Cap. 04 – Controle de Fluxo

4.1 – Introdução
4.2 – Protocolo de Janela
4.3 – Número de Sequência
4.4 – Reconhecimento Negativo
4.5 – Evitando Congestionamento

Referências Bibliográficas

- Gerard J. Holzmann – “Design and Validation of Computer Protocols” – Prentice Hall; Englewood Clifs; New Jersey; 1991.

- Paulo Coelho - “Material de Aula” - Arquitetura de Redes de Computadores (FACOM49070 - Mecatrônica)
- Pedro Frosi - “Material de Aula” - Arquitetura de Redes de Computadores (GBC056 - Ciência da Computação)

4.1 - Introdução

- “Controle de Fluxo” - relacionado com o controle do diálogo na troca de mensagens entre as entidades pares.
- “esquema simples de controle de fluxo” - consiste em ajustar a velocidade da origem das primitivas à velocidade com que o receptor pode receber e processar as primitivas
- … esquemas mais elaborados podem evitar erros tais como: remoção, inserção, duplicação e reordenação.

- Nota: Observem que isto é o que acontece nas camadas mais baixas da pilha de comunicação, p.ex., camada física.

… 4.1 - Introdução

- Fundamentos Filosóficos do Controle de Fluxo .. 04 pilares:

- garantia que as primitivas são enviadas em uma frequência que a entidade receptora pode receber e processá-las;
- otimização na utilização do canal;
- diminuição ou eliminação da “perda de carga” do canal;
- distribuição criteriosa do uso dos recursos de comunicação.

… 4.1 - Introdução

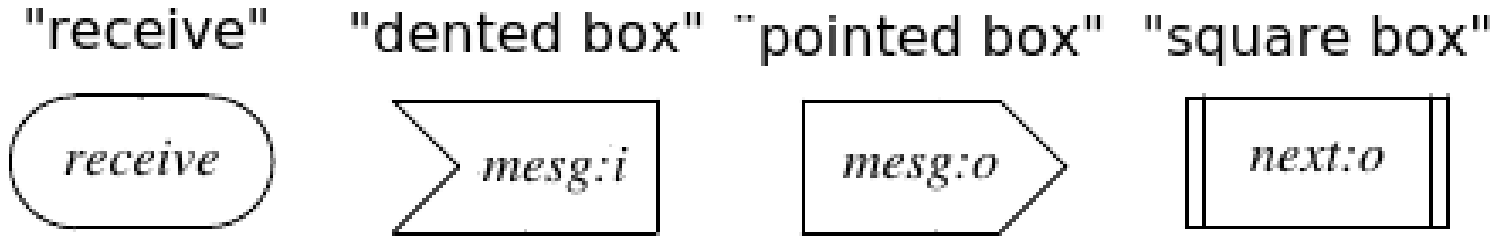
- O caminho entre a origem e destino pode conter esquemas de interconexão com as seguintes características:
- … capacidade limitada para “storing” and “forwarding”;
- … compartilhamento entre vários pares de entidades.

- Obs.: … esquemas prudentes de controle de fluxo evitam que pares monopolizem todo o espaço de recursos disponíveis.

- “Objetivo” - … tendo por base um protocolo básico, propõe-se a concepção de um modelo de controle de fluxo.

… 4.1 - Introdução

- “receive” - representa o estado no qual a recepção de uma nova mensagem do canal está sendo aguardada;
- “dented box” - representa o reconhecimento de uma msg. que está associada - “match” com o rótulo - “label” da caixa;
- “pointed box” - indica a transmissão de uma msg. cujo tipo é indicado pelo rótulo - “label” da caixa;
- “square/retangle box” - indica uma ação interna para obter o próximo item de dado, p.ex., caracter a ser transferido.



… 4.1 - Introdução

- “objetivo” - … tendo por base um protocolo básico, propõe-se a concepção de um modelo de controle de fluxo.
- … protocolo “simplex” - … usado para transmitir dados em apenas um sentido (canal “half-duplex”)

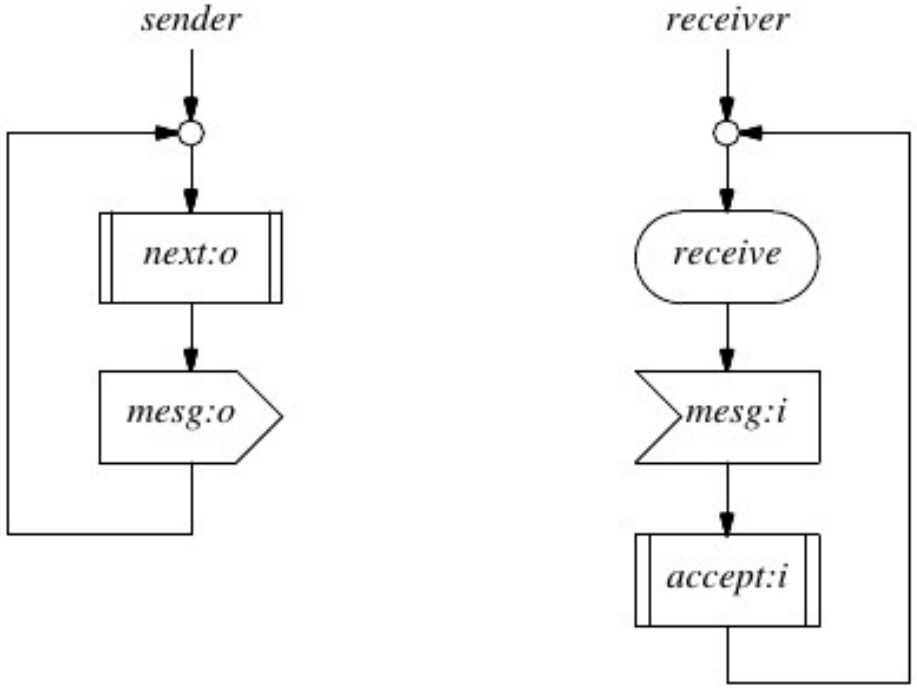


Figure 4.1 — No Flow Control

… 4.1 - Introdução

- “premissa” - protocolo apresentado funciona apenas se o receptor RX for mais rápido que transmissor TX.
  - … se esta suposição for falsa, a entidade que envia pode sobrecarregar o esquema de entrada da entidade receptora.

- 1º Postulado:

  “Never make assumptions about the relative

  speeds of concurrent entities.”

- Nota: … gargalo do protocolo é provavelmente o receptor, logo, não se deve assumir que o receptor suporte o transmissor.

… 4.1 - Introdução

- … como a recepção é geralmente mais dispendiosa (processamento) do que a transmissão, então receptor deve:
  - interpretar as primitivas; decidir o que fazer com elas;
  - alocar memória; … até encaminhá-las ao recurso final.

- ... transmissão não precisa de um provedor para funcionar (ativa-se quando há algo para transmitir), mas quem transmite deve:
  - liberar memória após a transmissão (liberação de recurso);
  - não assumir que o receptor irá se adaptar à entidade que transmite, ou seja, não assumir sincronismo entre transmissor e receptor.

… 4.1 - Introdução

- “Técnica mais Antiga” de controle de fluxo para tratar a sincro-nização, sem que seja necessário negociação a priori entre receptor estabelece o compasso entre msgs. transmitidas ...
- … método utiliza 02 msgs. de controle:
  - … uma msg. para suspender → “x-off”
  - … uma msg. para retomar o tráfego → “x-on”

- “premissa” - canal é livre de erros e o vocabulário do protocolo contempla 03 (três) mensagens:

$$V = \{ \text{mesg}, \text{suspend}, \text{resume} \}$$

… 4.1 - Introdução

- Seja um protocolo “duplex”, no qual o vocabulário contempla 03 tipos de mensagens, ou seja, V = { “mesg”, “suspend”, “resume”}

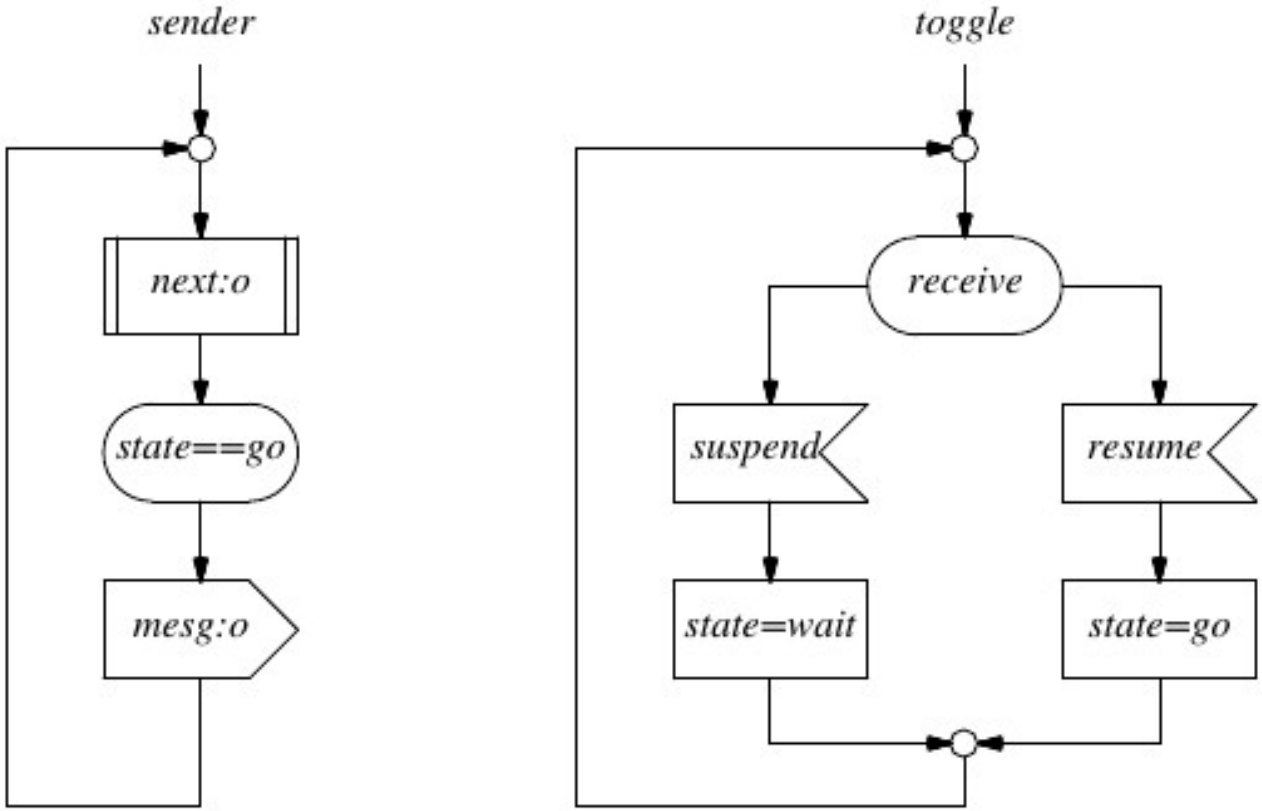
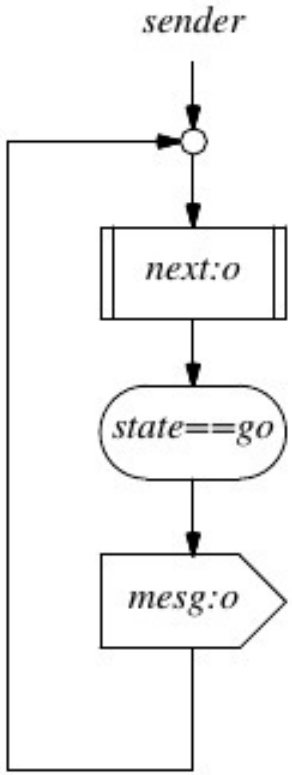
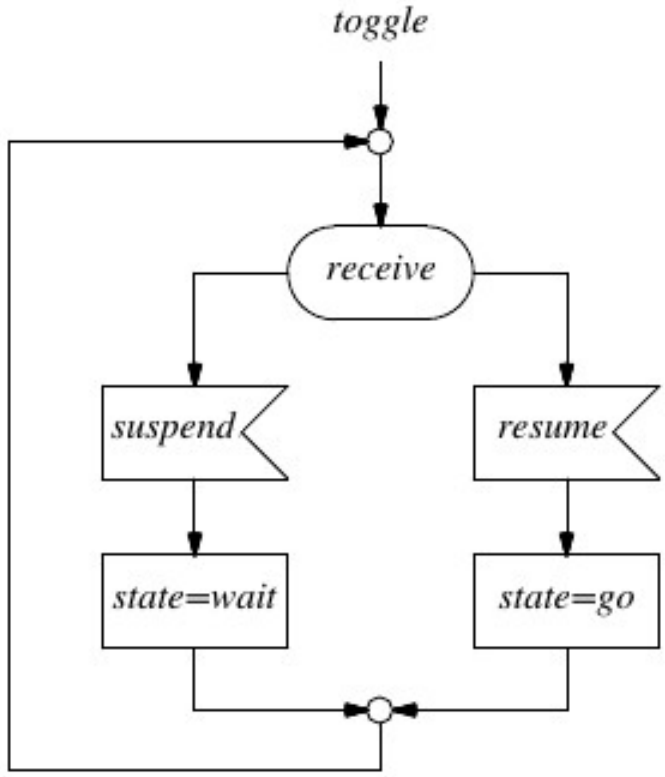


Figure 4.2 — X-on/X-off Protocol: Sender Processes

… 4.1 - Introdução

- Protocolo “X-on/X-off” … no remetente (transmissor):
  - mensagem “mesg” apenas é enviada se “state” = “go”;
  - estado é alterado pelo recebimento de msg. “suspend” ou “resume”.





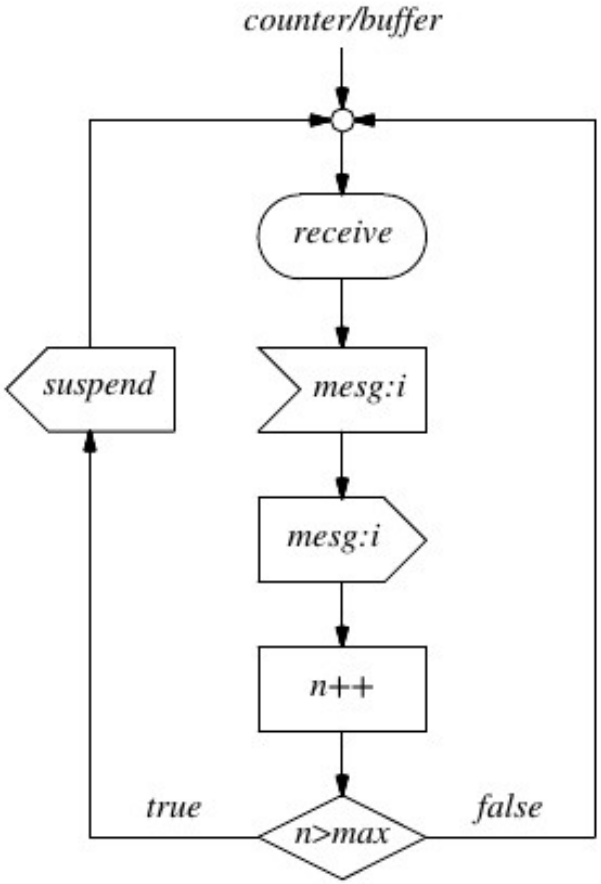
X-on/X-off Protocol: Sender Processes

… 4.1 - Introdução

- Protocolo “X-on/X-off” - receptor também é dividido em 02 partes:
- … após a chegada de uma msg. de dados o processo contador incrementa a variável “n”;
  - “n” representa o nro. de msgs. que foram recebidas do transmissor e que ainda estão esperando para serem aceitas pelo receptor;
  - … se o valor de “n” ultrapassa um limite, uma msg. “suspend” é enviada para o transmissor (controle de fluxo);
  - … se o valor de “n” decresce abaixo de um limite inferior, uma msg. “resume” é enviada para o transmissor (controle de fluxo).
  - … qdo o receptor aceita uma mensagem, o processo “acceptor” decrementa a variável “n”.
- Obs.: … msg. de dados é passada do processo “contador” para o processo “acceptor” através de um fila interna.

