Guilherme Sividal - 09054512

Vitor Rodrigo Vezani - 10159861

**PUC-Campinas**

Faculdade de Engenharia de Computação - FECOMP

Centro de Ciências Exatas, Ambientais e de Tecnologias - CEATEC

**CAMADA DE TRANSPORTE**

Disciplina: Redes de Computadores II

Professor: Edmar Roberto Santana de Rezende

Campinas

2013

**Sumário**

1. **Introdução**........................................................................................................... 3

2. **Especificação dos protocolos**.............................................................................. 4

3. **Plano de implementação** ................................................................................... 9

4. **Plano de integração e teste**............................................................................... 14

5. **Compilação e execução do programa** ............................................................ 27

6. **Cronograma e divisão de responsabilidades** ................................................. 28

7. **Considerações finais** ........................................................................................ 30

8. **Referências e Bibliografia** ............................................................................... 31

**Introdução**

Neste projeto visamos simular o funcionamento de uma rede completa, implementando todas as suas camadas (Enlace, Rede, Transporte e Aplicação) e testar seu funcionamento como um todo.

Na terceira parte do projeto será implementada a camada de transporte, sendo esta, a camada responsável pela transferência eficiente, confiável e econômica dos dados entre a máquina de origem e a máquina de destino, independente do tipo, topologia ou configuração das redes físicas existentes entre elas, garantindo ainda que os dados cheguem sem erros e na sequência correta. A camada de transporte é uma camada fim-a-fim.

A função básica da camada de transporte é aceitar dados da camada de aplicação, dividi-los em unidades menores em caso de necessidade, passá-los para a camada de rede e garantir que todas essas unidades cheguem corretamente à outra extremidade.

Seu funcionamento: A camada de aplicação deseja enviar algum pacote para determinado nó e repassa para a camada de transporte, este pacote é segmentado e enviados por meio de um algoritmo de janela deslizante. A janela só deslizará à chegada de um ack.

Caso haja estouro do timer atrelado à janela. A camada de transporte reenvia os dados, garantindo assim, a confiabilidade e eficiência da transmissão.

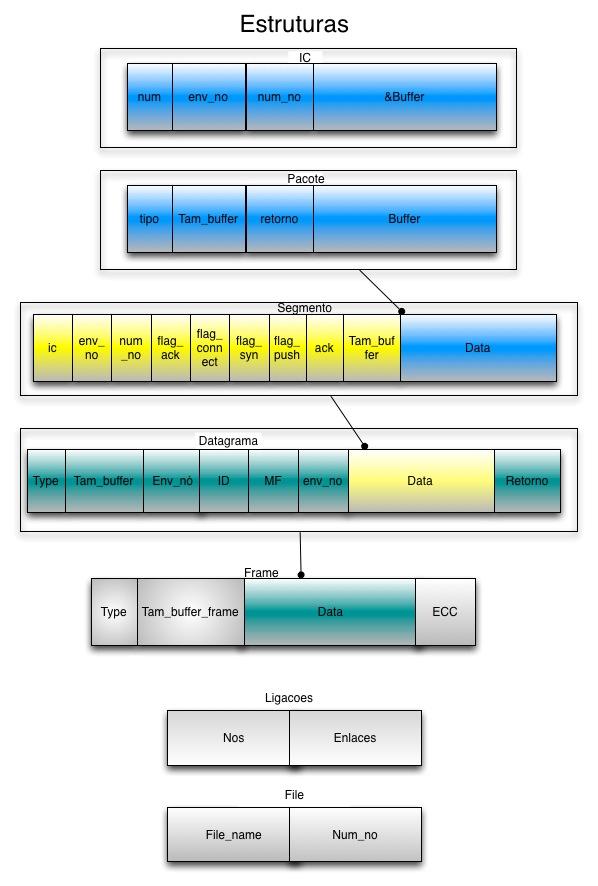
Os segmentos ao serem recebidos são colocados em ordem no buffer de recebimento e é enviado um segmento de ack contendo o próximo segmento esperado, os segmentos fora de ordem são descartados.

Implementado pelo Protocolo de Transporte Confiável (PTC), que terá as seguintes características:

* A semântica de comunicação será do tipo fluxo de bytes bidirecional confiável.
* Utilizará um algoritmo de janela deslizante para implementar controle de fluxo e confiabilidade.
* Utilizará reconhecimentos cumulativos*.*

**Especificação dos protocolos**

**Formato do frame, datagrama, segmento, pacote, ic.**



O segmento contem o ic, env\_no, num\_no, flag\_ack, flag\_connect, flag\_syn, flag\_push, ack, o tamanho do buffer efetivo e os dados que serão enviados que não são visíveis pela camada de rede e enlace.

Já o pacote contem o tipo, tam\_buffer (tamanho dos dados), retorno(endereço do buffer alocado pela camada de transporte) e os dados.

