

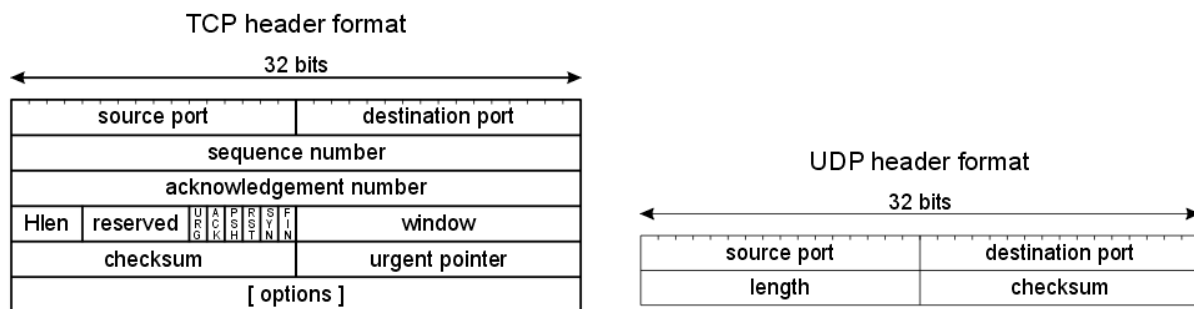
Redes I – TP - Implementação de pilha de protocolos

DNS

Jonathan Henrique, Rúbio Torres e Igor Miranda

Camada de Transporte

A implementação da camada de transporte foi feita em PHP, usando a biblioteca de socket. De acordo com a especificação dos protocolos, ela opera com cabeçalhos no seguinte formato:



A camada recebe uma mensagem da camada de Aplicação, e coloca o cabeçalho com todos os dados necessários. Caso o protocolo usado seja UDP, ela envia para a outra camada a mensagem inteira, e, caso o protocolo seja TCP, ela executa o Three-Way Handshake, monta o cabeçalho do protocolo, faz o controle de fluxo e envia para o destinatário a mensagem dividida em pacotes menores. Ao receber uma mensagem de uma camada inferior, ela faz o processo inverso e entrega à camada de aplicação.

Implementação

Loop principal do servidor

Ao executar o servidor, ele fica constantemente “escutando” por conexões. Quando uma máquina solicita uma conexão, ele aceita e espera o recebimento de dados. Ao receber os dados, a máquina verifica a origem para saber se os dados são provenientes da camada de Aplicação ou de uma camada inferior, e toma a ação apropriada.

Envio de mensagens – TCP

Quando a camada recebe um código da camada de Aplicação a ser transferida por TCP, ela, primeiro, inicia um Three-Way Handshake para iniciar a conexão com o destino, se essa conexão já não existia, e, depois, divide a mensagem em pacotes menores e os transmite. Se é solicitada uma retransmissão (3 ACKs consecutivos), é recomeçada a retransmissão a partir do ponto da última mensagem não confirmada pelo cliente. Além disso, o tamanho de cada mensagem é incrementado a cada mensagem recebida com sucesso, e retorna ao valor inicial (10 bytes) quando ocorre algum erro.

Envio de mensagens – UDP

Mensagens UDP são bem mais simplesmente tratadas, o código simplesmente insere os cabeçalhos e passa para a camada inferior.