… 4.1 - Introdução

- … após a chegada de uma msg. de dados o processo contador incrementa a variável “n”;



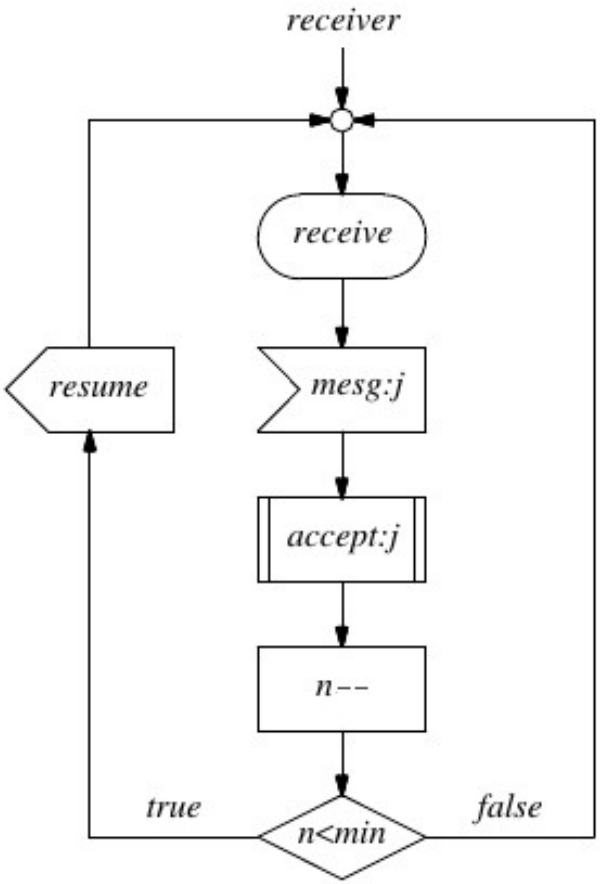


Fig 4.3 — X-on/X-off Protocol: Receiver Processes

… 4.1 - Introdução

- … “n” representa o nro. de msgs. que foram recebidas do transmissor e que aguardam serem aceitas pelo receptor;

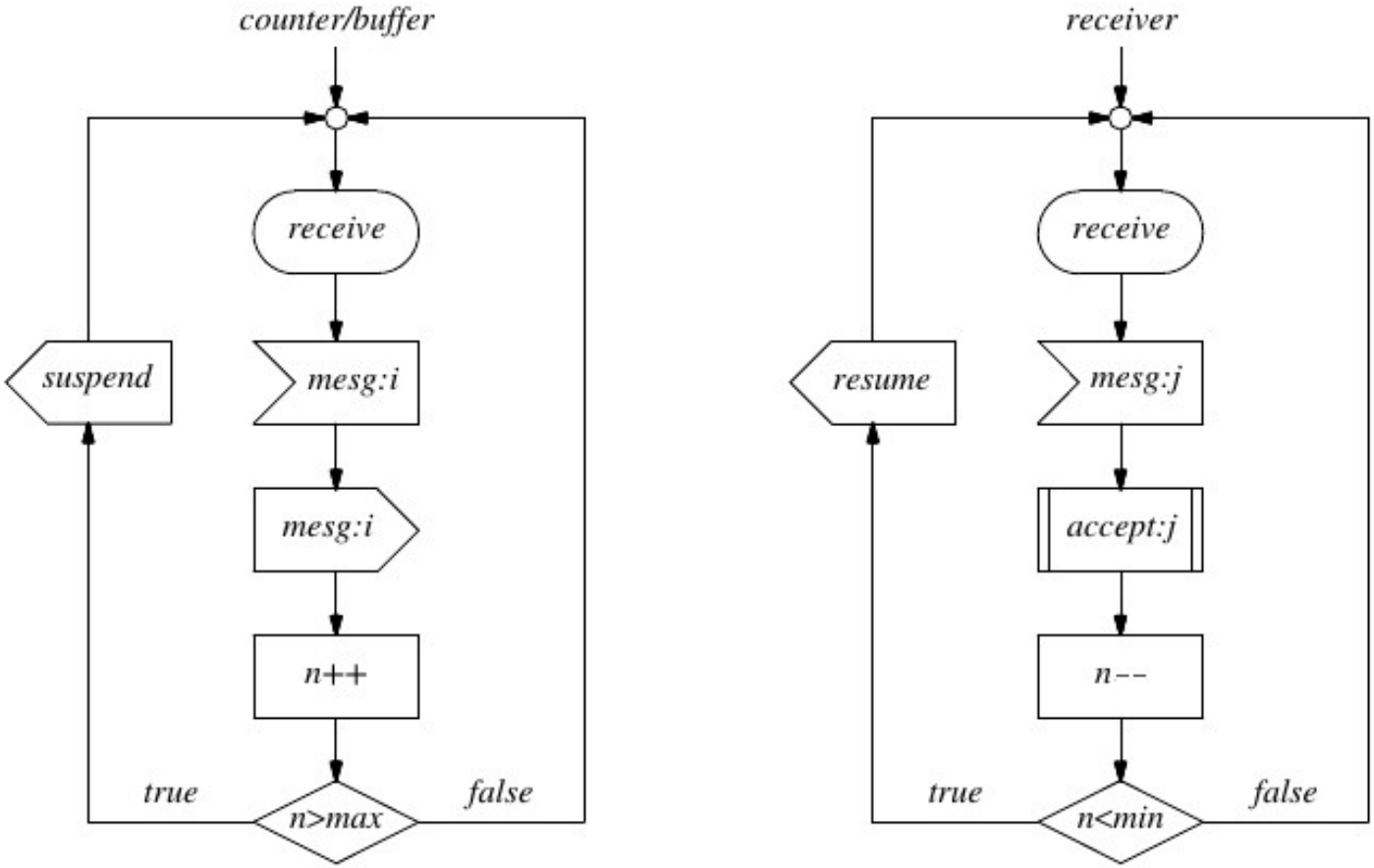
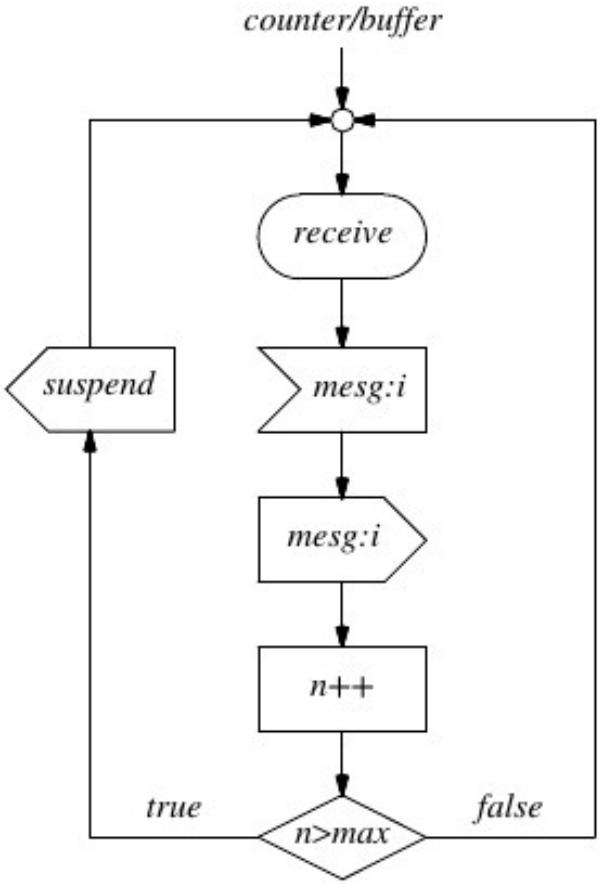


Fig 4.3 — X-on/X-off Protocol: Receiver Processes

… 4.1 - Introdução

- … se o valor de “n” decresce abaixo de um limite inferior, uma msg. “resume” é enviada para o transmissor (controle de fluxo).



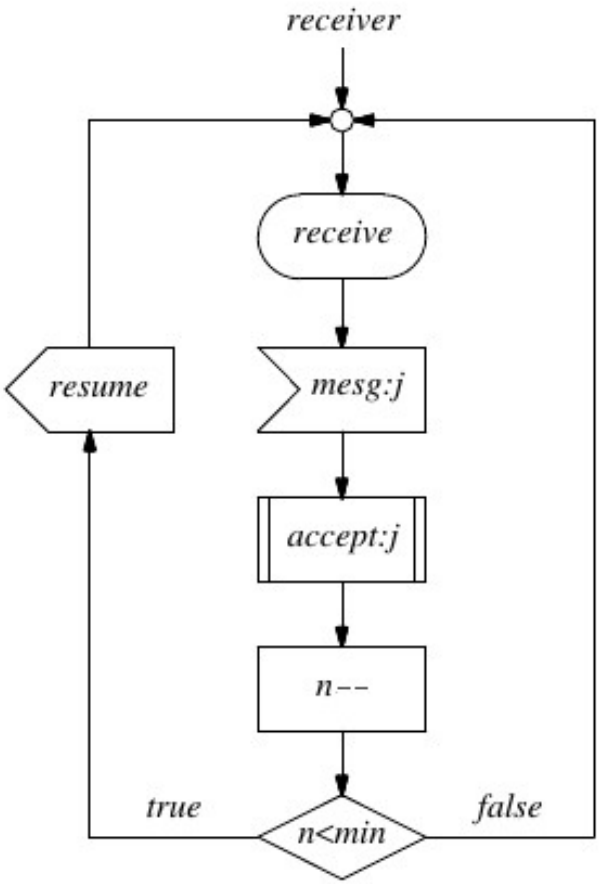
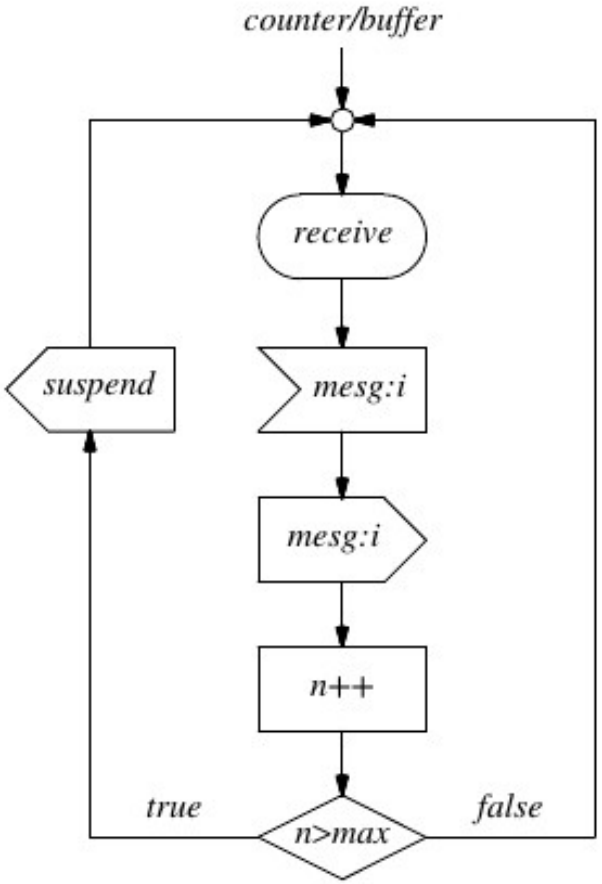


Fig 4.3 — X-on/X-off Protocol: Receiver Processes

… 4.1 - Introdução

- … qdo o receptor aceita uma mensagem, o processo “acceptor” decrementa a variável “n”.



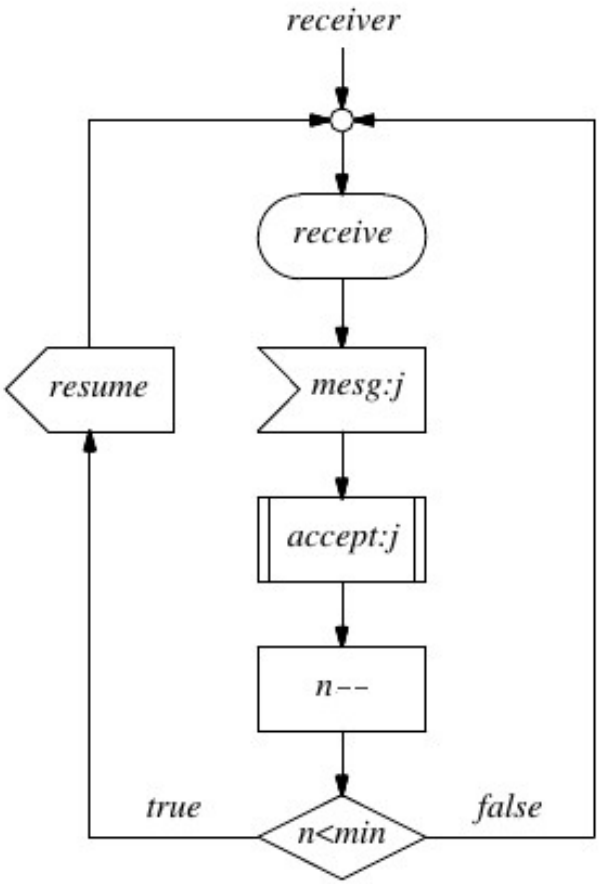
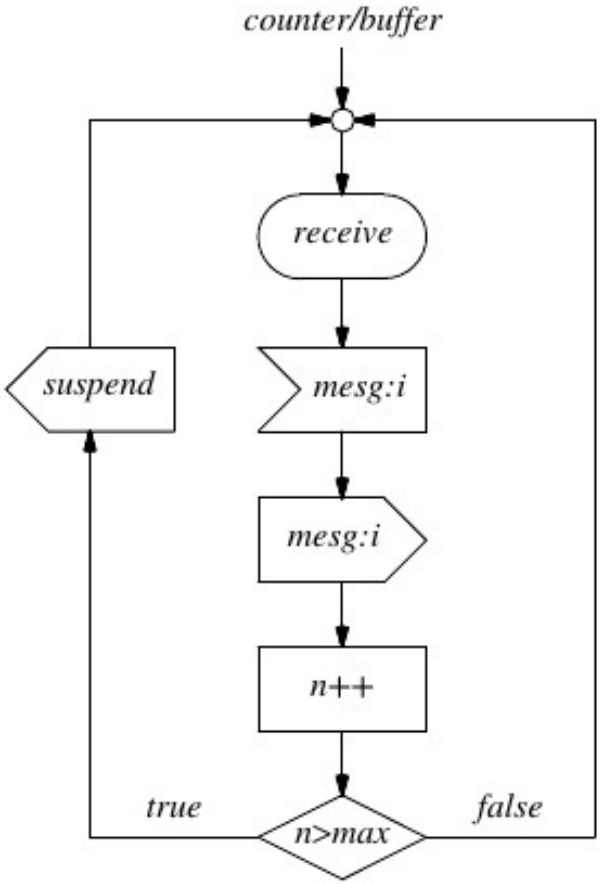


Fig 4.3 — X-on/X-off Protocol: Receiver Processes

… 4.1 - Introdução

- Obs.: … msg. de dados é passada do processo “contador” para o processo “acceptor” através de um fila interna.



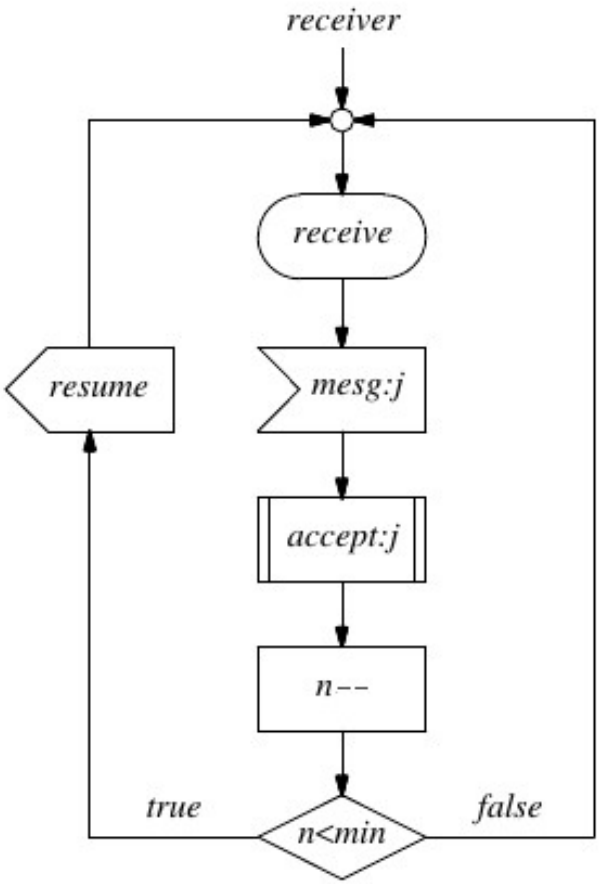


Fig 4.3 — X-on/X-off Protocol: Receiver Processes

… 4.1 - Introdução

- Problemas do Protocolo X-on/X-off:
  - total confiança no meio de transmissão (não há erros ou perdas);
  - se mensagem “suspend” é perdida/atrasada, pode ocorrer “overflow”;
  - se msg “resume” é perdida, tem-se um colapso total do sistema;
  - funcionamento de um protocolo não deve depender do tempo que uma mensagem de controle leva para atingir o destino – descreva este cenário ??