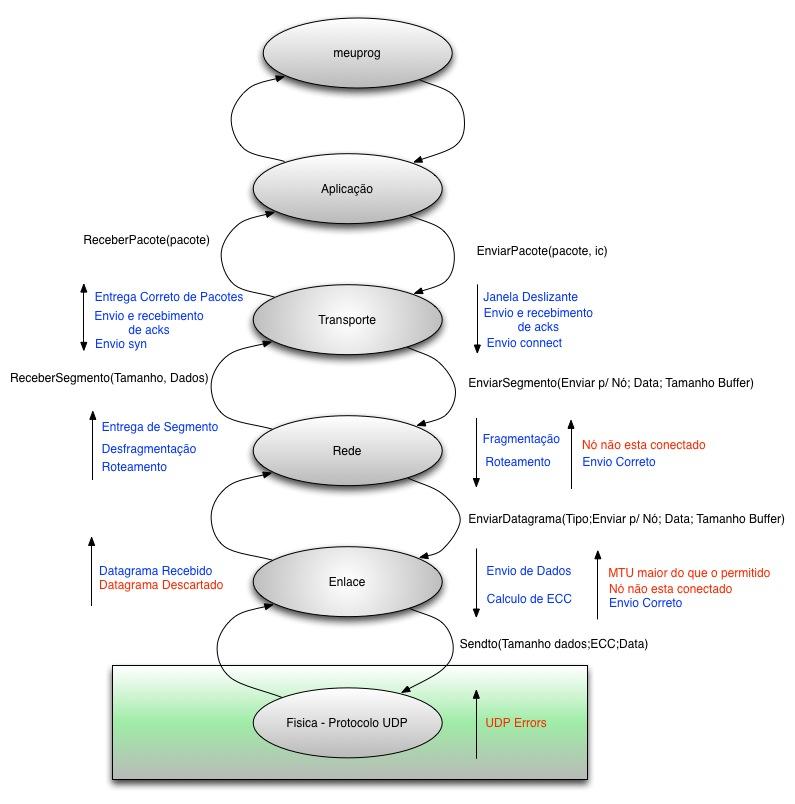
**Tabela de Visibilidade**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CAMADA**  **DADOS \** | **Enlace** | **Rede** | **Transporte** | **Aplicação** |
| **Frame** | **O** | **X** | **X** | **X** |
| **Datagrama** | **X** | **O** | **X** | **X** |
| **Segmento** | **X** | **X** | **O** | **X** |
| **Pacote** | **X** | **X** | **X** | **O** |

Legenda:

O - Visível  
X - Invisível

**Visão Sistêmica das Camadas**



Funcionamento da rede. Todas as camadas envolvidas e a camada que estamos implementando, a camada de transporte. Como podemos observar, ela recebe um pacote da camada de aplicação e o envia ao nó requisitado. A camada de transporte coloca o pacote no buffer, após utiliza um algoritmo de janela deslizante para enviar o pacote em pequenos segmentos e sempre validar o seu respectivo acknowledge(ack) para poder assim, andar com a base da janela e enviar os demais segmentos, até que todos os segmentos que compõe o pacote sejam entregues, o algoritmo de janela deslizante garante a confiabilidade e eficiência da transmissão.

Ao receber um segmento, a camada de transporte verifica se o numero de sequência do segmento é o numero de sequência esperado, se estiver correto, o segmento é alocado no buffer se não o segmento é descartado.

**Variáveis globais à todo programa**

buffer\_rede\_enlace\_env, buffer\_rede\_enlace\_rcv - buffers da camada de rede/enlace para enviar/receber

buffer\_trans\_rede\_env, buffer\_trans\_rede\_rcv - buffers da camada de trans/rede para enviar/receber

file file\_info - propriedades do arquivo

ligacoes ligacao - propriedades de nós e enlaces

mutex\_rede\_enlace\_env1, mutex\_rede\_enlace\_env2,mutex\_rede\_enlace\_env3 - Mutex da camada de rede/enlace para enviar

mutex\_rede\_enlace\_rcv1, mutex\_rede\_enlace\_rcv2,mutex\_rede\_enlace\_rcv3 - Mutex da camada de rede/enlace para receber

mutex\_rede\_rede\_atualizei1, mutex\_rede\_rede\_atualizei2 - Mutex da camada de rede

mutex\_rede\_rede\_receberotas2 - Mutex da camada de rede

mutex\_rede\_rede\_env1, mutex\_rede\_rede\_env2,mutex\_rede\_rede\_env3 - Mutex da camada de rede para enviar

mutex\_rede\_rede\_rcv1, mutex\_rede\_rede\_rcv2,mutex\_rede\_rede\_rcv3 - Mutex da camada de rede para receber

mutex\_trans\_rede\_env1, mutex\_trans\_rede\_env2,mutex\_trans\_rede\_env3 - Mutex da camada de transporte/rede para enviar

mutex\_trans\_rede\_rcv1, mutex\_trans\_rede\_rcv2,mutex\_trans\_rede\_rcv3 - Mutex da camada de transporte/rede para receber

mutex\_apli\_trans\_env1, mutex\_apli\_trans\_env2 - Mutex da camada de aplicacao/transporte para enviar

mutex\_apli\_trans\_rcv1, mutex\_apli\_trans\_rcv2 - Mutex da camada de transporte/rede para receber

env\_seg\_rcv\_seg\_timer\_2 - Mutex interno à camada de transporte

**Variáveis globais à camada de transporte**

int ack; // Numero do Ack

int syn; // Sync = 1

ic\_num; // Numero correspondente ao IC

buffer\_trans\_trans\_env, buffer\_trans\_trans\_rcv; buffers da camada de transporte para enviar/receber

