

INSTITUTO FEDERAL
SANTA CATARINA

MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA

CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO JOSÉ

DIÁRIO DE BORDO

PROTOCOLO DE ENLACE: ENTREGA 2

Marcelo Bittencourt do Nascimento Filho

Março de 2021, São José



INSTITUTO FEDERAL
SANTA CATARINA

MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

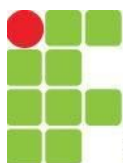
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA

CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO JOSÉ

Diário de bordo

A tabela abaixo contém o histórico de contribuição na segunda etapa do projeto de Protocolo de Enlace.

Data	Descrição da atividade
21/01/2021	Revisão das aulas gravadas referente ao assunto <i>mecanismo ARQ</i> . Em aula foi discutido alguns mecanismo que poderemos implementar, dentre eles <i>Stop-and-Wait</i> , <i>Go-Back-N</i> , <i>Selective-Repeat</i> . Em muitos materiais didáticos e em outras referências, há o equívoco de informar que esses mecanismos são protocolos, na realidade eles fazem parte de um protocolo, tendo a função de garantir a entrega dos pacotes. Os conceitos de eficiência e eficácia foram abordados, a fim de escolher o melhor dos três mecanismos para o cenário no qual estamos trabalhando. É importante destacar que todos são eficazes, ou seja, cumprem a função de garantia de entrega, porém se diferenciam em sua eficiência ("porcentagem de sucesso"). Quanto maior a eficiência, mais complexo fica a implementação do mecanismo. Dessa forma, foram realizados alguns cálculos com os dados do nosso <i>link</i> de comunicação, sendo foi possível perceber que o modelo <i>Stop-and-Wait</i> obteve um alto valor de eficiência (cerca de 99,6%) e juntando com o fato de ser o mecanismo mais fácil de implementar, o mesmo foi escolhido para fazer parte do nosso protocolo.
23/01/2021	Leitura do material recomendado sobre os mecanismos ARQ. Dei início à modelagem da Máquina de Estados Finita referente o mecanismo <i>Stop-and-Wait</i> para a aplicação transmissora e a receptora. Utilizei um bom tempo para entender bem todas as etapas e exceções que podem ocorrer em um <i>link</i> de comunicação que utiliza esse serviço em seu protocolo de enlace. Com a utilização do <i>software</i> DIA fiz o diagrama FSM para as duas aplicações, porém, acredito que não esteja totalmente correto, necessitando de alguns ajustes. A equipe combinou um dia da semana para se reunirmos e conversarmos o que cada um fez sobre o tópico.
24/01/2021	Revisão do que foi realizado no último dia. Percebi alguns erros e fiz um texto para explicar melhor as etapas da FSM. Mesmo com essas correções acredito que haverá algumas mudanças quando conversarmos com o professor.



INSTITUTO FEDERAL
SANTA CATARINA

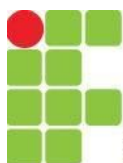
MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA

CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO JOSÉ

Data	Descrição da atividade
03/02/2021	Primeiro dia de aula após as férias. Meu grupo teve a oportunidade de apresentar em aula a modelagem inicial que desenvolvemos sobre o mecanismo ARQ. O professor relatou que a nossa modelagem referente à aplicação transmissora estava correta, porém fez ressalvas quanto à receptora. Para um melhor entendimento do que se deveria ser feito foi realizada uma nova abordagem de análise, utilizando diagramas no tempo que representavam a comunicação entre duas aplicações. Com isso, foi possível obter uma modelagem de estado único para a aplicação receptora (a qual ficou muito mais simples e inteligível do que a desenvolvida pela equipe). Dessa forma, foi possível representar com apenas uma máquina de estado os processos de transmissão e recepção do mecanismo ARQ. Realizei um documento final especificando todas as etapas da Máquina de Estados Finita desenvolvida e coloquei no repositório <i>master</i> da equipe.
07/02/2021	Iniciei a implementação em <i>software</i> do mecanismo ARQ, para isso criei uma classe denominada de CallbackARQ que irá conter toda lógica necessária para o funcionamento da garantia de entrega. Por enquanto, há somente um "esqueleto" da classe, com algumas variáveis e métodos sem nenhuma funcionalidade, porém foi possível ter uma noção de onde a subcamada desta etapa deve estar. Realizei a criação de uma nova classe que irá representar um <i>frame</i> , porém fiquei com dúvidas quanto a minha implementação e espero tirá-las com o professor na aula seguinte. Outras dúvidas: <ol style="list-style-type: none">1. Na subcamada CallbackARQ o que será o método <i>handle</i>?2. Ao enviar um ACK, o quadro será formatado de forma diferente?3. Como tratar o Timeout na MEF, como um If ou habilitar no início da máquina?
08/02/2021	Durante a aula consegui sanar todas as dúvidas do dia anterior além de explicar como pensei em construir a classe Quadro, sendo que o professor aprovou minha maneira de desenvolvimento. Dessa forma, pude detalhar melhor o pseudocódigo referente à MEF do mecanismo ARQ, assim como apliquei a funcionalidade da classe Quadro em todas as subcamadas desenvolvidas até então. Minha maior dificuldade está sendo em como representar cada campo do Quadro, no momento estou utilizando cada campo como <i>bytes</i> (exemplo b'quadro de teste"), porém acredito que futuramente posso utilizar a classe <i>bytearray</i> ()).