- Logo, 02 principais problemas devem ser resolvidos:
  - erros de “overrun” (transbordo) de maneira mais confiável;
  - proteção contra perda de mensagens (suspend / resume).

… 4.1 - Introdução

- “solução para erros de transbordo” - para resolver este problema basta deixar o transmissor esperar explicitamente o reconhecimento das mensagens já transmitidas.
  - e.g. Protocolo “Ping-Pong” ou Protocolo “Stop and Wait”

- “Ping-Pong Protocol” - … problema do transbordo é resolvido, mas o sistema ainda pode sofrer bloqueio se o controle ou alguma mensagem de dados se perder no canal.

- “premissa” - total confiança no canal ou meio de transmissão, pois não há erros ou perdas no canal. (Ainda não é o Caso Real !)

… 4.1 - Introdução

- “Ping-Pong Protocol” - .. problema do transbordo é resolvido, mas o sistema ainda pode sofrer bloqueio se o controle ou alguma mensagem de dados se perder no canal.
  - “t” - tempo de propagação em um canal;
  - “a” - tempo que o receptor precisa para processar e aceitar a mensagem que acabou de ser entregue pelo canal;
  - “p” - tempo que o transmissor precisa para preparar a msg. a fim de a mesma possa ser transmitida pelo canal.

- … com estas considerações, transmissor terá atraso de “2t + a - p” unidades de tempo para toda e qualquer primitiva transmitida.
  - premissa – threads em paralelo para “p” e para “2*t + a”

… 4.1 - Introdução

- “Protocolo Ping-Pong” - … mesmo que Protocolo “Stop and Wait”;

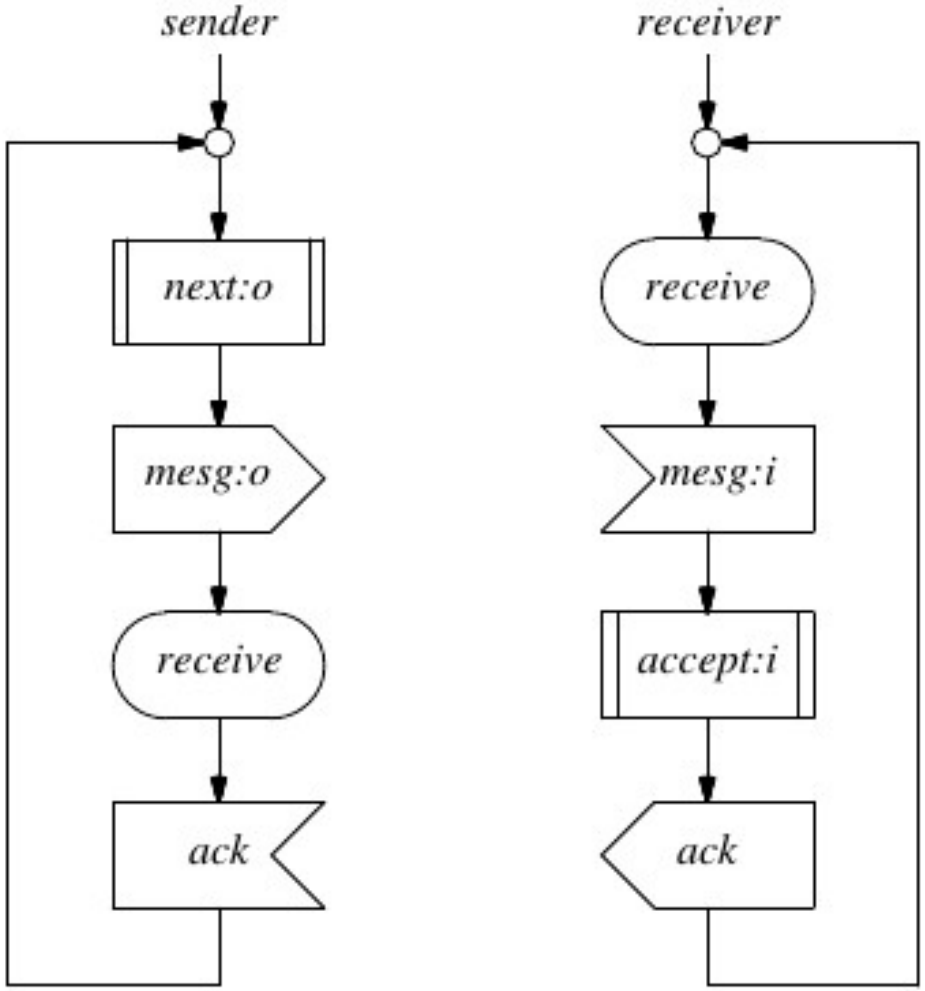


Figure 4.4 — Ping-Pong Protocol

… 4.1 - Introdução

- “relembrando” as variáveis ...
  - “t” - tempo de propagação em um canal;
  - “a” - tempo que o receptor precisa para processar e aceitar a mensagem que acabou de ser entregue pelo canal;
  - “p” - tempo que o transmissor precisa para preparar a msg. a fim de a mesma possa ser transmitida pelo canal.

- “p < a” - obviamente que “t” aumenta no mínimo linearmente com a distância entre o transmissor e receptor.
  - … observe-se que o reconhecimento não significa apenas a chegada da última primitiva, mas também significa um crédito que o receptor oferece ao transmissor para que envie a próxima primitiva.

… 4.1 - Introdução

- “p < a” - obviamente que “t” aumenta no mínimo linearmente com a distância entre o transmissor e receptor;
- … observe-se que o reconhecimento não significa apenas a chegada da última primitiva;
- … reconhecimento também significa um crédito que o receptor oferece ao transmissor para que envie a próxima primitiva.

- Obs.: Sequência de passos contempla a idéia de janela!

4.2 – Protocolo de Janelas

- Na fase de estabelecimento de conexão, as entidades definem quanto de buffer, banda, etc. haverá para a comunicação;
  - … entidades definem espaços para um número de requisições pendentes, espaço este comumente referenciado como crédito;
  - … créditos podem ser alterados dinamicamente quando os espaços para primitivas pendentes mudarem;
  - “premissa” - assumir que por hora não há perdas de primitivas.

- Cada primitiva recebida é reconhecida com um único reconhecimento ou “acknowledgement” no sentido contrário.
  - … tudo o que se tem a fazer é manter a contabilidade das mensagens em trânsito … (ao longo do canal).

… 4.2 – Protocolo de Janelas

- … crédito inicial pode ser negociado ou pode ser predefinido para um número de mensagens “W”.

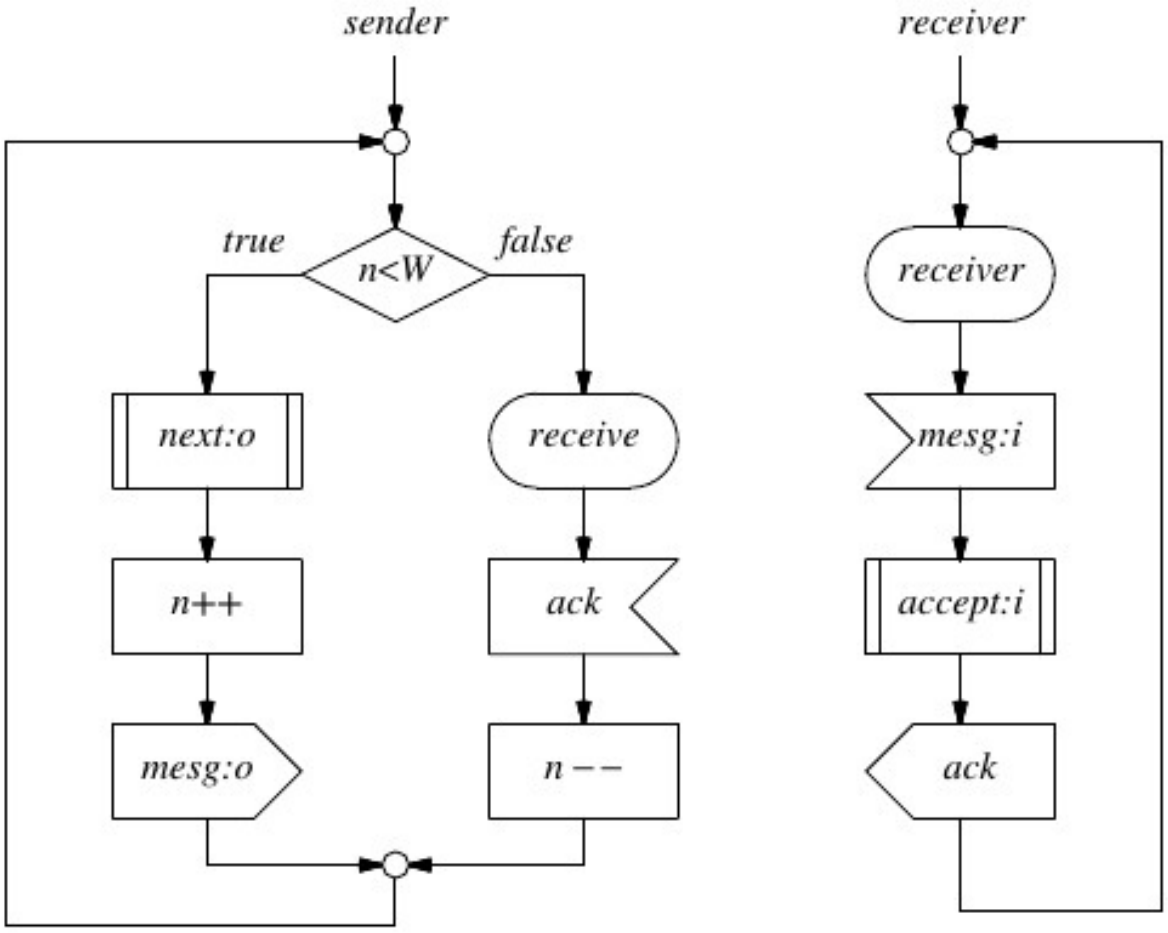


Fig 4.5 — Window Protocol for an Ideal Channel

… 4.2 – Protocolo de Janelas

- … para cada primitiva enviada, TX decrementa seu crédito;
- … para cada primitiva recebida por RX, devolve-se 01 novo crédito ao TX através do canal de retorno.

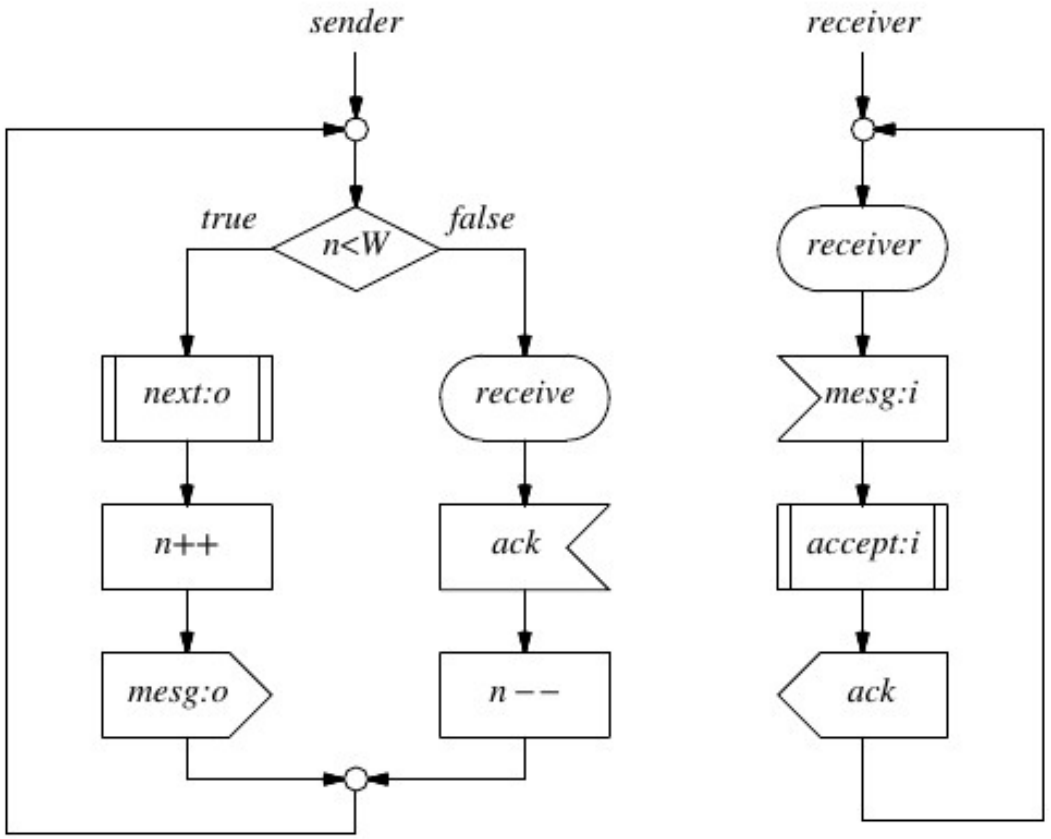


Fig 4.5 — Window Protocol for an Ideal Channel

… 4.2 – Protocolo de Janelas

- … “W – n” representa o nro. de créditos não utilizados.

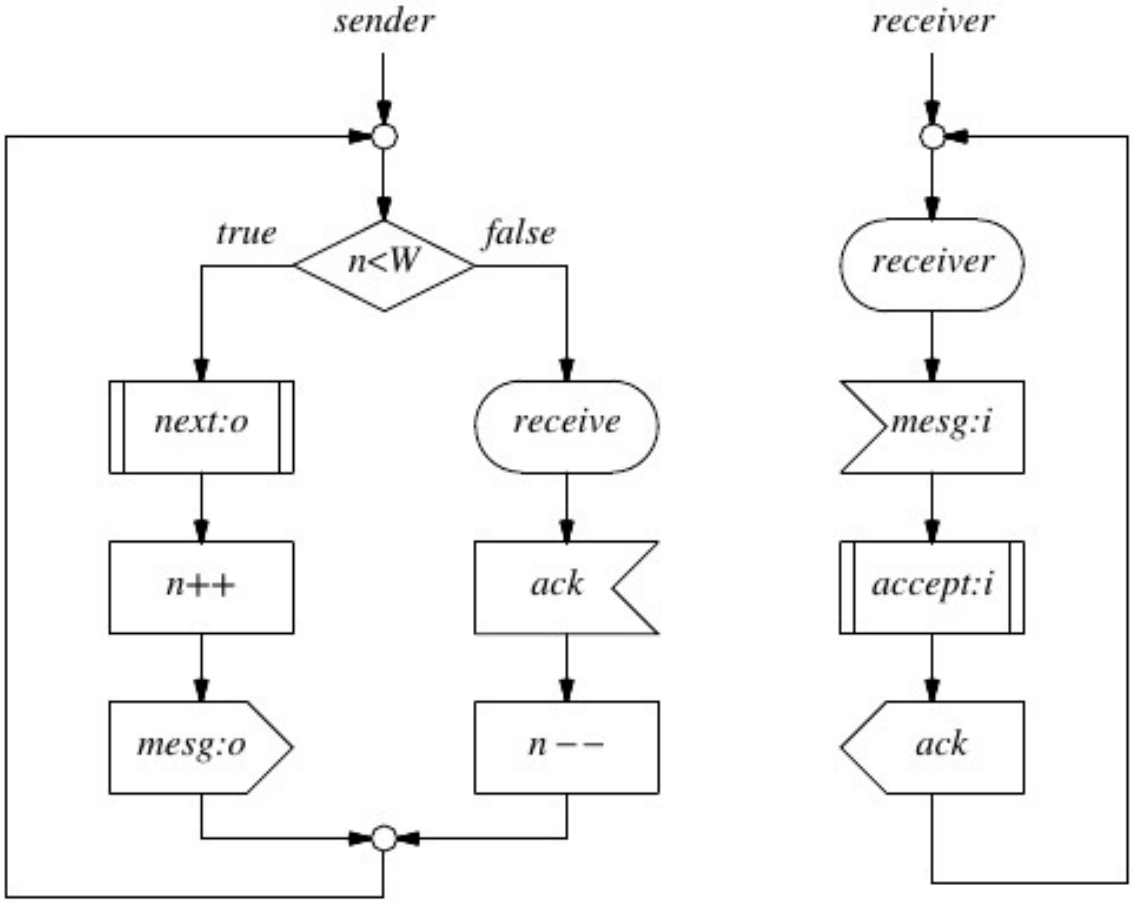


Fig 4.5 — Window Protocol for an Ideal Channel

… 4.2 – Protocolo de Janelas

- Seja
  - a(t) - número de créditos recebidos pelo transmissor no instante t;
  - p(t) - número de primitivas enviadas ao receptor no instante t;
  - n(t) - nro de msgs. pendentes de reconhecimento no instante t.