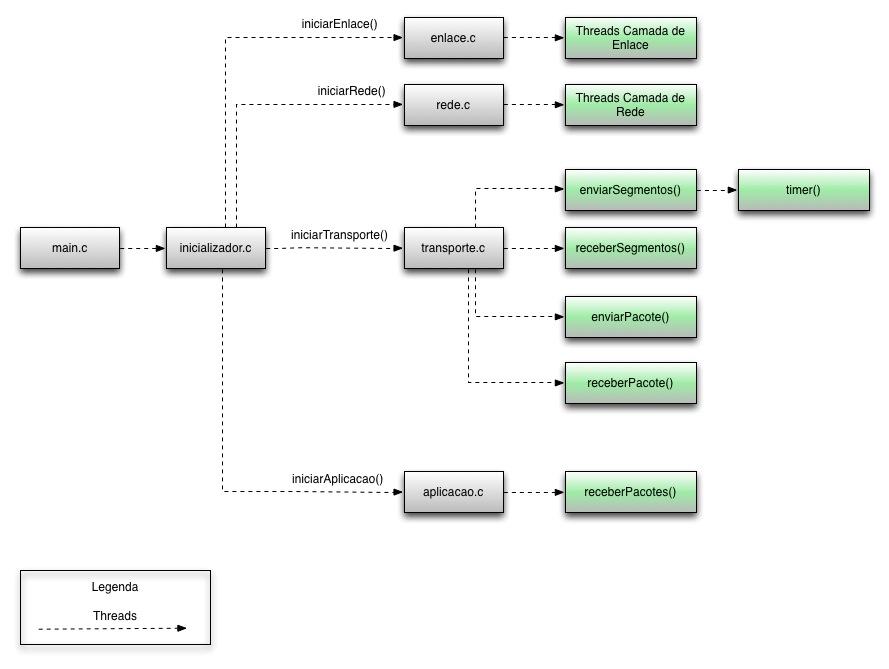
Lembrando que o buffer\_trans\_trans que está alocado é um ponteiro para o buffer que será alocado em tempo de execução após a conexão do IC.

**Plano de implementação**

O processo principal main tem a função de inicializar a thread inicializarCamadas() que por sua vez inicializa as variáveis, estruturas e definições globais. Também tem como função disparar as linhas de execução correspondente de cada camada. Cada camada, fora especificados para serem independentes entre si, instanciados novas linhas de execução correspondente às suas funções (iniciarTransporte() dispara threads relacionadas à camada de transporte), que por sua vez tem funções distintas.

Para a implantação da camada de enlace, rede, transporte e aplicação(teste) cada nó dispara 20 threads, como visto no diagrama abaixo:

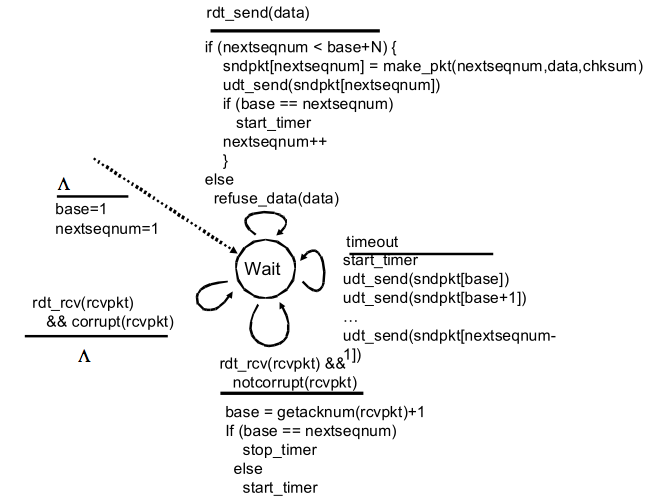
**Funções das Threads:**



**Camada de Rede**

* void iniciarTransporte(): Tem como função, iniciar as seguintes threads :
  + void \*enviarSegmentos();
  + void \*receberSegmentos();
  + void \*enviarPacote();
  + void \*receberPacote();
  + void \*timer();
* receberPacote : Essa thread tem como principal função receber pacotes da camada de aplicação e segue o fluxo:
  + retirarPacoteBufferApliTransEnv()
  + colocarPacoteBufferTransTransEnv()
* enviarSegmentos : Essa thread tem como principal função enviar o segmentos do buffer interno à camada de transporte para a camada de rede. É aonde está implementado o algoritmo da janela deslizante. Recebe também alem do sinal de estouro de timer, os syn e acks da thread receberSegmentos e segue o fluxo:
  + Cada envio de janela:
    - inicializar timer()
  + retirarSegmentoBufferTransTransEnv()
  + colocarSegmentoBufferTransRedeEnv()
* timer: Essa thread tem a função de ficar dormente por um tempo e caso ela eventualmente acorde, terá a função de desbloquear a thread enviarSegmentos() fazendo que ela reenvie toda a janela.

