Atividade 4 - Máquina de estados Finitos do Protocolo TCP

Gabriel Hidasy Rezende

September 20, 2015

Objetivo: Estudar uma implementação real da máquina de estados finitos do protocolo TCP.

1 Questão 1:

Analise o arquivo `tcp_states.h` e verifique os estados disponiveis. Estão inclusos todos os estados descritos anteriormente?

Logo no inicio do arquivo temos um enum com os estados disponiveis, segue anexa:

```
enum { TCP_ESTABLISHED = 1,
       TCP_SYN_SENT,
TCP_SYN_RECV,
TCP_FIN_WAIT1,
TCP_FIN_WAIT2,
TCP_TIME_WAIT,
TCP_CLOSE,
TCP_CLOSE_WAIT,
TCP_LAST_ACK,
TCP_LISTEN,
TCP_CLOSING, /*Now a valid state */
TCP_MAX_STATES /* Leave at the end!*/
};
```

Obviamente não vemos o estado CLOSED, que já era descrito como fictício. Todos os outros estados estão presentes e ainda temos o estado CLOSE que não está na tabela, mas é equivalente ao closed logo após o fechamento (é o estado da estrutura logo antes da limpeza).

2 Questão 2:

Identifique três funções envolvidas no processo de fechamento de uma conexão TCP tanto do lado cliente como do lado servidor. Suponha que o fechamento é inicializado pelo cliente com o envio de uma mensagem FIN para o servidor após ter acabado a transmissão de todos os pacotes de uma sessão. Descreva o papel da função no processo.

No arquivo `tcp_output.c` temos a função `tcp_send_fin` que tem um nome auto-explicativo, ela é responsavel por enviar os pacotes FIN, ela também é responsavel por enviar os pacotes restantes no buffer, o cliente chamaria ela para iniciar o processo de finalizar a transmissão (`__tcp_push_pending_frames` com um FIN)

No arquivo `tcp_input.c` temos a função `tcp_fin` que é responsavel por tratar o recebimento de um FIN, limpar a memoria usada pela estutura sk e levar a maquina de estados para um dos estados de encerramento.

E temos em `tcp.c` a função `tcp_close` é usada para iniciar o processo de finalização de uma conexão.

3 Questão 3

Escolha 2 funções nos arquivos `tcp.c`, `tcp_input.c` ou `tcp_output.c` que trabalhem diretamente com o estado do TCP. Pelo menos uma das funções deve mudar o estado dentro da função. Crie um relatório explicando a implementação das funções escolhidas.

3.1 `tcp_set_state` do arquivo tcp.c

Como o proprio nome diz, essa função é reponsavel por efetivamente modificar o estado da maquina de estados do TCP, com algum tratamente (e coleta de estatisticas) dependendo das transições requeridas. Ela é estraturada como um grande switch do estado atual, uma implementação comum para maquinas de estado (por exemplo, uma mudança para CLOSE leva o socket a ser removido da tabela de sockets, enquanto uma mudança de ESTABILISHED para ESTABILISHED apenas contabiliza estatisticas.

3.2 `tcp_fin` do `tcp_input.c`

Essa função, já vista na questão 2, é responsavel por tratar o recebimento de pacotes com a flag FIN. Ela aje sobre a maquina de estados da seguinte forma:

- se estava `ESTABILISHED` ou `SYN_RECV` vai para `CLOSE_WAIT`
- se estava em `CLOSE_WAIT` ou `CLOSING` assume que é retransmissão do `FIN` e não faz nada
- se estava em `LAST_ACK` não faz nada de acordo com a RFC793
- se estava em `FIN_WAIT1` enviar um `ACK` pelo `FIN` e vai para `CLOSING`
- se estava em `FIN_WAIT2` enviar um `ACK` e entrar em `TIME_WAIT`.

Em seguida ela programa o shutdown do socket e termina