- O máximo número de primitivas que o transmissor pode ter pendentes, ou seja, esperando reconhecimento, é :

$$W - n(t) + p(t) - a(t)$$

  - W – n(t) ... número de créditos disponíveis;
  - p(t) - a(t) ... número de créditos usados.

… 4.2 – Protocolo de Janelas

- Como o número máximo de primitivas pendentes de reconhe-cimento - “acknowledgement” é W:

$$W – n(t) + p(t) - a(t) \leq W$$

$$\text{i.e. } p(t) - a(t) \leq n(t)$$

  - Lembrete.: Inicialmente todas as variáveis na inequação são 0 (zero).

- Obs.: O máximo número de créditos (W), que é o máximo nro. de primitivas pendentes de reconhecimento é denominado o tamanho da janela ou “window size”.

… 4.2 – Protocolo de Janelas

- Toda ação de envio no transmissor incrementa ambos os lados da inequação, lado direito primeiro e assim preserva sua validade;
- … similarmente, cada ação de recepção do processo receptor decrementa ambos os lados de 1, inicialmente o lado esquerdo, e novamente preserva a corretude.

$$\text{i.e. } p(t) - a(t) \leq n(t)$$

- Obs.: Há de fato concordância com as definições apresentadas para a(t) e p(t) ? … elas se referem a visão do transmissor ? … n(t) não tem o mesmo significado de p(t) ? … discuta …

… 4.2 – Protocolo de Janelas

- Como visto, o valor máximo de “W” é o tamanho da janela - “window size” do protocolo.
- … durante a transferência de dados, o número de créditos varia entre 0 e “W-1”, dependendo da velocidade entre TX e RX.
- … canais com tempo de trânsito alto podem ser otimizados se habilitarmos o transmissor a enviar uma ou mais primitivas enquanto espera por uma confirmação.
- … os problemas citados em comunicações (inserções, duplicações, ...) ainda persistem, exigindo um melhor controle de fluxo.

… 4.2 – Protocolo de Janelas

- Protocolo Ping-Pong com “timeouts”

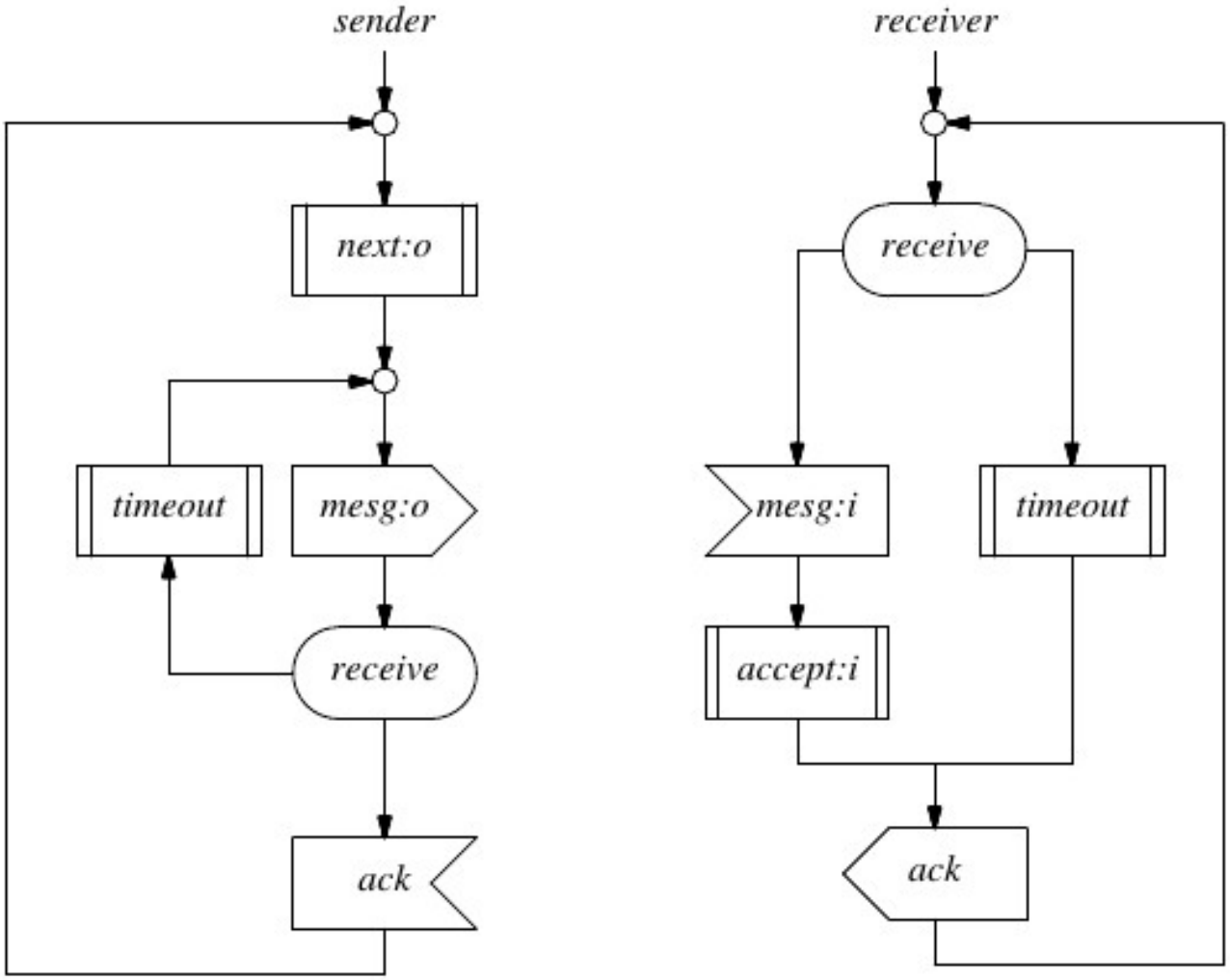


Figure 4.6 — Ping-Pong Protocol with Timeouts

… 4.2 – Protocolo de Janelas

- Com uso de múltiplas primitivas pendentes por reconhecimento, é necessário manter rastreado o controle do tempo decorrido entre as transmissões e as recepções;
- … para isto é necessário dimensionar o pior caso nos tempos de transmissão, isto para não considerar como perdida uma primitiva em pleno trânsito, que ainda não chegou no receptor.
- A expressão a seguir é frequentemente utilizada:

$$T_{worst} = T_{médio} + N * sqrt(var(T))$$

  - T é o tempo de ida e volta da primitiva
  - N é tipicamente 1 e raramente 2

… 4.2 – Protocolo de Janelas

- Comportamento de entidades fins e o canal de comunicação podem ser modelados como um processo de Markov M/M/1.
- Neste caso tem-se que: $var(T) = T_{médio} * T_{médio}$

- … esta é uma consideração importante pois o cálculo de desvio padrão, ou seja, sqrt(var(T)) envolve as medidas anteriores e isto nem sempre está disponível.
- Se considerarmos N=1, temos: $T_{worst} = 2.T_{médio}$

… 4.2 – Protocolo de Janelas

- Um erro comum, entretanto, é quando ambas as entidades fins implementam mecanismos de “timeout”;
- … ambos transmissor e receptor decidem retransmitir a última msg. enviada se o erro de remoção ocorre.

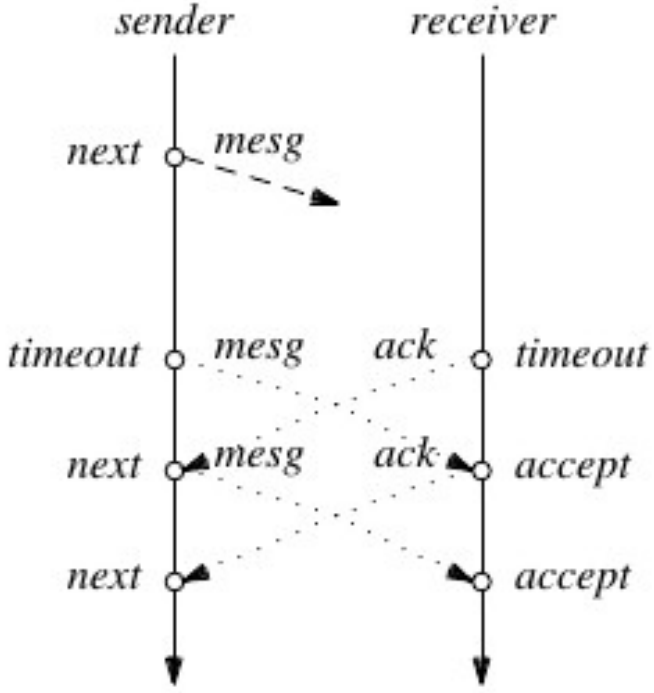


Figure 4.7 — Time Sequence Diagram of An Error

… 4.2 – Protocolo de Janelas

- “conclusão” - ambos transmissor e receptor não devem estar habilitados para iniciar retransmissões;
  - … suficiente colocar esta habilidade em uma das duas entidades;
  - … tradicionalmente é o transmissor que tem esta prerrogativa uma vez que somente ele tem certeza de quando um novo dado deve ser enviado.

- Deve haver um mecanismo que identifique exatamente a qual primitiva pertence um reconhecimento que acaba de chegar
  - mesmo se houver uma primitiva pendente em qualquer instante;
  - isto normalmente é feito por um número de seqüência (seq. number);
  - como números são finitos (necessário mecanismo para verificar os nro. de sequência reutilizados ou reciclados).

4.3 – Número de Sequência

- Número de sequência rotula cada primitiva enviada, que por sua vez terá msg. de reconhecimento - “acknowledgement”;
  - … i.e., número da primitiva reconhecida está presente, de alguma forma, na mensagem de reconhecimento.

- e.g., … considere um número de seqüência de um bit, isto permite analisar o uso de “timeout” no Protocolo Bit Alternante.

- e.g., … como exemplo de um melhor uso do “timeout” e do nro. de sequência de 1 bit, vamos considerar uma “versão estendida” do Protocolo Bit Alternante.

… 4.3 – Número de Sequência

- Bartlett, Scantlebury and Wilkinson [1969] – propuseram o “Protocolo Bit Alternante” - 02 FSM com 6 estados cada.
- … propuseram procedimentos para alcançar transmissão “full-duplex” confiável em enlaces “half-duplex”;
- … esquema proposto foi comparado com esquemas do mesmo tipo, descritos na mesma época (década de 1960).
- W.C. Lynch - “Reliable Full-Duplex File Transmission over Half-Duplex Telephone Lines”; Communications ACM; 1968.
- … método proposto utiliza 01 bit de controle que se mostrou infalível, desde que todos os erros possam ser detectados.
  - … semelhante ao esquema de transmissão proposto por W.C. Lynch.

… 4.3 – Número de Sequência

- “Protocolo Bit Alternante” - 02 FSM com 6 estados cada.
- “arestas” - especifica a troca de mensagens, sendo que cada aresta é identificada por 02 letras, p.ex., B0; B1; A0; A1 … etc.

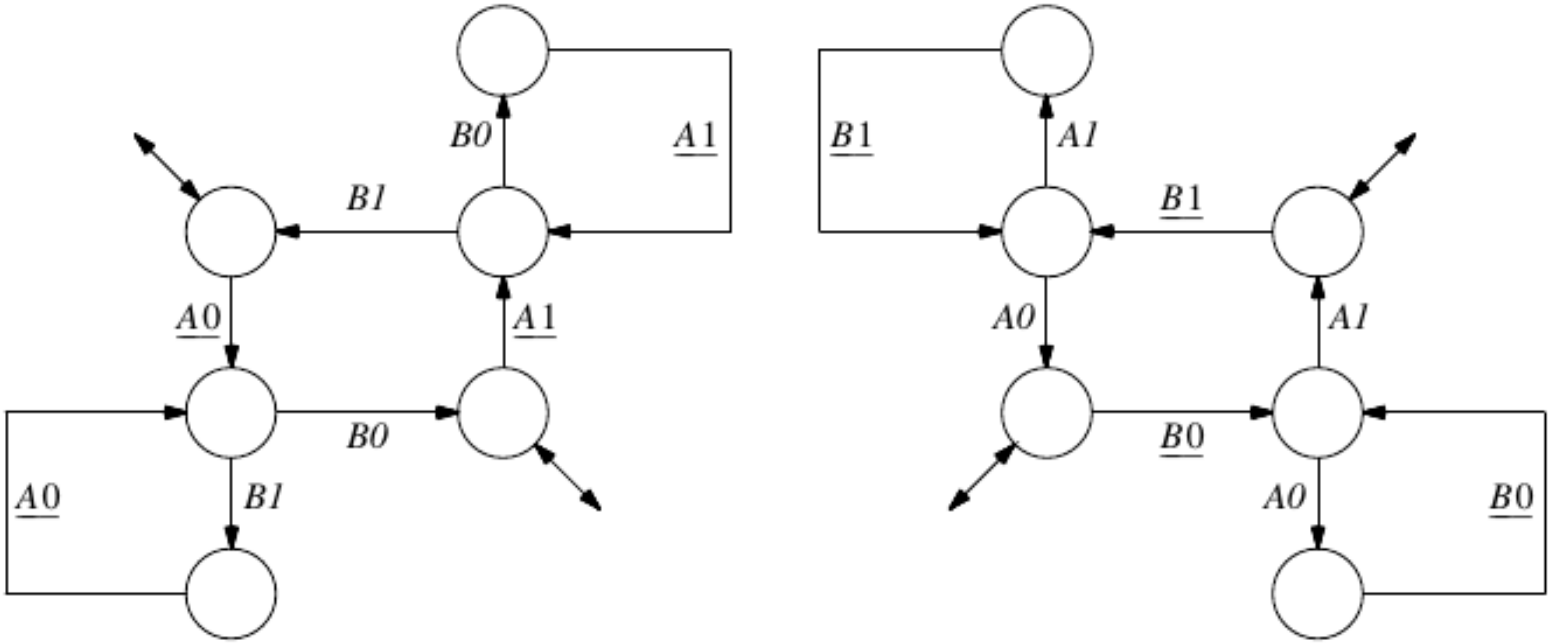


Figure 4.8 — Original Alternating Bit Protocol

… 4.3 – Número de Sequência

- … 1ª letra especifica a origem da msg. sendo recebida ou transmitida, enquanto a 2ª letra especifica o nro. de sequência;
- … aresta com 02 setas indica entrada para ser aceita no receptor ou uma nova msg. será transmitida no transmissor;