Algoritmo da janela deslizante:



* receberSegmentos : Essa thread tem como principal função receber segmento da camada de rede e segue o fluxo:
  + retirarSegmentoBufferTransRedeRcv()
  + Se for um datagrama com connect
    - Aloca buffer
    - Enviar syn
  + Se for um datagrama com push
    - Numero de sequência esperado igual o recebido
      * Colocar segmento no buffer
      * Incremente numero de sequencia esperado
      * Monta e envia ack
    - Senão
      * descarta datagrama
  + Se for um datagrama com ack
    - desbloqueia a thread enviarSegmentos para que possa andar com a base
  + Se for um datagrama com syn
    - desbloqueia a thread enviarSegmentos para que possa estabelecer conexão
* enviarPacote: Essa thread tem como principal função enviar pacotes para a camada de aplicação e segue o fluxo:
  + retirarPacoteBufferTransTransRcv()
  + colocarPacoteBufferApliTransRcv()

**Camada de Aplicação**

* receberPacotes(): Essa thread recebe e mostra o conteúdo do pacote recebido pela camada de transporte:
  + retirarPacotesBufferApliTransRcv
  + printar(pacote)

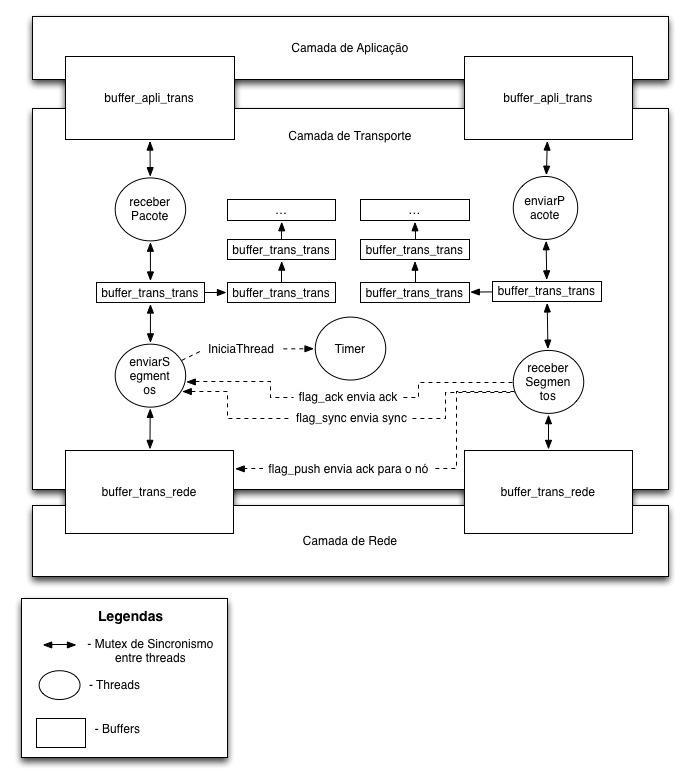
**Funções da camada de aplicação:**

A camada de aplicação fornece ao programa algumas funções que podem ser utilizadas:

* aps: define um ponto de serviço no nó local. O comando retorna um Identificador de Ponto de Serviço (IPS) único para cada ponto de serviço no nó. Outros nós podem conectar-se a IPS e descarregar arquivos a partir desse ponto de serviço.
* fps: requisita o fechamento do ponto de serviço IPS no nó local. Todas as conexões abertas em IPS serão canceladas.
* conectar: requisita o estabelecimento de uma conexão entre o nó local e o ponto de serviço IPS no nó NóDest. O comando retorna um Identificador de Conexão (IC) único para cada conexão aberta no nó local.
* desconectar: fecha a conexão com identificador IC.

baixar: solicita a transferência de um arquivo remoto com o nome Arq através da conexão IC.

**Relação entre Threads, Mutex e buffers**

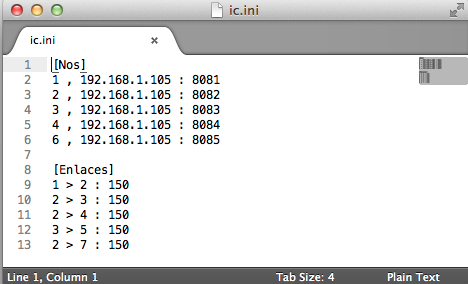


**Plano de integração e teste**

Para a validação dos requisitos do projeto foram estabelecidos e realizados os seguintes casos de testes.

**1. Enviar do nó 1 para o nó 3 utilizando janela deslizante, fragmentação, roteamento, perda de segmento, corrompimento de segmento e duplicação de segmentos.**

**Conteúdo do arquivo de configuração:**



**Parâmetros:**

**TAM\_SEGMENT 50**

**TAM\_JANELA TAM\_SEGMENT \* 4**

**TAM\_BUFFER\_TRANS 6000**

**set\_garbler**(3, 3, 3);

**Tamanho dos dados:** 292

**Programa principal:**

**ps** = aps();

if (ps == -1)

printf("Impossivel criar ps\n");

else

printf("ps criado '%d'\n", ps);

**ic** = conectar(env\_no, ps);

if(ic.env\_no == -1)

printf("Erro ao conectar, verifique a existencia do ps\n");

**baixar**(ic, "O núcleo Linux foi desenvolvido pelo programador finlandês Linus Torvalds, inspirado no sistema Minix. O seu código fonte está disponível sob a licença GPL (versão 2) para que qualquer um o possa utilizar, estudar, modificar e distribuir livremente de acordo com os termos da licença.");

**desconectar**(ic);

**Resultado dos Programas:**

**Nó 1:**

MacBook-Pro-de-Vitor-Vezani:Camada de Transporte macbookpro$ ./meuprog.exe ic.ini 1

sh-3.2# ./meuprog.exe ic.ini 1

Nome do arquivo: 'ic.ini'

Numero nó: '1'

[REDE - ROT]Enviei Tabela de Rotas para o nó '2'

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - ROT]Recebi tabela de rotas do nó '2'

ps criado '0'

**[APLIC - CON]Recebi pedido para conectar no no : '3', ps '0'**

----------------------------------------------------------------------

**[APLIC - RET]Alocado o buffer -> end\_buffer: '0x7f8e91e00000'**

[TRANS - ENV] Vou enviar a Janela

**Esperando syn para estabelecer conexão!**

[TRANS - ENV] base: '0', nextseqnum: '0'

[TRANS - TIMER] Iniciei a Thread Timer!!