INSTITUTO FEDERAL
SANTA CATARINA

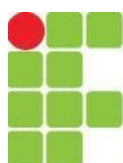
MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA

CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO JOSÉ

	<p>Mesmo com esse detalhe, a aplicação está funcionando. Dúvidas para o próximo atendimento:</p> <ol style="list-style-type: none">1. Como utilizar a classe <i>bytearray()</i>? Do modo que estou realizando (b'palavra') não consigo acessar cada item do <i>byte</i>.2. O que acontece no <i>handle_timeout</i>?3. Declarar o nSeq tx e rx como <i>boolean</i> ou não?
09/02/2021	<p>Fui ao atendimento paralelo e consegui sanar as dúvidas do dia anterior. A classe <i>Quadro()</i> foi totalmente alterada, os itens a seguir demonstram as mudanças realizadas:</p> <ol style="list-style-type: none">1. As variáveis que representam os campos do quadro estão definidas agora como do tipo inteiro e não mais como <i>byte</i>.2. Utilização de <i>decorators property</i> e <i>setter</i> para atuar como métodos <i>getter</i> e <i>setter</i>.3. Criação de um novo método que irá realizar a construção do quadro utilizando <i>bytearray()</i>. <p>Ocorreram também mudanças na classe <i>CallbackARQ()</i>, sendo elas:</p> <ol style="list-style-type: none">1. Utilização de uma classe que implementa <i>Enum</i> a fim de representar os estados da MEF.2. Criação de uma classe para representar os Eventos da MEF. <p>Depois de todas essas dicas e alterações acredito que o desenvolvimento se tornará mais fácil e claro.</p>
11/02/2021	<p>Dúvidas e respostas discutidas no atendimento paralelo:</p> <ol style="list-style-type: none">1. Preciso guardar globalmente o quadro que chega do Enquadramento e o que foi enviado anteriormente? R: Apenas o que foi enviado anteriormente para casos de retransmissão, em especial no evento <i>Payload</i>.2. Condição para o segundo evento <i>timeout</i>? R: Realizar o tratamento de qual evento realizar (<i>DATA</i> ou <i>ACK_TX</i>) diretamente no método <i>recebe()</i>.3. Em um quadro <i>ACK</i> deixamos o campo <i>payload</i> vazio ou quando montarmos o quadro todo não incluímos esse campo? R: Deixamos vazio, não irá causar problemas ao realizar a montagem do quadro completo. Dica: Para selecionar os bytes do <i>payload</i> na classe <i>Quadro()</i>: <code>quadro[3:-2]</code>.4. Como ir para o Estado <i>ACK_TX</i>? R: Com verificação no método <i>recebe()</i>.5. Como realizar o <i>not RX/TX</i>? R: Através da operação <i>XOR</i>. $x = x \wedge 1$.



INSTITUTO FEDERAL
SANTA CATARINA

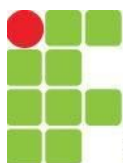
MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA

CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO JOSÉ

15/02/2021	Realizei a adequação da subcamada Enquadramento ao padrão da classe Quadro(). Tentei implementar o padrão de Eventos, porém não obtive sucesso, visto que a simplificação do código a qual era esperada não foi alcançada. Problemas encontrados: Ao enviar uma mensagem obtida pela aplicação, a mesma é enviada normalmente porém logo em seguida entra na condição de de Estados.Espera > TipoEvento.DATA > evento.dados.sequencia == self.rx, sendo que o esperado seria aguardar uma mensagem vinda da subcamada inferior. Irei marcar um atendimento para tentar resolver essa questão, pois acredito que esse é o último detalhe para a conclusão desta etapa.
16/02/2021	Durante o atendimento relatei o que percebi ao tentar aplicar o padrão de classes TipoEvento e Evento na Subcamada Enquadramento. O professor demonstrou alguns métodos para implementar esta etapa, e dentre eles escolhi o de passar para o método desenquadra um parâmetro informado se o evento a ser tratado é de Timeout ou não. Algumas melhorias de lógica e em especial na função finaliza() também foram discutidas. Não conseguimos descobrir o erro que está acontecendo ao enviar uma mensagem, porém o professor sugeriu realizar mais testes.
17/02/2021	Em aula foi dado o início da discussão a respeito do mecanismo de acesso ao meio que será implementado, o qual foi escolhido o método ALOHA. Trabalhei na classe enquadramento e dei continuidade com as dicas vistas no atendimento, precisei implementar um método na classe Quadro responsável por limpar o quadro do método envia() em Enquadramento, pois ao enviar a segunda mensagem a verificação do CRC estava afirmando como incorreta. Hoje ainda pretendo tratar o erro que está acontecendo com a classe ARQ.
18/02/2021	Problema encontrado no dia 15/02/2021 foi resolvido hoje, o erro estava no método setter Tipo da classe Quadro() o qual estava definido referenciando outro método (método sequência). A previsão de hoje é realizar mais testes para concluir essa etapa e dar início à fase de acesso ao meio.
25/02/2021	Fomos ao atendimento discutir sobre o tópico de Mecanismo de Acesso ao Meio. Conseguimos sanar todas as dúvidas, assim como discutimos bastante sobre a questão dos <i>timers</i> .
28/02/2021	Realizei a modelagem da máquina de estados do mecanismo ARQ junto com o controle de acesso ao meio. Consegui desenvolver o <i>software</i> referente à essa etapa. Não consegui fazer com que o <i>timeout</i>