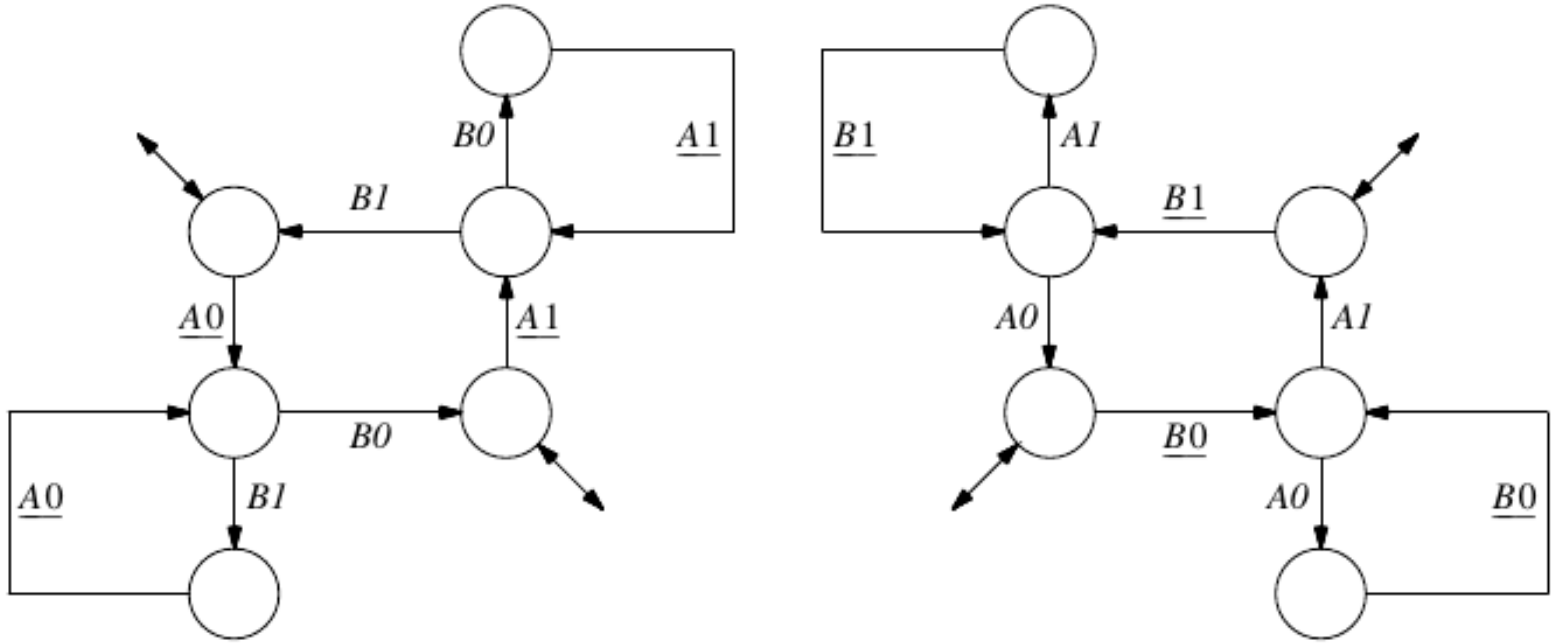


Figure 4.8 — Original Alternating Bit Protocol

… 4.3 – Número de Sequência

- … entradas errôneas, ou seja, msgs. que carregam um nro de sequência incorreto, causam uma retransmissão da última mensagem já transmitida.

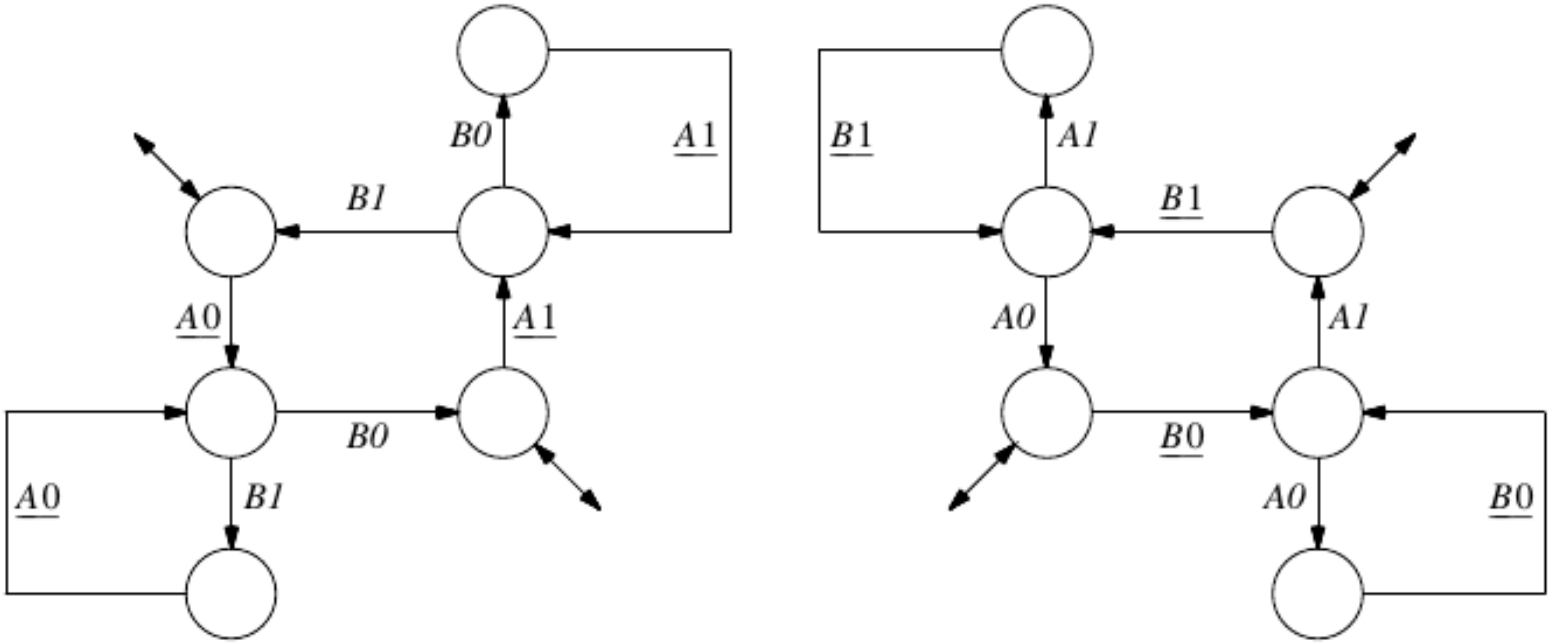


Figure 4.8 — Original Alternating Bit Protocol

… 4.3 – Número de Sequência

- e.g., pesquisar o protocolo e entender os detalhes.
- Bartlett, Scantlebury and Wilkinson [ACM - 1969] – propuseram o “Protocolo Bit Alternante” - 02 FSM com 6 estados cada.

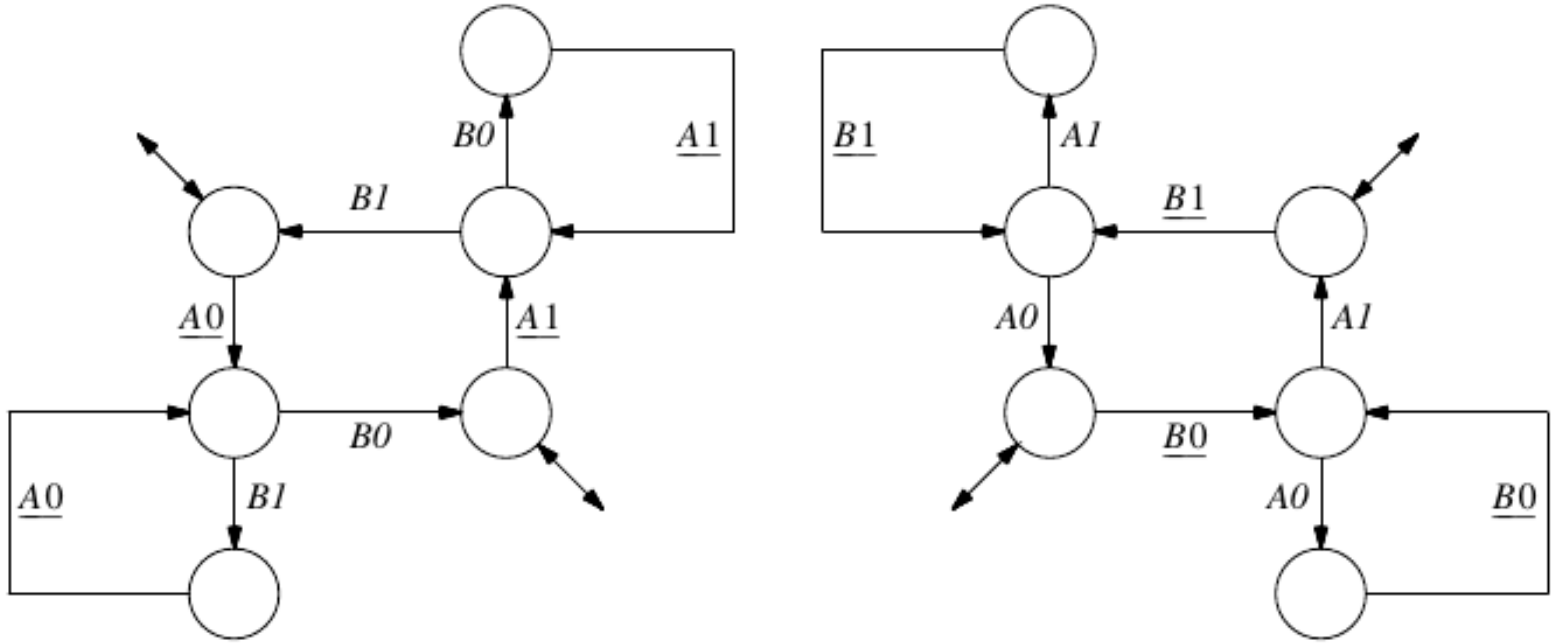


Figure 4.8 — Original Alternating Bit Protocol

… 4.3 – Número de Sequência

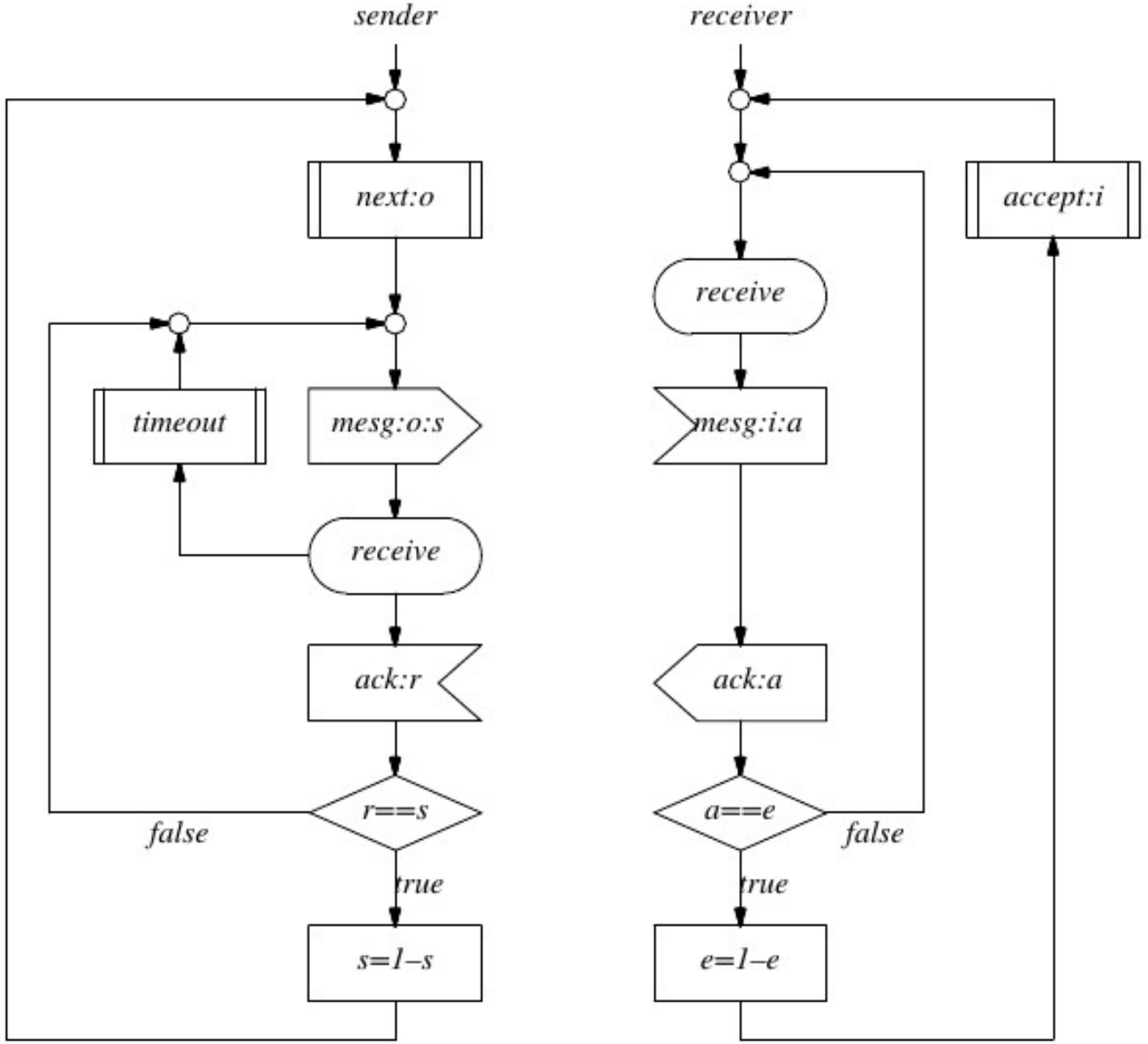


Figure 4.9 — Alternating Bit Protocol with Timeouts

… 4.3 – Número de Sequência

- 02 tipos de msgs. “mesg” e “ack” com os seguintes formatos:

- {mesg, data, sequence number} - … “mesg:o:s” indica mensagem “mesg” com o campo de dados “o” e nro. de seq. “s”;
- “s” indica ao transmissor que armazene a msg. com último nro. de sequência; “r” mantém o último nro. de sequência recebido;

- “e” indica o último nro. de sequência esperado no receptor; “a” armazena o último nro. de seq. atualmente recebido no receptor.
- {ack, sequence number}

… 4.3 – Número de Sequência

Considere o que acontece se a msg. de “ack” está tão atrasada que o transmissor sofre “timeout” e retransmite a última msg.

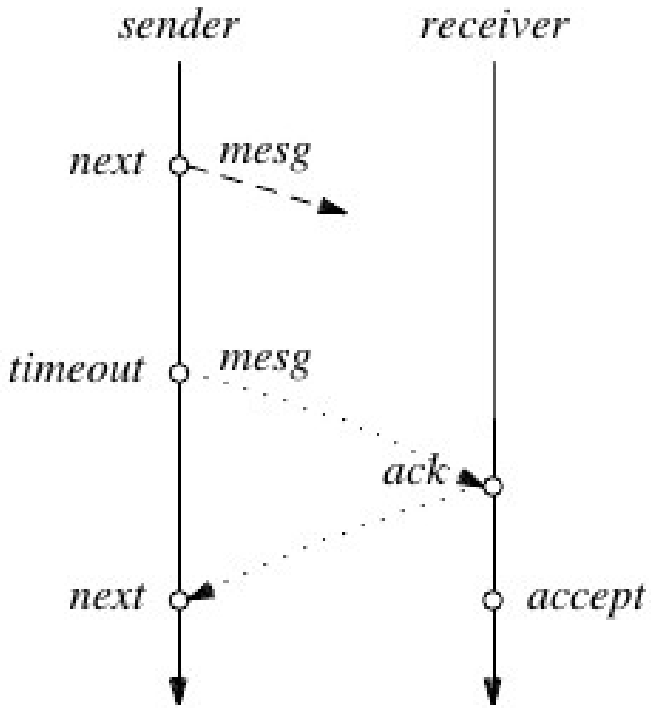


Figure 4.10 — Time Sequence Diagram of Error

… 4.3 – Número de Sequência

- “duplicação” e “reordenação” - presentes em redes de datagrama, isto é, em redes que não há circuito lógico ou circ. virtual.
  - “redes de datagrama” - são aquelas nas quais primitivas podem usar diferentes rotas para alcançar a(s) entidade(s) parceira(s);
  - “circuito lógico” - são canais que de algum modo têm mapeamento no meio físico (por ex. circuitos de linhas para telefones).
  - “circuito virtual” - são canais que têm as características de comunicação de circuito lógicos mas não existem de facto (obtidas por configuração).