[TRANS - ENV] Mandei um Segmento

[TRANS - ENV] base: '0', nextseqnum: '50'

[TRANS - ENV] Mandei um Segmento

[TRANS - ENV] base: '0', nextseqnum: '100'

[REDE - ENV] Não há ligacao do nó: '3'!

[TRANS - ENV] Mandei um Segmento

[TRANS - ENV] base: '0', nextseqnum: '150'

[TRANS - ENV] Mandei um Segmento

[REDE - ENV] Não há ligacao do nó: '3'!

[REDE - ENV] Não há ligacao do nó: '3'!

[REDE - ENV] Não há ligacao do nó: '3'!

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

**[TRANS - ENV] Recebi um pacote de syn**

Conexão estabelecida com sucesso!

**[TRANS - ENV FIM DA JANELA] sizeof(buffer): '40', base final: '0', nextseqnum final: '200'**

**[APLIC - BAIXAR] Data: 'O núcleo Linux foi desenvolvido pelo programador finlandês Linus Torvalds, inspirado no sistema Minix. O seu código fonte está disponível sob a licença GPL (versão 2) para que qualquer um o possa utilizar, estudar, modificar e distribuir livremente de acordo com os termos da licença.'**

**[APLIC - BAIXAR] Tam\_buffer: '292'**

----------------------------------------------------------------------

**[TRANS - ENV] Vou enviar a Janela**

**[TRANS - ENV] base: '0', nextseqnum: '0'**

**[TRANS - ENV] Mandei um Segmento**

**[TRANS - TIMER] Iniciei a Thread Timer!!**

**[TRANS - ENV] base: '0', nextseqnum: '50'**

**[TRANS - ENV] Mandei um Segmento**

**[TRANS - ENV] base: '0', nextseqnum: '100'**

**[REDE - ENV] Não há ligacao do nó: '3'!**

[TRANS - ENV] Mandei um Segmento

[TRANS - ENV] base: '0', nextseqnum: '150'

**[ENLACE] ERRO de MTU Tamanho: '292', MTU: '150'**

**[TRANS - ENV] Mandei um Segmento**

**[REDE - ENV] MTU Excedido MAX '150'**

**[REDE - FRAG] MTU excedido dividindo o pacote em max '150' bytes**

**[REDE - ENV] Não há ligacao do nó: '3'!**

[ENLACE] ERRO de MTU Tamanho: '292', MTU: '150'

[REDE - ENV] MTU Excedido MAX '150'

[REDE - FRAG] MTU excedido dividindo o pacote em max '150' bytes

[REDE - ENV] Não há ligacao do nó: '3'!

[ENLACE] ERRO de MTU Tamanho: '292', MTU: '150'

[REDE - ENV] MTU Excedido MAX '150'

[REDE - FRAG] MTU excedido dividindo o pacote em max '150' bytes

[REDE - ENV] Não há ligacao do nó: '3'!

[ENLACE] ERRO de MTU Tamanho: '292', MTU: '150'

[REDE - ENV] MTU Excedido MAX '150'

[REDE - FRAG] MTU excedido dividindo o pacote em max '150' bytes

[TRANS - ENV] Thread de timer estourou. Reenviando Janela

----------------------------------------------------------------------

[TRANS - ENV] Vou enviar a Janela

[TRANS - ENV] base: '0', nextseqnum: '0'

[TRANS - ENV] Mandei um Segmento

[TRANS - TIMER] Iniciei a Thread Timer!!

[TRANS - ENV] base: '0', nextseqnum: '50'

[TRANS - ENV] Mandei um Segmento

[TRANS - ENV] base: '0', nextseqnum: '100'

[REDE - ENV] Não há ligacao do nó: '3'!

[TRANS - ENV] Mandei um Segmento

[TRANS - ENV] base: '0', nextseqnum: '150'

[ENLACE] ERRO de MTU Tamanho: '292', MTU: '150'

[TRANS - ENV] Mandei um Segmento

[REDE - ENV] MTU Excedido MAX '150'

[REDE - FRAG] MTU excedido dividindo o pacote em max '150' bytes

[REDE - ENV] Não há ligacao do nó: '3'!

[ENLACE] ERRO de MTU Tamanho: '292', MTU: '150'

[REDE - ENV] MTU Excedido MAX '150'

[REDE - FRAG] MTU excedido dividindo o pacote em max '150' bytes

[REDE - ENV] Não há ligacao do nó: '3'!

[ENLACE] ERRO de MTU Tamanho: '292', MTU: '150'

[REDE - ENV] MTU Excedido MAX '150'

[REDE - FRAG] MTU excedido dividindo o pacote em max '150' bytes

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - ENV] Não há ligacao do nó: '3'!

