

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS
CENTRO DE CIÊNCIAS EXATAS, AMBIENTAIS E TECNOLOGIAS
FACULDADE DE ENGENHARIA DE COMPUTAÇÃO

REDES DE COMPUTADORES B
PROF. EDMAR ROBERTO SANTANA DE REZENDE

Projeto de Redes
Camada de Enlace

Leonardo Bonetti Maffei
Murilo Silva Bergo
Murilo Silva Oliveira de Santana

CAMPINAS
2015

Sumário

Introdução	3
Especificação dos Protocolos	3
Plano de Implementação	4
Plano de integração e Teste	6
Compilação e Execução do Programa	6
Cronograma e Divisão de Responsabilidades	7
Considerações Finais	8
Problemas e Discursão	8
Referencias e Bibliografia	8

Introdução

O objetivo deste projeto é a construção da arquitetura de redes de computadores organizados em sistema de camadas, para a elaboração do projeto e facilitar o entendimento o dividiu-se em etapas, para esta primeira focou-se na implementação da camada de enlace e seus protocolos.

A Camada de Enlace tem a responsabilidade de transferir datagramas entre nós adjacentes através de um enlace. Nesta etapa do projeto foi passado um arquivo de configuração inicial que contem a especificação dos nós da rede virtual e os enlaces que deveram ser montados.

Especificação dos Protocolos

Não haverá bufferização na camada de enlace.

Para o envio simultâneo de quadros de mensagens será utilizado o mecanismo de threads.

Quando detectado um erro não o pacote será descartado, não havendo retransmissão de pacotes.

Para a detecção de erros será utilizado o checksum.

Plano de Implementação

Conforme sugerido pelo professor da matéria o projeto foi dividido em mais um arquivo extensão .c e neles feitos partes específicas do projeto desta camada.

topologia1.txt

Este é arquivo de configuração inicial e nele estão as especificações dos Nós e os Enlaces que a rede devesse possuir.

Na primeira parte do arquivo estão identificados em cada linha o número do nó, o IP e a Porta associados a este nó.

Na segunda parte estão identificados em cada linha a ligação dos nós, seguida do MTU daquele enlace. Através dessa ligação será feito o enlace, então se o nó 1 estiver ligado ao nó 2 se solicitado deve ser possível enviar uma do nó 1 para o nó 2

Enlace.c

Este arquivo é onde se projeta a Camada de Enlace. Nela utilizaremos as funções: sendPhy (Responsável pelo envio de dados), receivePhy(Responsável pelo recebimento de dados), TopologyLink (Responsável por ler o arquivo de texto para montar a topologia de rede que será utilizada), checksNeighbors(Responsável por checar os vizinhos), showResults(Mostra o resultado da leitura do arquivo), startLink(Inicia o link), analyzeMTU(Checa se o pacote é maior que o MTU) e readingFile(Executa em sequencia a leitura e exibição dos dados e do arquivo).

Teste.c

Responsável por realizar os testes de envio e recebimento. Suas funções são: sendNetwork(Envia a mensagem na rede, e se gerar erros explica quem erro está acontecendo) , receiveNetworkLink(Recebe a mensagem da rede) e initnetwork(Inicia a conexão).

Main.c

O Main.c é responsável por executar todas as funções e coordená-las a fim de não gerar concorrência. Para a concorrência não ocorrer utilizamos a thread de mutex. Seus principais pontos são:

Cria os mutex para os buffers de envio e recebimento

```
if(pthread_mutex_init(&mutex[0], NULL)!=0)
{
    printf("Erro Mutex\n");
    return (-1);
}
```

Inicia teste do enlace

```
pthread_create(&layerNetwork, NULL, layerinNetwork, (void *)1);
usleep(1000);
```

Função que executa a camada de enlace

```
pthread_create(&layerLink, NULL, layerinLink, (void *)argv[1]);

pthread_join(layerLink, NULL);
pthread_join(layerNetwork, NULL);
```

Finaliza os mutex

```
if(pthread_mutex_destroy(&mutex[0])!=0)
{
    printf("Erro destroy\n");
    return (-1);
}
```

Library.h

É onde a biblioteca dos arquivos é criada.