INSTITUTO FEDERAL
SANTA CATARINA

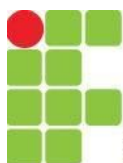
MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA

CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO JOSÉ

	dos estados de <i>Backoff</i> sejam aleatórios, mas acredito que isso não seja um ponto crítico de se resolver.
01/03/2021	Fomos ao atendimento e o professor deu algumas dicas, dentre elas retirar a lógica do método <i>handle_timeout()</i> e acrescentar tudo à máquina de estados. Realizei as devidas alterações, também conseguimos resolver o problema do tempo aleatório para o <i>timer</i> . Hoje iremos marcar um novo atendimento para revisar a subcamada ARQ e tentar resolver a Tun.
02/03/2021	Realizamos outro atendimento, e finalmente acredito ter finalizado a subcamada ARQ com Mecanismo de Acesso ao Meio. Um dos pontos que alterei hoje foi realizar um <i>reload</i> do <i>timeout</i> no Estado Ocioso e no Evento Payload. Realizei vários testes e todos funcionaram como deveriam.
04/03/2021	Realizamos os testes finais do projeto, e percebemos que um erro estava ocorrendo na classe Enquadramento. O comportamento do erro era que sempre que um quadro era descartado o programa não avançava mais. Analisei todo o funcionamento da máquina de estados dessa classe e percebi que no estado de Escape não estava ocorrendo o retorno para o estado de Recebendo em determinados eventos. Também alterei alguns erros relacionados ao acionamento do Timeout.
05/03/2021	Criamos o relatório e realizamos mais uma vez os testes finais, aparentemente tudo está funcionando como deveria.



INSTITUTO FEDERAL
SANTA CATARINA

MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA

CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO JOSÉ

CONCLUSÃO

Nessa etapa do protocolo participei diretamente no desenvolvimento da Subcamada ARQ e de todas as suas funcionalidades, incluindo o Mecanismo de Acesso ao Meio. Nos primeiros dias de trabalho, procurei compreender totalmente a parte teórica referente à garantia de entrega realizando leituras, revendo aulas e modelando a Máquina de Estados Finita. Após sanar algumas dúvidas iniciei a implementação em *software*, porém o professor sugeriu que fizéssemos uma alteração na estrutura do projeto para que o torna-se mais flexível. A mudança realizada foi a criação de uma classe para representar toda a estrutura de um quadro, a qual foi denominada de Quadro. Com essa classe todas as subcamadas se comunicam através de um objeto quadro, sendo que cada uma é responsável por um desses campos.

Concluindo a classe Quadro comecei o desenvolvimento da subcamada ARQ, que com sua utilização tornou tudo mais fácil pois fez com que o acesso aos campos do quadro fizesse mais direto, sem a implementação de lógicas complexas para consultar os valores dos *bytes*. O professor também sugeriu a criação de classes dentro da subcamada ARQ para representar os estados e os eventos da máquina de estados, o que também tornou a leitura do código muito mais agradável e organizada.

Nas semanas seguintes surgiu uma nova funcionalidade para o nosso protocolo denominado de Mecanismo de Acesso ao Meio, o qual é responsável por garantir que dois pontos do canal não o utilizem no mesmo momento. Para isso, escolhemos implementar a tecnologia Aloha e toda a sua lógica de funcionamento foi adicionada à mesma máquina de estados da ARQ, alterando alguns fluxos e adicionando novos estados.

Passei por bastante dificuldades nessa etapa, em especial nos tratamentos dos diferentes *timeout* existentes. Porém, revendo as aulas e indo em vários atendimentos consegui realizar essa etapa e fazer alguns testes, os quais todos apresentam o resultado esperado. Por fim, o último ajuste a se fazer no projeto foi implementar a subcamada Tun que irá substituir a nossa atual camada de Aplicação responsável por gerar dados para a transmissão. Com essa nova subcamada haverá uma integração direta com o sistema operacional e utilizar os pacotes das interfaces de redes como dados para a transmissão. Não participei diretamente do desenvolvimento dessa etapa, porém acompanhei os atendimentos referentes a esse assunto e auxiliei em atividades para concluí-la.

Dessa forma, o desenvolvimento do protocolo de enlace me permitiu conhecer novas tecnologias e relembrar outras já vistas em outras disciplinas do curso. Poder integrá-las e fazer com seus funcionamentos se completem foi uma experiência muito agradável, pois pude perceber que assim como outros tipos de projetos, a criação de um projeto é constituído por etapas que vão se complementando e dando forma a algo que pode ser utilizado em diversas outras aplicações.