

Redes de Computadores – 2018

Terceiro Trabalho

Protocolo ARP

Prof. Ronaldo Alves Ferreira

1 Descrição do Trabalho

Neste trabalho, você implementará o protocolo de resolução de endereços ARP de acordo com a RFC que o define (RFC 826). O protocolo deve ser implementado como um daemon que responderá a requisições ARP (xarpd). Além disso, você também implementará programas auxiliares para manipulação e visualização dos dados da tabela ARP do daemon e para configuração de interfaces de rede, os detalhes desses programas estão descritos na Seção 2. Os programas auxiliares devem se comunicar com o daemon ARP via TCP no endereço 127.0.0.1 e porta 5050.

O daemon ARP deve rodar da seguinte maneira:

```
% xarpd <interface> [<interface> ...]
interface: nome de uma interface de rede a ser gerenciada pelo daemon.
```

Note que você deve especificar pelo menos o nome de uma interface de rede.
Exemplos:

```
% xarpd enp0s3
```

ou

```
% xarpd enp0s3 enp0s8
```

O seu daemon ARP deve manter estatísticas de pacotes e bytes recebidos e transmitidos em cada interface especificada.

2 Programas Auxiliares

Você deverá implementar um programa (xarp) que lê comandos do usuário e executa esses comandos. Para cada comando, você talvez precise armazenar novas informações em cada interface gerenciada pelo daemon ARP. A lista de comandos está definida abaixo.

- **xarp show**

mostra a tabela ARP, utilizando o seguinte formato:

Entrada	Endereço IP	Endereço Ethernet	TTL
0	200.129.192.101	00:27:0B:3F:8E:01	60

Cada entrada da tabela deve ter um tempo de vida associado. Quando esse tempo de vida expirar, a entrada deve ser removida da tabela.

- **xarp res EndereçoIP**

resolve EndereçoIP e mostra o endereço Ethernet associado ao endereço IP especificado. Você deve primeiramente consultar a tabela ARP do daemon e verificar se o endereço IP já está armazenado na tabela. Caso esteja, simplesmente mostre o endereço Ethernet associado. Caso contrário, o daemon deve construir um pacote ARP Request e o enviar para a rede. A função que implementa essa funcionalidade deve ficar bloqueada até que a resposta da rede retorne ou até que um tempo máximo de espera (*timeout*) seja atingido. Caso o tempo máximo de espera seja atingido, imprima uma mensagem de erro (Endereço IP desconhecido). Para implementar o bloqueio da thread que executa essa tradução, consulte o funcionamento da função *sem_timedwait*.

O resultado desse comando, em caso de sucesso, deve ser o seguinte:

(200.129.192.101, 00:27:0B:3F:8E:02, 60)

- **xarp add EndereçoIP EndereçoEthernet ttl**

adiciona na tabela ARP do daemon o mapeamento entre os endereços IP e Ethernet especificados. A nova entrada da tabela deve ter tempo de vida igual a ttl segundos. Caso o valor ttl seja igual a -1, o mapeamento será permanente, ou seja, a entrada terá tempo de vida infinito. Você deve verificar se já existe um mapeamento para o endereço IP especificado. Em caso positivo, substitua o mapeamento existente pelo novo mapeamento especificado.

- **xarp del EndereçoIP**

remove da tabela ARP o mapeamento para o endereço IP especificado.

- **xarp ttl**

define o tempo de duração das entradas ARP (ttl) que são descobertas dinamicamente. Antes da chamada deste comando, o tempo de duração das entradas deve ser igual a 60 segundos.

Você também deverá implementar um programa (xifconfig) que recebe comandos do usuário e altera configurações das interfaces de rede. A lista de comandos está definida abaixo.

- `xifconfig`

quando executado sem parâmetros, o comando deve mostrar informações sobre todas as interfaces de rede controladas pelo daemon.

```
eth0  Link encap:Ethernet  Endereço de HW 00:1e:4f:43:48:06
      inet end.: 200.129.207.50  Bcast:200.129.207.63  Masc:255.255.255.224
      UP MTU:1500
      RX packets:64763539 TX packets:146111148
      RX bytes:9474581484 (8.8 GiB)  TX bytes:202753664071 (188.8 GiB)
```

- `xifconfig <interface> <IP address> <IP Netmask>`

configura a interface com o endereço IP e máscara fornecidos.

- `xifconfig <interface> mtu size`

configura um novo valor de MTU para a interface. Caso não seja modificado, o valor padrão de MTU deve ser 1500 bytes.

3 Fontes para Consultas

A principal fonte de consulta é a RFC 826. Além disso, utilize as páginas de manual (man pages) das funções C para entender o funcionamento das funções utilizadas no programa disponibilizado. Algumas funções importantes são: `sem_wait`, `sem_post`, `sem_timedwait`, `pthread_create`, `sleep`, `signal`, `socket`, `recvfrom`, `sendto`.

4 Entrega do Trabalho

O prazo de entrega se encerra no dia **11 de novembro às 23h59min**.

Além do código documentado em C/C++, você deve entregar um relatório descrevendo o seu trabalho. Neste relatório, você deve incluir uma breve introdução, decisões de implementação, funcionalidades não implementadas, problemas enfrentados na implementação, etc. O relatório deve ser entregue em um arquivo PDF.

Para a submissão do trabalho, crie um diretório chamado **t3** e inclua o seu código fonte e o arquivo com o relatório. Submeta todo o diretório compactado com zip ou gzip via Moodle (não submeta arquivos .rar). Se o seu programa for composto de vários arquivos em C/C++, é recomendado que voce crie também um arquivo Makefile. Remova os arquivos temporários (.o, .bak, etc) antes da submissão.

O trabalho pode ser feito em grupo com no máximo dois alunos. Casos de plágio serão tratados com rigor.

5 Avaliação

Além da correção do programa, será feita uma entrevista. Na entrevista, o grupo deverá explicar o funcionamento do programa e responder a perguntas relativas ao projeto.