- “solução” - codificar a primitiva na ordem original com um nro. de sequência com espectro grande o suficiente para numerar qualquer quantidade de primitivas.

… 4.3 – Número de Sequência

- … normalmente o número de sequência segue acoplado ao cabeçalho da primitiva (quanto maior … maior a quantidade de mensagens sem que o nro. de sequência seja reutilizado).

- … se considerarmos primitivas de 15Kb, com um número de sequência de 16 bits, e um canal de 100Mbps, este sistema vai conseguir funcionar por aproximadamente 9,5 segundos
  - … 100 * 10^6 bps / 15.000 bits = 6666,67 primitivas por segundo, ou seja, cada primitiva irá utilizar um nro. de sequência;
  - … considerando que tem-se 16 bits para os nro. de sequência, então temos 65536 nro. de sequência possíveis;
  - … logo, basta dividir 65536 / 6666,67 = 9,83 segundos serão gastos para utilizarmos todos os nro. de sequência possíveis.

… 4.3 – Número de Sequência

- “problema” do limite do nro. de sequência desaparece se limitar-mos o máximo número de primitivas que estejam em trânsito através do crédito concedido ao transmissor;
- … gama de possíveis números de sequência tem de ser maior que os créditos máximos negociados, permitindo assim ao receptor distinguir primitivas duplicadas de originais.

… 4.3 – Número de Sequência

- Suponha “M” números de sequência disponíveis e “W” números de créditos iniciais de primitivas, então:
  - “M > W” - evita nro. de sequência fora de ordem sem que o mesmo nro. de sequência se repita na janela.

- Que estruturas/mecanismos utilizar para este controle ??
  - … transmissor deve contabilizar cada primitiva pendente - “outstanding primitives” dentro da janela corrente;
  - … para “acks” individuais é necessário uma estrutura de dados que mantenha o estado, o nro. de sequência e a própria primitiva;
  - … (continua)

… 4.3 – Número de Sequência

- Suponha “M” números de sequência disponíveis e “W” números de créditos iniciais de primitivas, então:
  - “M > W” - evita nro. de sequência fora de ordem sem que o mesmo nro. de sequência se repita na janela.

- Que estruturas/mecanismos utilizar para este controle ??
  - … uso de vetores para controle de quais msgs já foram transmitidas, tendo ou não sido reconhecidas;
  - … se uma msg. com nro. de sequência “s” foi transmitida mas ainda não foi reconhecida, então “busy[s] = true”;
  - … já “store[s] = primitiva” .. mensagem com nro. de sequência “s” já foi transmitida;
  - … inicialmente, todos os elementos do vetor “busy” = “true”.

… 4.3 – Número de Sequência

- Transmissor – modularizado com 03 procedimentos:
  - Transmissão de Primitivas;
  - Processamento dos “ACKS”;
  - Retransmissão de Primitivas.

- … além dos parâmetros “W” e “M”, são necessários mais 4 parâmetros (com valor inicial zero):
  - “s” - nro. de sequência da próxima primitiva a ser enviada;
  - “window” - número de primitivas pendentes (não reconhecidas);
  - “n” - nro. de sequência da primitiva mais antiga sem reconhecimento;
  - “m” - nro. de sequência da última primitiva reconhecida.

… 4.3 – Número de Sequência

- “Transmissão de Primitivas” - … transmissor rastreia toda mensagem pronta para ser enviada dentro da janela;
- ... para cada primitiva transmitida, incrementa-se o número de primitivas pendentes de “acks” - “window++”;
- … primitiva transmitida (o) é armazenada na posição “s”, onde “s” corresponde ao número de sequência - “store[s] = o”
- … busy[s] = true indica que msg. com nro. de sequência “s” foi enviada mas ainda não foi reconhecida;
- … incremento de “s” em módulo “M”, ou seja, “s” = “(s+1) % M”, para obter o próximo nro. de seq. a ser utilizado.

… 4.3 – Número de Sequência

- “Processamento de Acks”
- … atualiza-se o número (m) do último reconhecimento;
- … recebe-se um reconhecimento e libera-se o buffer atualmente utilizado pela primitiva reconhecida - busy[m] = false.

- “Retransmissão de Primitivas” … window > 0
- … percorre-se a estrutura de envio periodicamente - “timeout period”, se “busy[n]==true”, então reenvia primitiva “store[n]”
- … se “busy[n]==false”, então “window--” e “n = (n+1)%M”.

… 4.3 – Número de Sequência

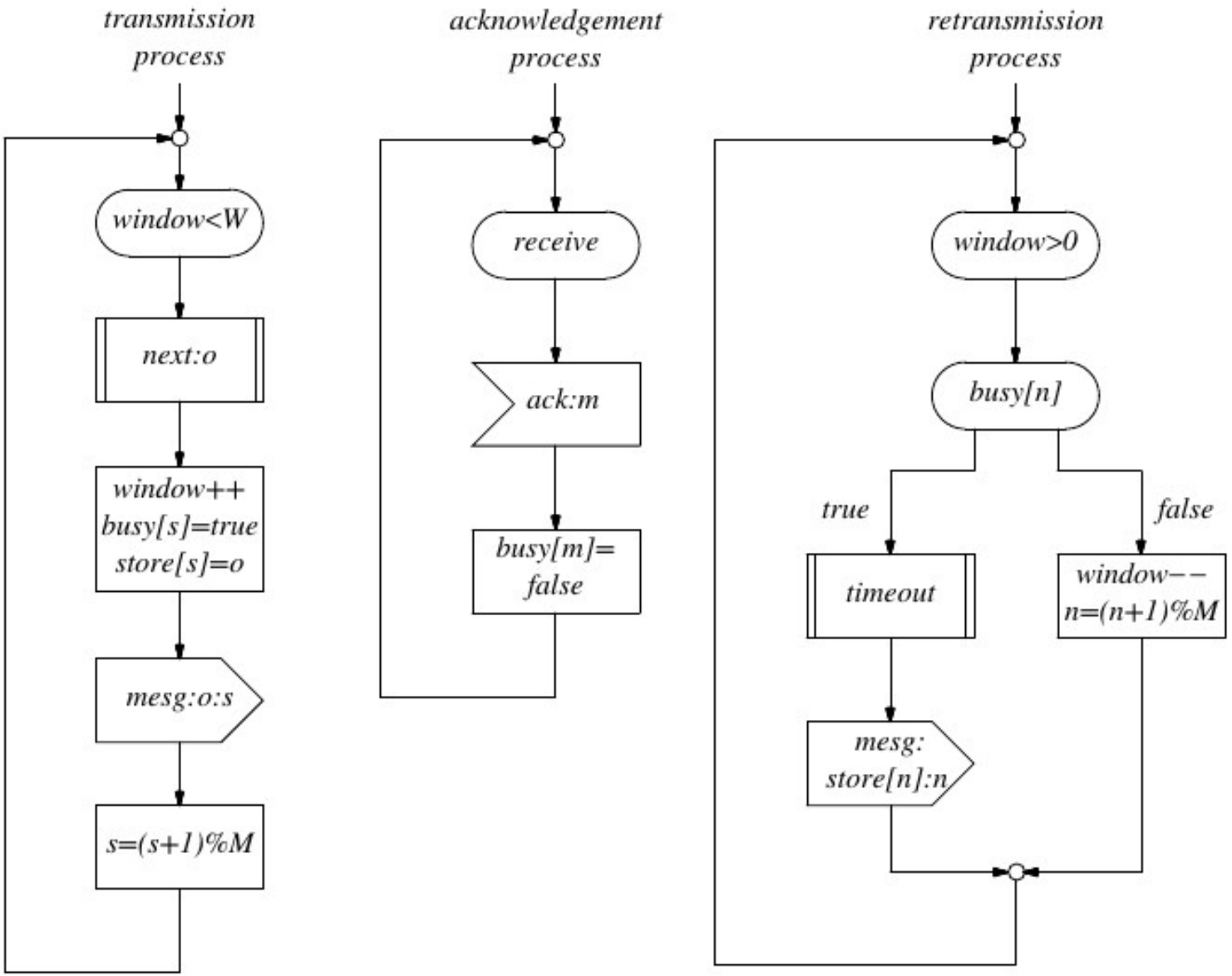


Figure 4.11 — Sender Processes, Sliding Window Protocol

… 4.3 – Número de Sequência

- Receptor - normalmente modularizado e com 02 procedimentos
  - “Recepção de Primitivas”;
  - “Aceitação de Primitivas”.

- “Recepção de Primitivas” - ... além dos parâmetros W e M, são necessários mais 02 parâmetros e 02 vetores:
  - “m” - número de sequência da última primitiva recebida;
  - “p” - número de sequência da primitiva sendo esperada;
  - “buffer[M]” - conteúdo das mensagens recebidas;
  - “recvd[M]” - representa números de sequência recebidos, mas não processados como “true”, ou “false” caso contrário.

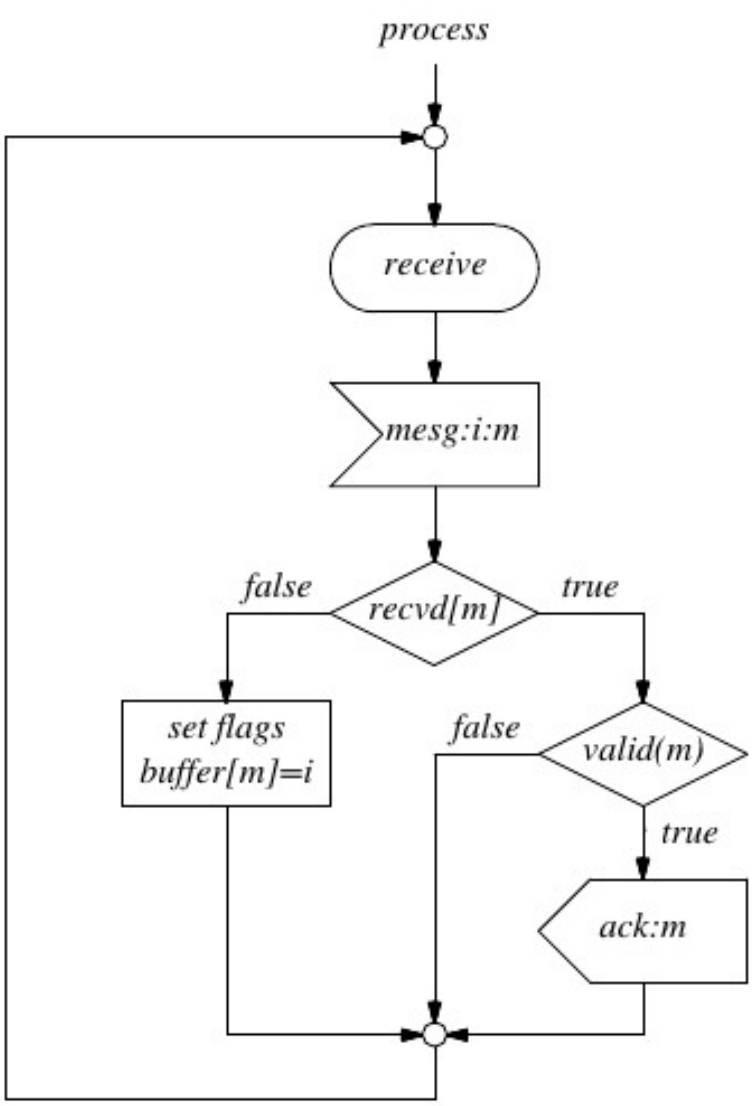
… 4.3 – Número de Sequência

- “Aceitação da Primitiva” - para serem aceitas, as primitivas devem ser aceitas primeiro para então serem reconhecidas, pois se o forem correm o risco de usar “buffers” fora do espaço definido.
- … recepção consome tempo e interpretação da primitiva, decisão do que fazer, alocação de memória, e até encaminhamento.
- … processo de aceitação pode ser mais lento que o de recepção, o que sugere o reconhecimento somente depois da aceitação.

… 4.3 – Número de Sequência

- … utiliza-se o vetor “recvd[M]” para armazenar o nro. de sequência de msgs. que foram recebidas, mas ainda não aceitas;
- … utiliza-se o vetor “buffer[M]” para armazenar o conteúdo destas msgs, ou seja, recebidas mas ainda não aceitas;
- … adicionalmente, faz-se necessário uma variável extra “p” que é o nro. de sequência da próxima msg. que será aceita.

… 4.3 – Número de Sequência



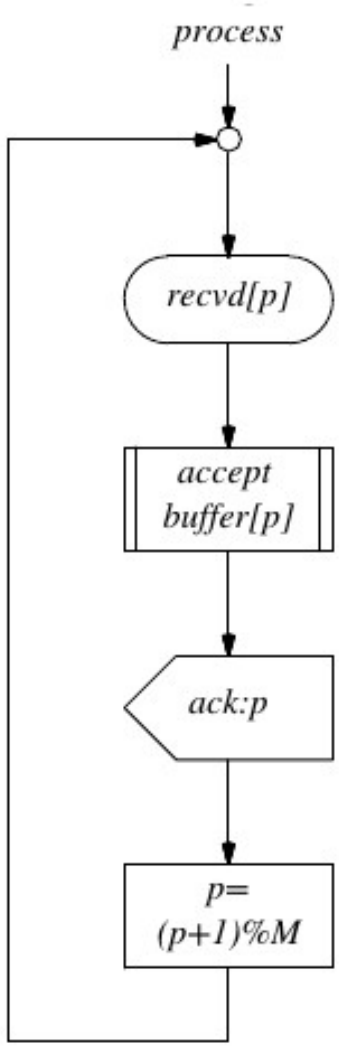


Figure 4.12 — Receiver Processes, Sliding Window Protocol

… 4.3 – Número de Sequência

- … qdo o receptor recebe uma primitiva, cabe ao mesmo verificar se a posição equivalente está livre “recvd[m] == false”, e então:
  - “recvd[m] = true”;
  - “recvd[(m – W + M) % M] = false” ou “recvd[(m - W) % M] = false”

- … duplicação é percebida ao verificar o estado da posição sendo recebida - “recvd[m] == true” → indicação de duplicação.
- Há 02 (dois) motivos para primitivas duplicadas:
  i. primitiva foi recebida mas ainda não foi reconhecida;
  ii. primitiva foi recebida, reconhecida mas o reconhecimento ainda não chegou ao transmissor (somente neste deve ter re-reconhecimento!)

