-s` (Endereço MAC de Origem)
- `-d` (Endereço MAC de Destino)

Extensões MATCH:

- `arp` (manipulação de protocolo arp e/ou rarp)
- `ip`

  `--ip-source / --ip-destination`

  `--ip-source-port / --ip-destination-port`
- `vlan` (manipulação de protocolo 802.1Q)

`http://ebtables.sourceforge.net/ebtables-man.html`

Uso de Bridges Linux no Controle de Tráfego entre Sub-Redes em Uma Mesma Rede Lógica

Perguntas ??

Uso de Bridges Linux no Controle de Tráfego entre Sub-Redes em Uma Mesma Rede Lógica

Ricardo Kléber M. Galvão
(rk@ufrn.br)

http://naris.info.ufrn.br

Núcleo de Atendimento e
Resposta a Incidentes de Segurança