Sistemas de Telecomunicações 2022 / 2023

Aula prática: P10

Aluno: Martim Duarte Agostinho n 62964

Protocolo: GoBackN

1. Emissão

Ficheiro

GoBackN.java

Variáveis de estado

|  |
| --- |
| * **win\_size**   Tamanho da janela de transmissão |

|  |
| --- |
| * **next\_dframe\_ts**   Número de sequência da próxima trama DATA a enviar |

|  |
| --- |
| * **seq\_buff**   Número de sequencia da primeira trama do buffer |

|  |
| --- |
| * **it\_buff**   Itera o buffer, aponta para a posição no buffer que contem a próxima trama para enviar. |

|  |
| --- |
| * **sending\_buffer**   Buffer que contem as *N ultimas tramas que enviei.* |

Eventos recebidos / Ação

|  |
| --- |
| * **start\_simulation**   Arranque da aplicação / Envia primeira trama DATA |
| * **handle\_Data\_end**   Fim de envio de trama DATA / Começa o temporizador da trama de DATA, e envia a próxima trama de DATA |

|  |
| --- |
| * **handle\_Data\_Timer**   Fim do temporizador de trama de DATA / Reenvia as últimas *N* tramas |

|  |
| --- |
| * **from\_physical\_layer**   Recebe tramas:  **–>ACK:** Trata de confirmar que recebeu todos os pacotes ate ao numero de sequencia do ACK  **–>NAK:** Confirma que recebeu todos os pacotes ate ao numero de sequencia anterior ao NAK e reenvia todos os pacotes a partir do numero sequencia do NAK. |

|  |
| --- |
| * **send\_next\_data\_packet**   Envia a próxima trama de dados, se houver próxima trama de dados por enviar |

|  |
| --- |
| * **get\_next\_frame**   Retorna o próximo pacote de DATA a ser enviado, guardando-o também no buffer, e atualizando as variáveis associadas |

|  |
| --- |
| * **send\_Dpacket**   Recebe o pacote a ser enviado, cria a trama de DATA e envia-a |

|  |
| --- |
| * **ack\_handler**   Recebe a sequencia do ack e gere o seu tratamento |

|  |
| --- |
| * **roll\_back\_it**   Recebe a sequencia do primeiro pacote a ser reenviado, atualiza o iterador de forma a reenviar o que houver no *buffer*. |

1. Receção

Ficheiro

GoBackN.java

Variáveis de estado

|  |
| --- |
| * **retrans\_state**   Booleano , indica se estou no tempo entre ter enviado um NAK e receber a DATA esperada. |

|  |
| --- |
| * **dframe\_expected**   Número de sequencia da próxima trama de DATA a receber |

Eventos recebidos / Ação

|  |
| --- |
| * **handle\_ack\_Timer**   Fim do temporizador de *ACK* / Envia o *ACK* sem *piggybacking.* |

|  |
| --- |
| * **from\_physical\_layer**   Recebe tramas:  **–>DATA** (campos ***seq*** e ***info***) / Se a sequencia for a esperada, envio o pacote para a camada de rede, começo o temporizador para enviar o ACK e verifico se houve *piggybacking.* Se não tiver enviado um NAK e a sequencia estiver dentro da janela de transmissão envio um NAK. De outra forma envio um ACK. |

|  |
| --- |
| * **send\_NAK**   Cria uma trama nak e envia, com a sequencia especificada. |

|  |
| --- |
| * **send\_ack**   Cria uma trama ack e envia***.*** |

Protocolo: GoBackN\_FlowC ( incompleto )

1. Emissão

Ficheiro

GoBackN\_FlowC.java

Variáveis de estado

|  |
| --- |
| * **win\_size**   Tamanho da janela de transmissão |

|  |
| --- |
| * **next\_dframe\_ts**   Número de sequência da próxima trama DATA a enviar |

|  |
| --- |
| * **seq\_buff**   Número de sequencia da primeira trama do buffer |

|  |
| --- |
| * **it\_buff**   Itera o buffer, aponta para a posição no buffer que contem a próxima trama para mandar. |

|  |
| --- |
| * **sending\_buffer**   Buffer que contem as *N ultimas tramas que enviei.* |

|  |
| --- |
| * **rcv\_buff**   Tamanho da janela do recetor. |

|  |
| --- |
| * **first\_packet**   Booleano que indica se estou a enviar o primeiro pacote. |

Eventos recebidos / Ação

|  |
| --- |
| * **start\_simulation**   Arranque da aplicação / Envia primeira trama DATA |
| * **handle\_Data\_end**   Fim de envio de trama DATA / Começa o temporizador da trama de DATA, e envia a próxima trama de DATA |

|  |
| --- |
| * **handle\_Data\_Timer**   Fim do temporizador de trama de DATA / Reenvia as últimas *N* tramas |

|  |
| --- |
| * **from\_physical\_layer**   Recebe tramas:  **–>ACK:** Trata de confirmar que recebeu todos os pacotes ate ao numero de sequencia do ACK  **–>NAK:** Confirma que recebeu todos os pacotes ate ao numero de sequencia anterior ao NAK e reenvia todos os pacotes a partir do numero sequencia do NAK. |

|  |
| --- |
| * **send\_next\_data\_packet**   Envia a próxima trama de dados, se houver próxima trama de dados por enviar |

|  |
| --- |
| * **get\_next\_frame**   Retorna o próximo pacote de DATA a ser enviado, guardando-o também no buffer, e atualizando as variáveis associadas |

|  |
| --- |
| * **send\_Dpacket**   Recebe o pacote a ser enviado, cria a trama de DATA e envia-a |

|  |
| --- |
| * **ack\_handler**   Recebe a sequencia do ack e gere o seu tratamento |

|  |
| --- |
| * **roll\_back\_it**   Recebe a sequencia do primeiro pacote a ser reenviado, atualiza o iterador de forma a reenviar o que houver no *buffer*. |

1. Receção

Ficheiro

GoBackN\_FlowC.java

Variáveis de estado

|  |
| --- |
| * **retrans\_state**   Booleano , indica se estou no tempo entre ter enviado um NAK e receber a DATA esperada. |

|  |
| --- |
| * **dframe\_expected**   Número de sequencia da próxima trama de DATA a receber |

Eventos recebidos / Ação

|  |
| --- |
| * **handle\_ack\_Timer**   Fim do temporizador de *ACK* / Envia o *ACK* sem *piggybacking.* |

|  |
| --- |
| * **from\_physical\_layer**   Recebe tramas:  **–>DATA** (campos ***seq*** e ***info***) / Se a sequencia for a esperada, envio o pacote para a camada de rede, começo o temporizador para enviar o ACK e verifico se houve *piggybacking.* Se não tiver enviado um NAK e a sequencia estiver dentro da janela de transmissão envio um NAK. De outra forma envio um ACK. |

|  |
| --- |
| * **send\_NAK**   Cria uma trama nak e envia, com a sequencia especificada. |

|  |
| --- |
| * **send\_ack**   Cria uma trama ack e envia***.*** |