[ENLACE] ERRO de MTU Tamanho: '292', MTU: '150'

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[REDE - ENV] MTU Excedido MAX '150'

[REDE - FRAG] MTU excedido dividindo o pacote em max '150' bytes

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[REDE - RCV] Datagrama Recebido com sucesso!

[TRANS - RCV] Novo Segmento

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

**[TRANS - ENV] Recebi um pacote de ack: '200'**

**[TRANS - ENV] Andei com a janela: '200'**

----------------------------------------------------------------------

[TRANS - ENV] Vou enviar a Janela

[TRANS - ENV] base: '200', nextseqnum: '200'

[TRANS - ENV] Mandei um Segmento

[TRANS - ENV] base: '200', nextseqnum: '250'

[TRANS - TIMER] Iniciei a Thread Timer!!

[TRANS - ENV] Mandei um Segmento

[TRANS - ENV] base: '200', nextseqnum: '300'

[REDE - ENV] Não há ligacao do nó: '3'!

[TRANS - ENV] Mandei um Segmento

[TRANS - ENV] base: '200', nextseqnum: '350'

[ENLACE] ERRO de MTU Tamanho: '292', MTU: '150'

[TRANS - ENV] Mandei um Segmento

[REDE - ENV] MTU Excedido MAX '150'

[REDE - FRAG] MTU excedido dividindo o pacote em max '150' bytes

[REDE - ENV] Não há ligacao do nó: '3'!

[ENLACE] ERRO de MTU Tamanho: '292', MTU: '150'

[REDE - ENV] MTU Excedido MAX '150'

[REDE - FRAG] MTU excedido dividindo o pacote em max '150' bytes

[REDE - ENV] Não há ligacao do nó: '3'!

[ENLACE] ERRO de MTU Tamanho: '292', MTU: '150'

[REDE - ENV] MTU Excedido MAX '150'

[REDE - FRAG] MTU excedido dividindo o pacote em max '150' bytes

[REDE - ENV] Não há ligacao do nó: '3'!

[ENLACE] ERRO de MTU Tamanho: '292', MTU: '150'

[REDE - ENV] MTU Excedido MAX '150'

[REDE - FRAG] MTU excedido dividindo o pacote em max '150' bytes

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

**[TRANS - ENV] Recebi um pacote de ack: '300'**

**[TRANS - ENV] Andei com a janela: '300'**

----------------------------------------------------------------------

[TRANS - ENV] Vou enviar a Janela

[TRANS - ENV] base: '300', nextseqnum: '400'

[TRANS - ENV] Mandei um Segmento

[TRANS - ENV] base: '300', nextseqnum: '450'

[TRANS - TIMER] Iniciei a Thread Timer!!

[TRANS - ENV] Mandei um Segmento

[REDE - ENV] Não há ligacao do nó: '3'!

[ENLACE] ERRO de MTU Tamanho: '292', MTU: '150'

[REDE - ENV] MTU Excedido MAX '150'

[REDE - FRAG] MTU excedido dividindo o pacote em max '150' bytes

[REDE - ENV] Não há ligacao do nó: '3'!

[ENLACE] ERRO de MTU Tamanho: '292', MTU: '150'

[REDE - ENV] MTU Excedido MAX '150'

[REDE - FRAG] MTU excedido dividindo o pacote em max '150' bytes

[TRANS - ENV] Thread de timer estourou. Reenviando Janela

----------------------------------------------------------------------

[TRANS - ENV] Vou enviar a Janela

[TRANS - ENV] base: '300', nextseqnum: '300'

[TRANS - ENV] Mandei um Segmento

[TRANS - TIMER] Iniciei a Thread Timer!!

[TRANS - ENV] base: '300', nextseqnum: '350'

[TRANS - ENV] Mandei um Segmento

[REDE - ENV] Não há ligacao do nó: '3'!

[TRANS - ENV] base: '300', nextseqnum: '400'

[TRANS - ENV] Mandei um Segmento

[TRANS - ENV] base: '300', nextseqnum: '450'

[ENLACE] ERRO de MTU Tamanho: '292', MTU: '150'

[TRANS - ENV] Mandei um Segmento

[REDE - ENV] MTU Excedido MAX '150'

[REDE - FRAG] MTU excedido dividindo o pacote em max '150' bytes

[REDE - ENV] Não há ligacao do nó: '3'!

[ENLACE] ERRO de MTU Tamanho: '292', MTU: '150'

[REDE - ENV] MTU Excedido MAX '150'

[REDE - FRAG] MTU excedido dividindo o pacote em max '150' bytes

[REDE - ENV] Não há ligacao do nó: '3'!

[ENLACE] ERRO de MTU Tamanho: '292', MTU: '150'

[REDE - ENV] MTU Excedido MAX '150'

[REDE - FRAG] MTU excedido dividindo o pacote em max '150' bytes

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[REDE - ENV] Não há ligacao do nó: '3'!