… 4.3 – Número de Sequência

- … ao analisar o valor corrente do parâmetro “p”, pode-se afirmar se o caso “ii” foi ou não devidamente tratado.
- Uma primitiva recebida é considerada válida, dependendo de:
  - … se o número de sequência não é do tipo módulo, ou seja, não é janela deslizante, o teste é simplesmente “valid(m) = m < p”;
  - … se considerarmos o efeito do módulo “M”, “p” será sempre menor do que “M”, então: “valid(m) = (0 < p - m ≤ W) || (0 < p + M - m ≤ W)

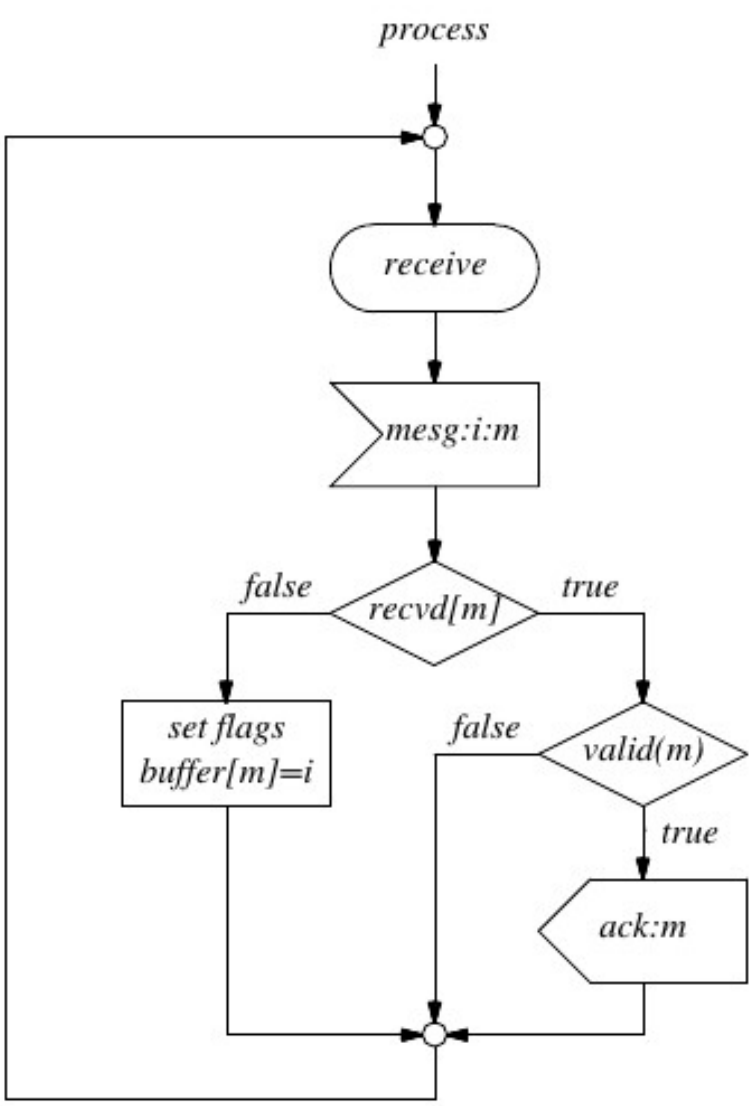
- Obs.: … se “s” é uma primitiva recebida que se encontra no buffer, então nenhuma ação é tomada pelo módulo de recepção.

… 4.3 – Número de Sequência

- Nota: protocolo de janela deslizante garante que a retransmissão de msgs. não trata nro. de sequência maior que “W” e nem menor que a última msg. que foi reconhecida.

- ... estrutura de dados é percorrida e o processo de aceitação (dependendo da aceitação) gera um Ack ou um NAck.
- … processa e aceita (ou não aceita) “buffer[p]”;
- … gera um Ack (ou NAck) para o número de sequência “p” para na sequência atualizar “p”, ou seja, “p = (p+1) % M“

… 4.3 – Número de Sequência



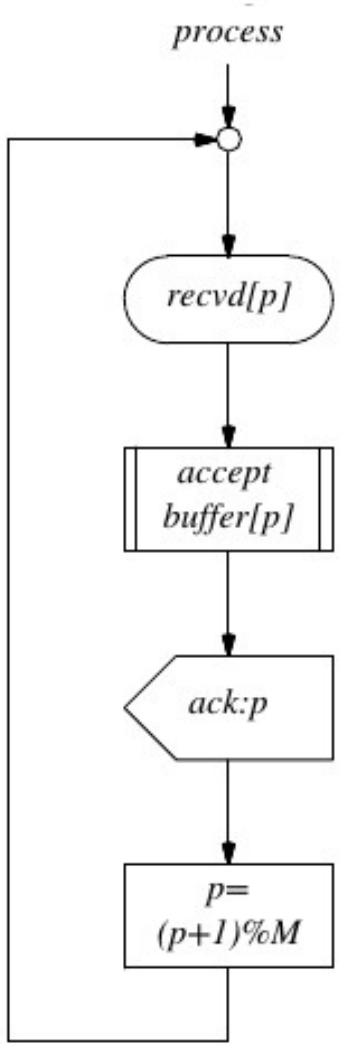


Figure 4.12 — Receiver Processes, Sliding Window Protocol

… 4.3 – Número de Sequência

- Seja “M” nros de sequência disponíveis e “W” nros de créditos de primitivas, então, qual o “Tamanho Máximo da Janela” ??

- “Tamanho Máximo da Janela” - dado “M” e sabendo-se que “M > W”, então qual o máximo “W” para otimizar o canal ?
- … se todas as mensagens que chegam fora de ordem são simplesmente rejeitadas, então a resposta é “M-1”.
- … se um nro. de sequência não for reutilizado até que a respectiva primitiva seja reconhecida, então, o cenário é outro;
  - … nestes casos, isto significa que “W” não pode exceder “M/2”:

… 4.3 – Número de Sequência

- Se um nro. de sequência não for reutilizado até que a respectiva primitiva seja reconhecida, então, “W” não pode exceder “M/2”.
- Considere “H” o maior número (%M) que o receptor reconheceu, então podemos ter 02 (dois) cenários:
  - … pior caso, transmissor recebeu apenas um “ack”;
  - … melhor caso, transmissor recebeu todos os “acks”.

… 4.3 – Número de Sequência

- Para o pior caso, o transmissor pode decidir retransmitir todas as “W-1” primitivas e a própria “H”-ésima primitiva:
  - … primitiva mais antiga que poderia ser retransmitida poderia ser [H – (W – 1)] % M, ou (H – W + 1) % M

- Para o melhor caso, podem ser retransmitidas até “W” msgs. que sucederam a msg. com nro. de sequência “H”.
  - … primeiras “W-1” mensagens podem ter sido perdidas no canal, então a msg. com nro. de seq. (H + W) % M é a primeira nova msg. a chegar.
  - ... maior nro. de seq. que suceda H tem de ser distinguível do menor nro. de seq. com potencial de ser retransmitido (que preceda H);
  - … isto significa que “M > 2W - 1” ou a janela de tamanho máximo será “W” igual a M/2 (W = M/2).

4.4 – Reconhecimento Negativo

- “reconhecimento” … até agora foi utilizado como método de controle de fluxo, mas não como controle de erro.
  - … se a msg. é perdida ou danificada e não é reconhecida, a ausência de reconhecimento positivo irá eventualmente causar o “timeout” no transmissor e, por conseguinte, a retransmissão da mensagem.

- “reconhecimento negativo” - … pode-se amenizar o problema, não obstante não se consegue eliminá-lo completamente.
  - … nack - usado pelo receptor qdo o mesmo recebe uma mensagem que foi danificada no canal durante a transmissão.
  - … qdo o transmissor recebe um “ack” negativo, ele sabe que deve retransmitir a msg. correspondente, sem ter que esperar pelo “timeout”.

… 4.4 – Reconhecimento Negativo

- Fig. 4.13 e Fig. 4.14 mostram uma extensão para o Protocolo de Bit Alternante já discutido na Fig. 4.9 que constitui no reconhecimento negativo (sem nro. de sequência).

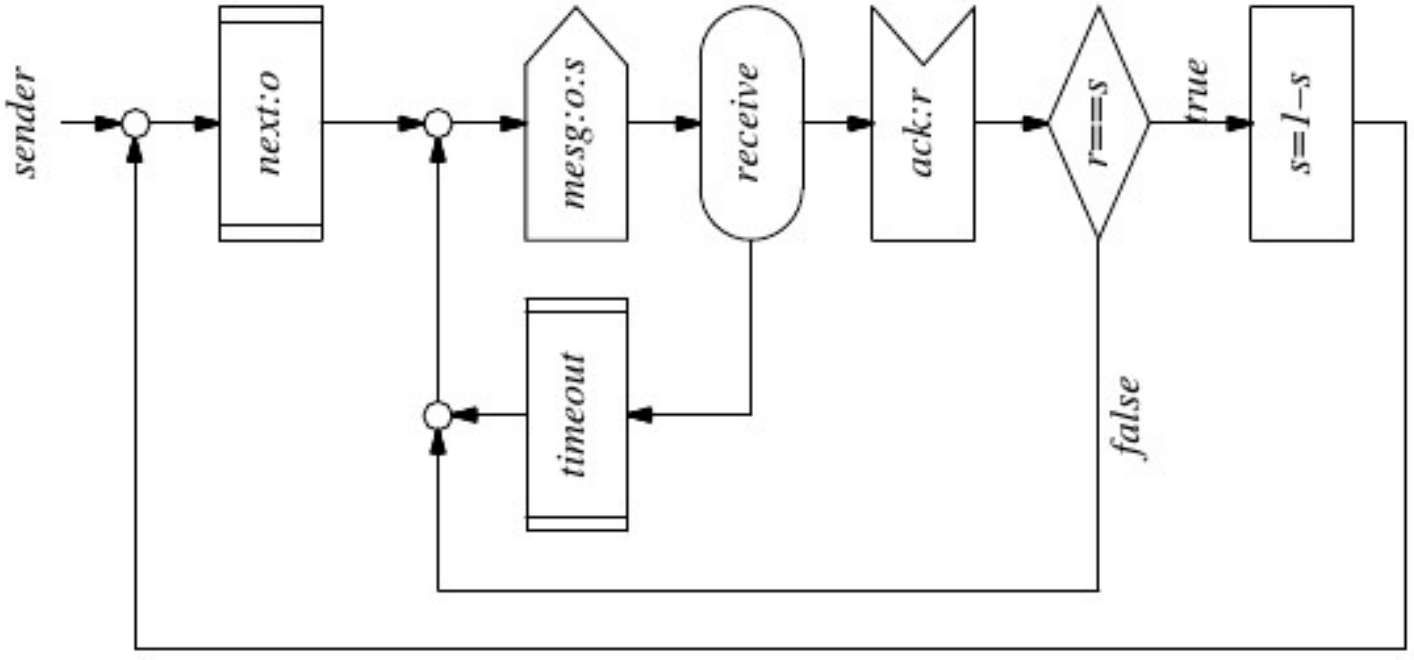


Figure 4.9 — Alternating Bit Protocol with Timeouts

… 4.4 – Reconhecimento Negativo

- … Fig. 4.13 e Fig. 4.14 mostram uma extensão para o Protocolo de Bit Alternante já discutido na Fig. 4.9 que constitui no reconhecimento negativo (sem nro. de sequência).

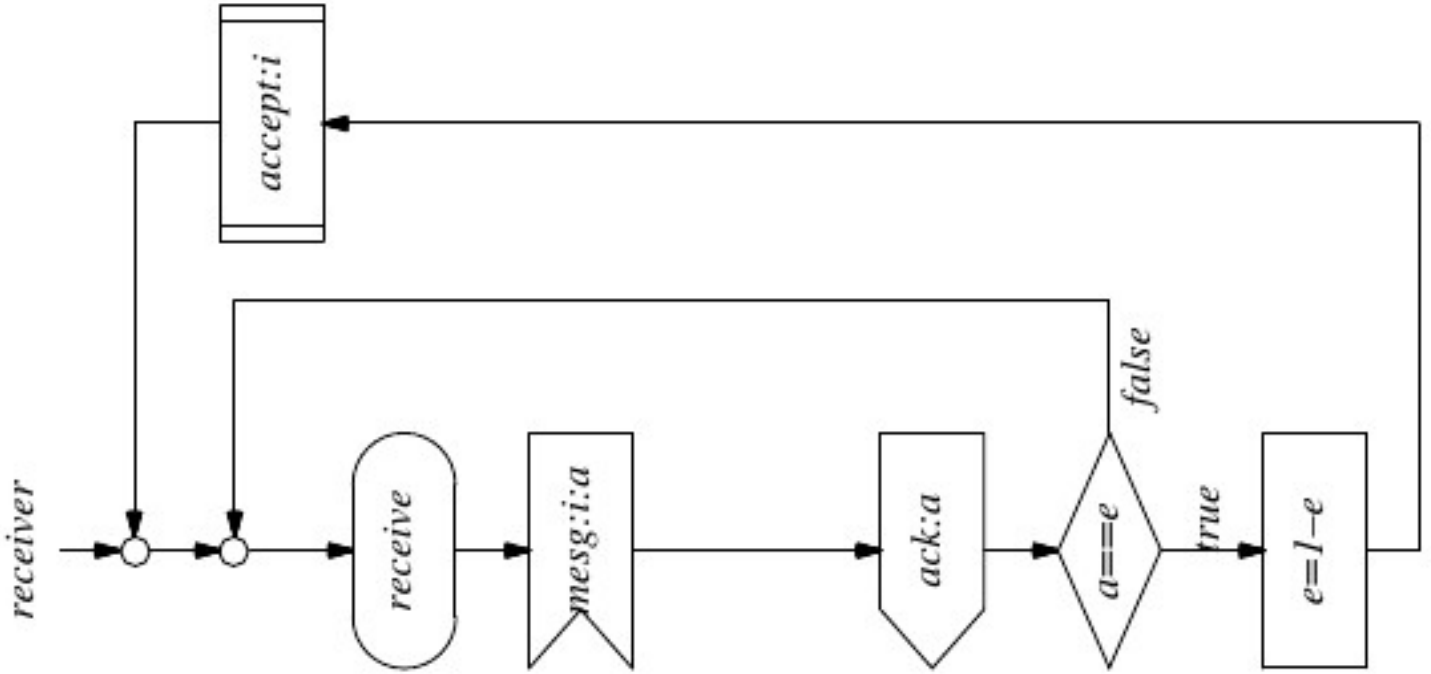
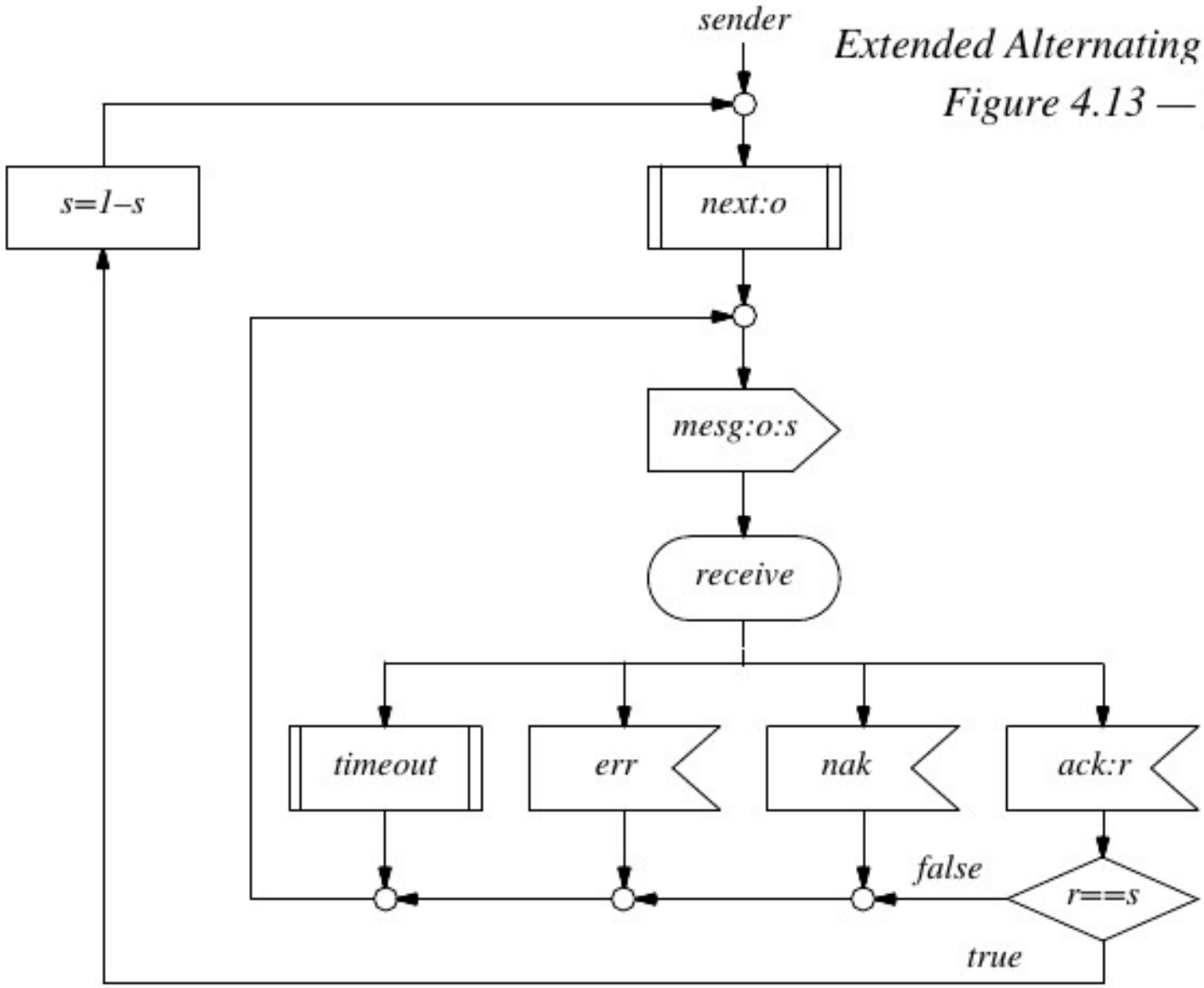