Recebimento de mensagens – TCP

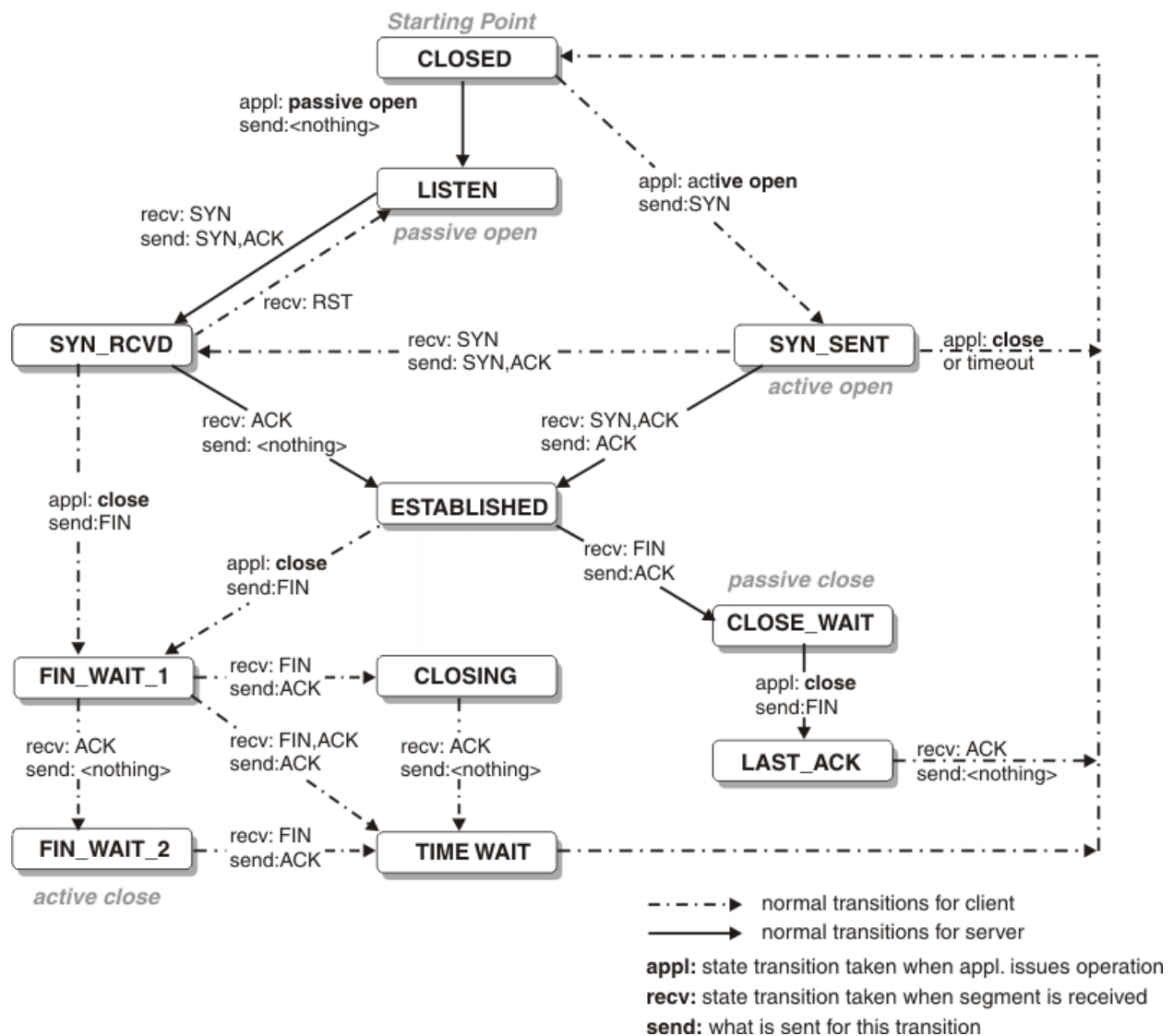
Ao receber segmentos TCP, considerando que já houve o Three-Way Handshake e a conexão já está estabelecida, a camada compara se ela esperava pelo número de sequência recebido, ou seja, se o segmento recebido é o próximo na mensagem que ela está recebendo, e, em caso afirmativo, ela adiciona o segmento à mensagem na memória. Em caso negativo, ela solicita retransmissão, enviando novamente o número de confirmação da mensagem anterior. Após a mensagem inteira ter sido recebida, a camada envia a mensagem completa à camada de Aplicação.

Recebimento de mensagens – UDP

Mensagens UDP recebidas pelo código tem o cabeçalho removido e, após isso, são repassadas para a camada de Aplicação.