[ENLACE] ERRO de MTU Tamanho: '292', MTU: '150'

[REDE - ENV] MTU Excedido MAX '150'

[REDE - FRAG] MTU excedido dividindo o pacote em max '150' bytes

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

**[TRANS - ENV] Recebi um pacote de ack: '450'**

**[TRANS - ENV] Andei com a janela: '450'**

**[TRANS - ENV FIM DA JANELA] sizeof(buffer): '332', base final: '450', nextseqnum final: '500'**

**[APLIC - DESC]Recebi pedido para desconectar do env\_no: '3' e ps: '0'**

**[APLIC - RET]Conexão encerrada com sucesso! end\_buffer: '0x7f8e91e00000'**

**Nó 2:**

MacBook-Pro-de-Vitor-Vezani:Camada de Transporte macbookpro$ ./meuprog.exe ic.ini 2

Nome do arquivo: 'ic.ini'

Numero nó: '2'

[REDE - ROT]Enviei Tabela de Rotas para o nó '1'

[REDE - ROT]Enviei Tabela de Rotas para o nó '3'

[REDE - ROT]Enviei Tabela de Rotas para o nó '4'

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV]Repassando datagrama nó dts: '3', nó inicial: '1'!

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV]Repassando datagrama nó dts: '3', nó inicial: '1'!

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV]Repassando datagrama nó dts: '3', nó inicial: '1'!

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV]Repassando datagrama nó dts: '3', nó inicial: '1'!

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV]Repassando datagrama nó dts: '1', nó inicial: '3'!

[...]

**Nó 3:**

MacBook-Pro-de-Vitor-Vezani:Camada de Transporte macbookpro$ ./meuprog.exe ic.ini 3

Nome do arquivo: 'ic.ini'

Numero nó: '3'

[REDE - ROT]Enviei Tabela de Rotas para o nó '2'

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - ROT]Recebi tabela de rotas do nó '2'

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[TRANS - RCV] Recebi um segmento de connect!

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

**[TRANS - RCV] Recebi um segmento de connect!**

**[TRANS - RCV] Enviei um pacote de syn!**

[REDE - ENV] Não há ligacao do nó: '1'!

----------------------------------------------------------------------

[REDE - RCV] Datagrama Recebido com sucesso!

[TRANS - RCV] Novo Segmento

[TRANS - RCV] Recebi um segmento de connect!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[TRANS - RCV] Recebi um segmento de connect!

[REDE - ENV] Não há ligacao do nó: '1'!

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[TRANS - RCV] Recebi um segmento de connect!

[REDE - ENV] Não há ligacao do nó: '1'!

[REDE - ENV] Não há ligacao do nó: '1'!

[REDE - ENV] Não há ligacao do nó: '1'!

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

**[TRANS - RCV] SeqNum('50') != ExpectedSeqNum('0')**

**[TRANS - RCV] Pacote Descartado !**

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[TRANS - RCV] SeqNum('100') != ExpectedSeqNum('0')

[TRANS - RCV] Pacote Descartado !

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[TRANS - RCV] SeqNum('150') != ExpectedSeqNum('0')

[TRANS - RCV] Pacote Descartado !

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

**[TRANS - RCV] SeqNum('0') recebido == ExpectedSeqNum('0')**

**[TRANS - RCV] Enviando Pacote de ACK: '50'**

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - ENV] Não há ligacao do nó: '1'!

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[TRANS - RCV] SeqNum('50') recebido == ExpectedSeqNum('50')

[TRANS - RCV] Enviando Pacote de ACK: '100'

[REDE - ENV] Não há ligacao do nó: '1'!

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[REDE - RCV] Datagrama Recebido com sucesso!

[TRANS - RCV] SeqNum('100') recebido == ExpectedSeqNum('100')

[TRANS - RCV] Enviando Pacote de ACK: '150'

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - ENV] Não há ligacao do nó: '1'!

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[TRANS - RCV] SeqNum('150') recebido == ExpectedSeqNum('150')

[TRANS - RCV] Enviando Pacote de ACK: '200'

[REDE - ENV] Não há ligacao do nó: '1'!

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[TRANS - RCV] SeqNum('200') recebido == ExpectedSeqNum('200')

[TRANS - RCV] Enviando Pacote de ACK: '250'

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - ENV] Não há ligacao do nó: '1'!

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[TRANS - RCV] SeqNum('250') recebido == ExpectedSeqNum('250')

[TRANS - RCV] Enviando Pacote de ACK: '300'

[REDE - ENV] Não há ligacao do nó: '1'!

[REDE - RCV] ECC não correspondente - Datagrama Descartado

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[REDE - RCV] Datagrama Recebido com sucesso!

[TRANS - RCV] Novo Segmento

**[TRANS - RCV] SeqNum('350') != ExpectedSeqNum('300')**

**[TRANS - RCV] Pacote Descartado !**

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[REDE - RCV] ECC não correspondente - Datagrama Descartado

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[REDE - RCV] ECC não correspondente - Datagrama Descartado

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[TRANS - RCV] SeqNum('450') != ExpectedSeqNum('300')

[REDE - RCV] Datagrama Recebido com sucesso!

[TRANS - RCV] Pacote Descartado !

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

[REDE - RCV] Datagrama Recebido com sucesso!