Garbler.h

É a biblioteca do garbler.c

Garbler.c

Para checar se todos os erros estão sendo verificados e retornar pacotes se erros forem encontrados.

Plano de integração e Teste

Nesta primeira etapa do projeto haviam poucos teste a serem realizados, porém não foi possível dispor de tanto tempo para o a execução de testes diferente do planejado devido a problemas na implementação do projeto.

Os testes realizados para esta etapa do projeto são:

Verificação do enlace: Verificar se existe enlace entre os Nós, se existir um enlace entre o nó escolhido e o destino deve ser possível enviar um pacote do nó escolhido ao nó destino. Caso não exista um enlace entre os nós especificados será retornado uma mensagem de erro.

Verificação do MTU: Verificar o tamanho do pacote, se o pacote for maior que o MTU resulta em erro.

Verificar checksum: Verificar a integridade do pacote se não estiver completo a camada de enlace descarta.

Compilação e Execução do Programa

Para compilação deste projeto utilizou-se o recurso de objetos que permite compilar programas cujo o código está distribuído em mais de um arquivo. Para isto usou o comando `gcc -c NomeDoArquivo.c -o NomeDoArquivo.o` para todos os arquivos extensão .C usados no projeto gerando os objetos.

Depois chamar todos os programas objetos em conjunto usando o comando `gcc` chamando todos os arquivos extensão .o com o comando `gcc NomeDoArquivo1.o NomeDoArquivo2.o NomeDoArquivo3.o -o NomeDoArquivoMain`. Também será necessário acrescentar ao final o parâmetro `lthread` que do suporte as threads usadas no neste projeto.

Para a execução do projeto basta chamar o arquivo objeto principal compilado `./main` passando como parâmetro o arquivo de configuração inicial `.txt` e o Nós emulador.

Também é possível para executar todos os objetos criados o arquivo makefile onde é possível passar todo este processo de compilação nele para então simplesmente lê-lo e executá-lo diretamente.

Cronograma e Divisão de Responsabilidades

Cronograma detalhado de implementação

Semana (04/08)	Apresentação do Projeto e Análise
Semana (11/08)	Análise e Programação
Semana (18/08)	Programação
Semana (25/08)	Programação Teste e Elaboração do Relatório
Semana (01/09)	Finalização e Entrega da Camada de Enlace

A princípio houve problemas de entendimento e na implementação do projeto o que resultou em um atraso em algumas partes.

Divisão de tarefas entre os membros do grupo

Leonardo Bonetti Maffei	Análise, Programação, Teste e Elaboração do Relatório
Murilo Silva Bergo	Análise, Programação, Teste e Elaboração do Relatório
Murilo Silva Oliveira de Santana	Análise, Programação, Teste e Elaboração do Relatório

Problemas e Discursão

A princípio houveram problemas com a leitura do arquivo no qual o grupo perdeu mais tempo do que o planejado na configuração de como manipular os dados que o arquivo disponibilizava, problema que foi resolvido usando o comando fscanf pegando cada um dos dados necessário e colocando-os em variáveis.

Considerações Finais

Nesta primeira etapa da construção da arquitetura de camadas de redes implementamos a Camada de Enlace responsável por transmitir os datagramas da Camada de Redes entregando a um nó adjacente.

Entre as diversas funcionalidades da Camada de Enlace estão enquadramento de pacotes, acesso ao enlace, entrega confiável, controle de fluxo e detecção de erros, ainda é possível haver a correção erros, mas é necessário ter informações suficiente, porém usualmente ao encontrar um erro o quadro é descartado e solicitado a retransmissão deste.

Houveram problemas com o tempo necessário para a implementação desta etapa do projeto, problemas que serão levados em conta na próxima fase da implementação dispondo de um pouco mais de tempo à tarefa.

Referencias e Bibliografia

http://www2.ufpa.br/abelem/disciplinas/redes-ppgee/cap5_ppgee.pdf

<http://fig.if.usp.br/~esdobay/c/gcc.html>

KUROSE, J. F.; ROSS, K. W.; Redes de Computadores e a Internet. 3ª. ed. Pearson Education, 2005.