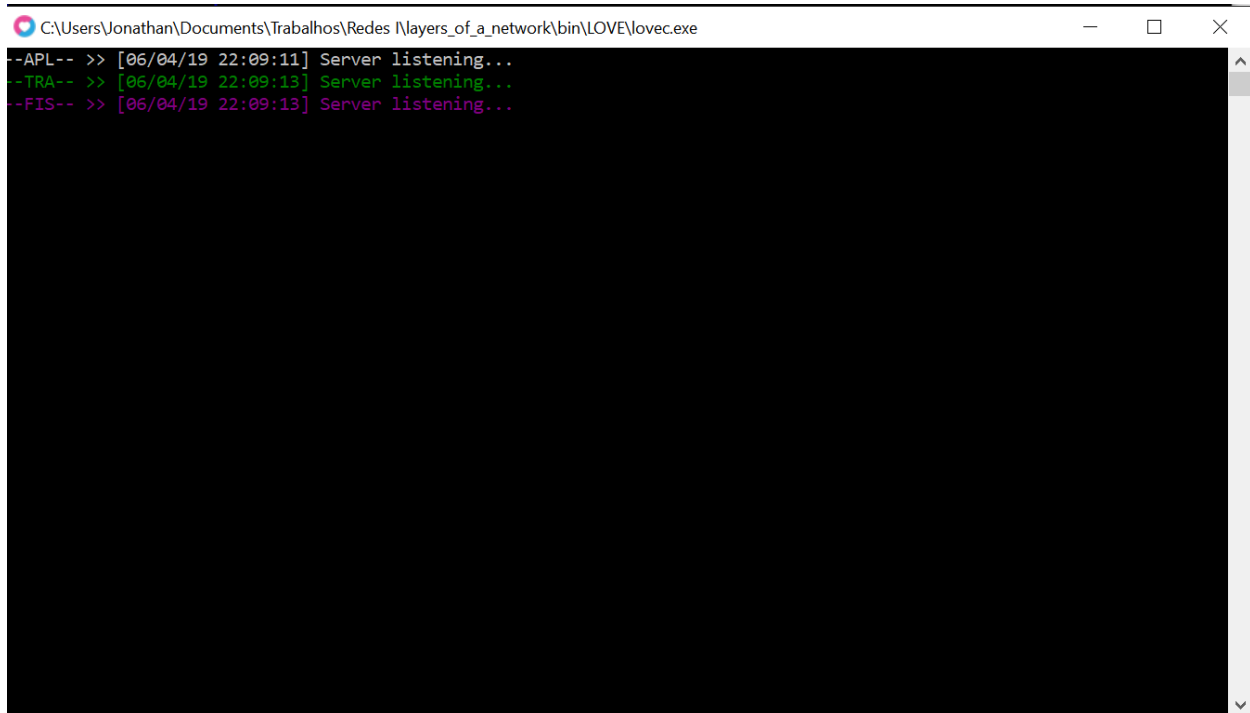
Three-Way Handshake

Ao iniciar ou finalizar uma conexão, o código faz um Three-Way Handshake quando está sendo usado o protocolo TCP. A cada passo desse processo, a camada assume um estado. Esse procedimento foi feito de acordo com o fluxograma a seguir:



Testes

Para testar o funcionamento da camada, ela foi integrada ao código já pronto das camadas de Aplicação e Física. Além disso, os logs de cada camada foram alterados para serem exibidos em cores diferentes, para facilitar a visualização dos logs.



```
C:\Users\Jonathan\Documents\Trabalhos\Redes I\layers_of_a_network\bin\LOVE\lovec.exe
--APL-- >> [06/04/19 22:09:11] Server listening...
--TRA-- >> [06/04/19 22:09:13] Server listening...
--FIS-- >> [06/04/19 22:09:13] Server listening...
```

Execução

Para executar o programa, simplesmente entre em um terminal, vá para a pasta “src” e execute “love ApplicationLayer” (é necessário o LOVE2D para executar o programa). Com a aplicação aberta, simplesmente digite uma consulta e pressione “Enter” para iniciar o envio de uma resposta DNS à requisição digitada. É possível clicar na caixa onde está sendo indicado o protocolo da camada de Transporte para alterá-lo entre TCP e UDP.

Considerações Finais

A camada foi implementada com um funcionamento que pode ter pequenas alterações para se adequar à implementação da camada de rede. Tudo foi implementado de acordo com a especificação.