----------------------------------------------------------------------

[TRANS - RCV] Novo Segmento

[TRANS - RCV] SeqNum('300') recebido == ExpectedSeqNum('300')

[TRANS - RCV] Enviando Pacote de ACK: '350'

[APLIC - RCV] Tam\_buffer: '292' Bytes

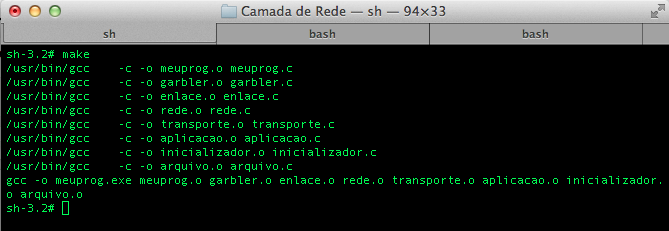
[APLIC - RCV] Buffer: 'O núcleo Linux foi desenvolvido pelo programador finlandês Linus Torvalds, inspirado no sistema Minix. O seu código fonte está disponível sob a licença GPL (versão 2) para que qualquer um o possa utilizar, estudar, modificar e distribuir livremente de acordo com os termos da licença.'

**Compilação e execução do programa**

**Compilação Mac OS X:**

Para compilar o programa é necessário gnu gcc.

* Ir no diretório do projeto.
* make



**Compilação no Ubuntu:**

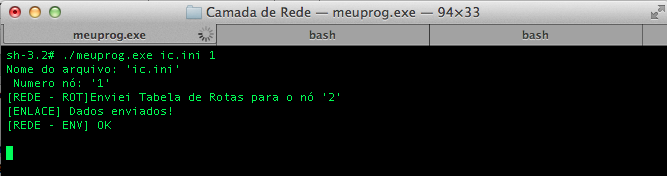
É necessário no makefile acrescentar o parâmetro de compilação '-lpthread' e trocar a chamada 'fpurge' para '\_\_fpurge'

**Execução:**

Para rodar o programa é necessário o sistema Mac OS X ou Ubuntu 12.10 ou superior

* Ir no diretório do projeto.
* ./meuprog.exe ic.ini x

x = numero do nó



**Cronograma e divisão de responsabilidades**

**Cronograma detalhado de implementação, semana a semana.**

[06/05/2013] - Primeiro Commit.

[07/05/2013] - Definindo API e Arquitetura.

[09/05/2013] - Definido a arquitetura da camada de transporte.

[16/05/2013] - Troca para apenas uma biblioteca.

[16/05/2013] - Minor changes.

[24/05/2013] - Logica da thread 'enviarSegmentos'.

[27/06/2013] - Adicionado logica do aps,fps,conectar,desconectar.

[30/06/2013] - Termino da logica de enviar/andar com a janela, receber pacote e acks.

[31/06/2013] - Adicionado a thread de job para enviar tabela de rotas (Camada de Redes).

[01/06/2013] - Funcionando Janela deslizante.

[01/06/2013] - Algumas mudanças, as vezes acontece deadlock.

[02/06/2013] - Não há mais deadlock.

Para maiores informações acessar: [Camada de Transporte - Github](https://github.com/vitorvezani/Camada-de-Transporte)

**Divisão de tarefas entre os membros do grupo**

**Programação:**

Primeiramente pensamos em alguns aspectos de solução do problema antes de começar a codificar.

Como nosso grupo é de apenas dois integrantes, adotamos a abordagem de 'programação em par' onde duas pessoas trabalham juntas. Eu (Vitor) codificava enquanto o outro integrante (Guilherme) revisava, procurando por erros e pensando na aplicação como um todo.

**Relatório:**

|  |  |
| --- | --- |
| **Vitor** | **Guilherme** |
| Introdução | Revisão |
| Diagramação |  |
| Especificação dos protocolos |  |
| Sumario |  |
| Plano de Implementação |  |
| Plano de integração e teste |  |
| Compilação e execução do programa |  |
| Cronograma e divisão de responsabilidades |  |
| Considerações finais |  |
| Referências e Bibliografia |  |

**Atividade III**

|  |  |
| --- | --- |
| **Vitor** | **Guilherme** |
|  | Atividade III |

**Considerações Finais**

Com a terceira parte do projeto foi possível, não só reforçar alguns conceitos aprendidos em nas aulas teóricas de redes I e redes II, como também usar e assim aprender conceitos, técnicas e tecnologias novas, treinando assim a parte teórica e pratica. Alguns exemplos estão listados abaixo:

* Melhor entendimento do funcionamento da camada de rede e a camada de transporte.
* Trabalho com fragmentação e desfragmentação de datagramas.
* Modularização de programas em C.
* Usar headers e variáveis extern.
* Treino em programação C.
  + Threads e sincronismo com mutex, exclusão mutua, estruturas, manipulação de arquivo.
* Diagramação do programa.

As maiores dificuldades encontradas foram no contexto de arquitetura e diagramação da solução a ser implementada. Problemas de sincronização de threads, acesso exclusivo à certos recursos também demandaram algum tempo. Problemas no entendimento do algoritmo de janela deslizantes, TCP e GBN também causaram algum tipo de problema. O destravamento da thread enviarSegmentos por duas partes do programa em paralelo deixou o programa com comportamento anômalo e foi bem complicado achar uma outra solução.

Para atividades futuras sugerimos de novo uma melhor divisão de atividades entre os integrantes do grupo para que assim não sobrecarregue nenhuma das partes.

**Referências e Bibliografia**

**ROSS**, KEITH W.; **KUROSE**, JAMES F. Redes de Computadores *e a Internet: Uma abordagem top-down*. 5.ª edição.