Figure 4.9 — Alternating Bit Protocol with Timeouts

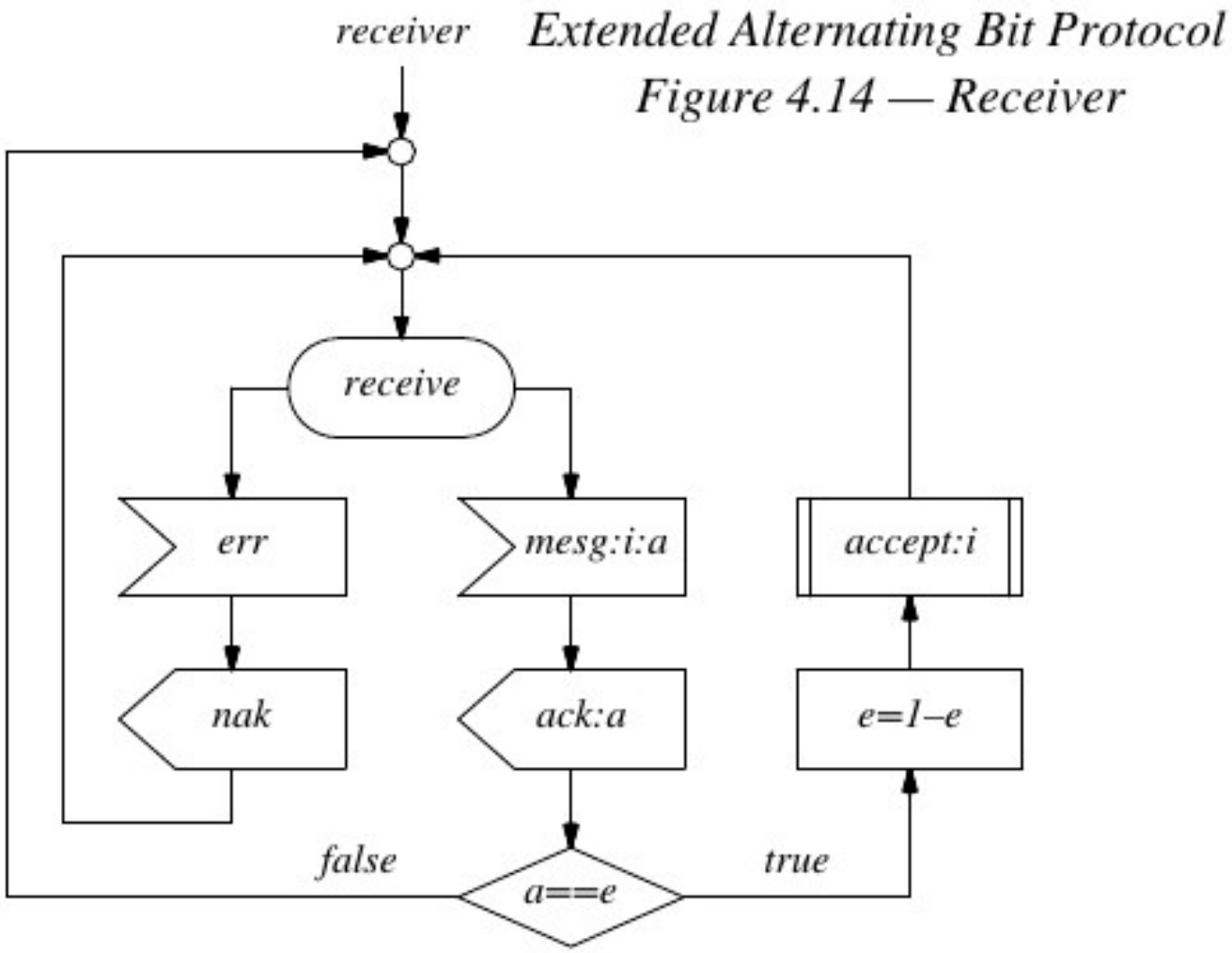
… 4.4 – Reconhecimento Negativo



Extended Alternating Bit Protocol

Figure 4.13 — Sender

… 4.4 – Reconhecimento Negativo



… 4.4 – Reconhecimento Negativo

- “Automated Repeat reQuest” ARQ – método que utiliza reconhecimentos para controlar a retransmissão de mensagens
  - Stop-and-Wait ARQ
  - Selective Repeat ARQ
  - Go-back-N Continuous ARQ

- e.g., .. Protocolo Ping-Pong estendido com reconhecimento negativo, pode ser classificado como “Stop-and-Wait ARQ”.
- … após cada mensagem enviada, o transmissor deve esperar por um NAck ou Ack, ou mesmo “timeout”.

… 4.4 – Reconhecimento Negativo

- … após cada mensagem enviada, o transmissor deve esperar por um NAck ou Ack, ou mesmo “timeout”.

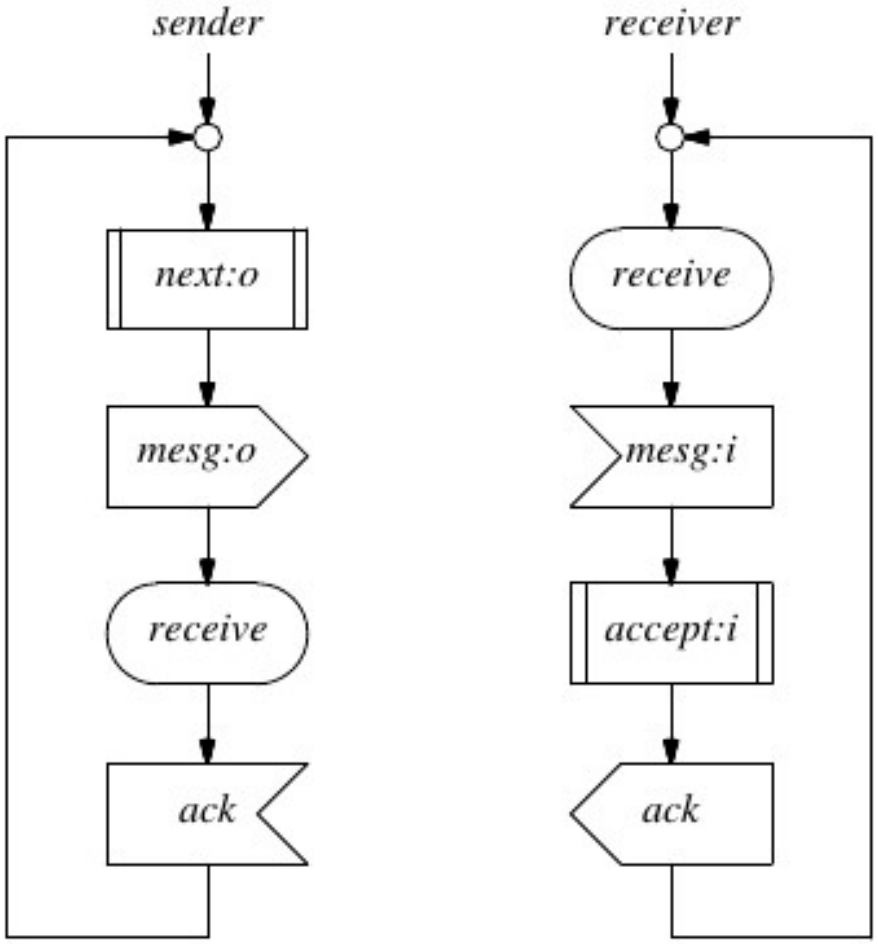


Figure 4.4 — Ping-Pong Protocol

… 4.4 – Reconhecimento Negativo

- Uso de reconhecimento no Protocolo de Janela Deslizante (Fig. 4.11 e 4.12) pode ser classificado como “Selective Repeat ARQ”

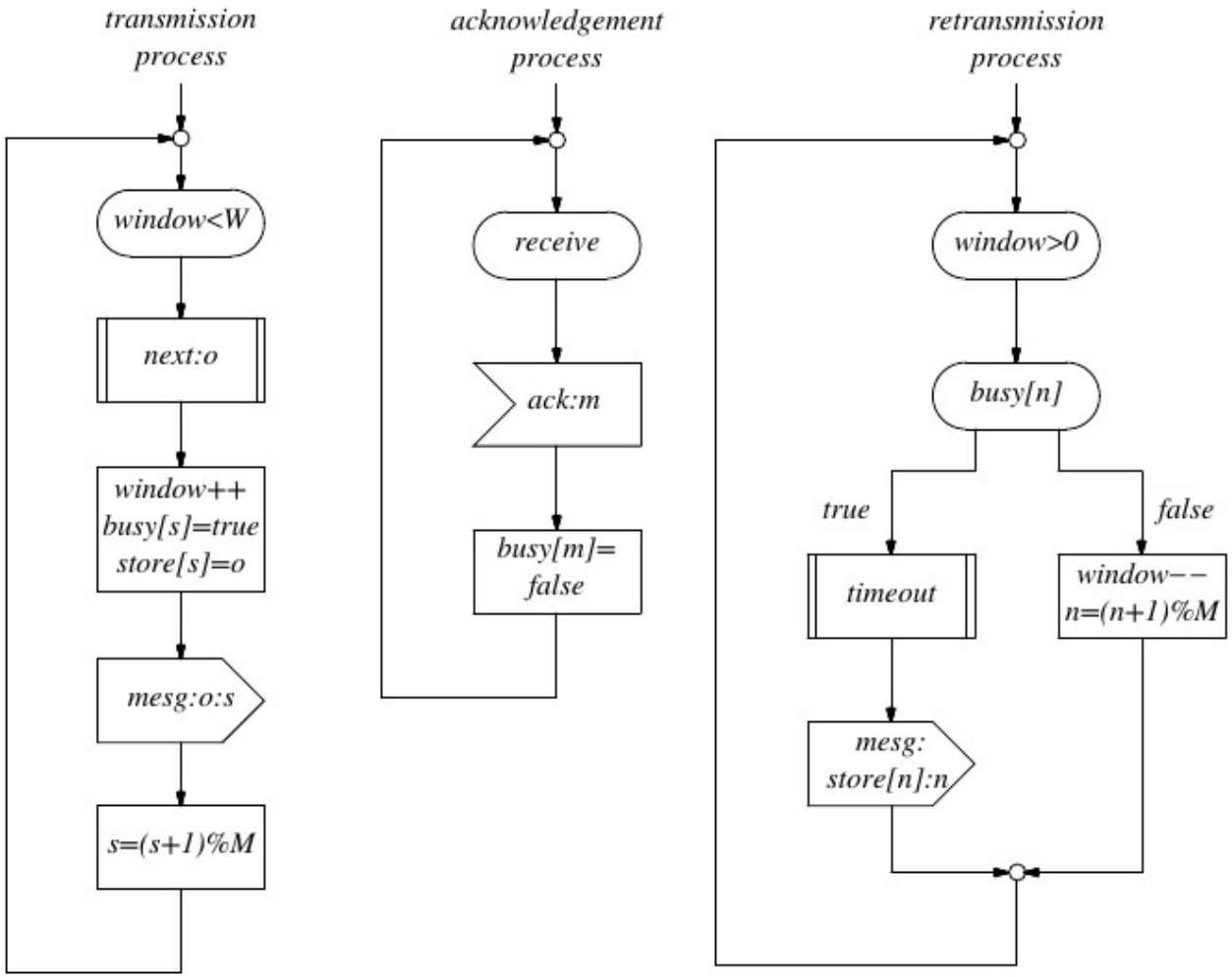


Figure 4.11 — Sender Processes, Sliding Window Protocol

… 4.4 – Reconhecimento Negativo

- Fig. 4.11 implementa método de repetição seletiva “one-at-a-time” onde somente a msg. não reconhecida mais velha é retransmitida

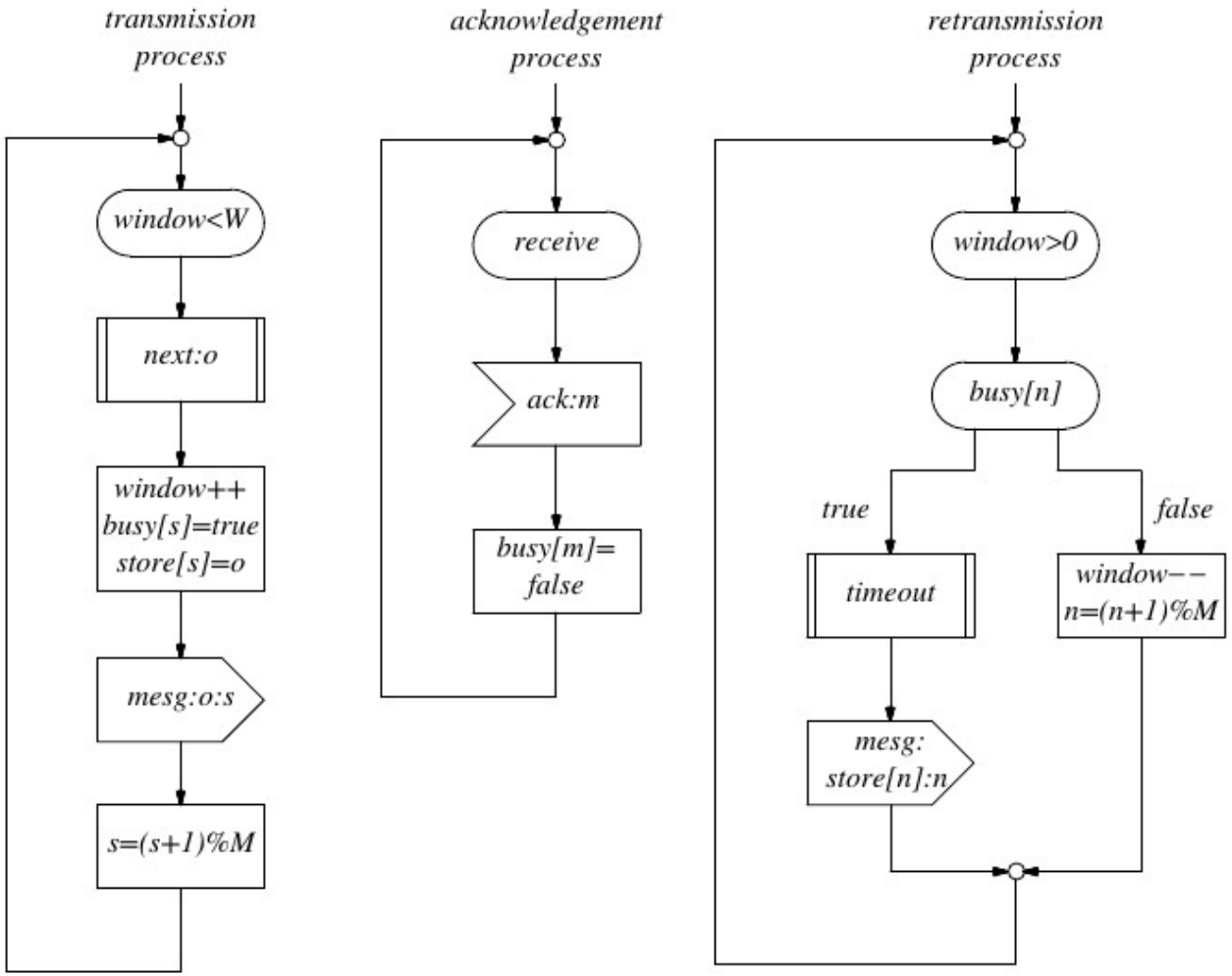


Figure 4.11 — Sender Processes, Sliding Window Protocol

… 4.4 – Reconhecimento Negativo

- Entretanto, qualquer msg. que gere um “ack” negativo ou “timeout” pode ser retransmitida, independente de outra mensagem.

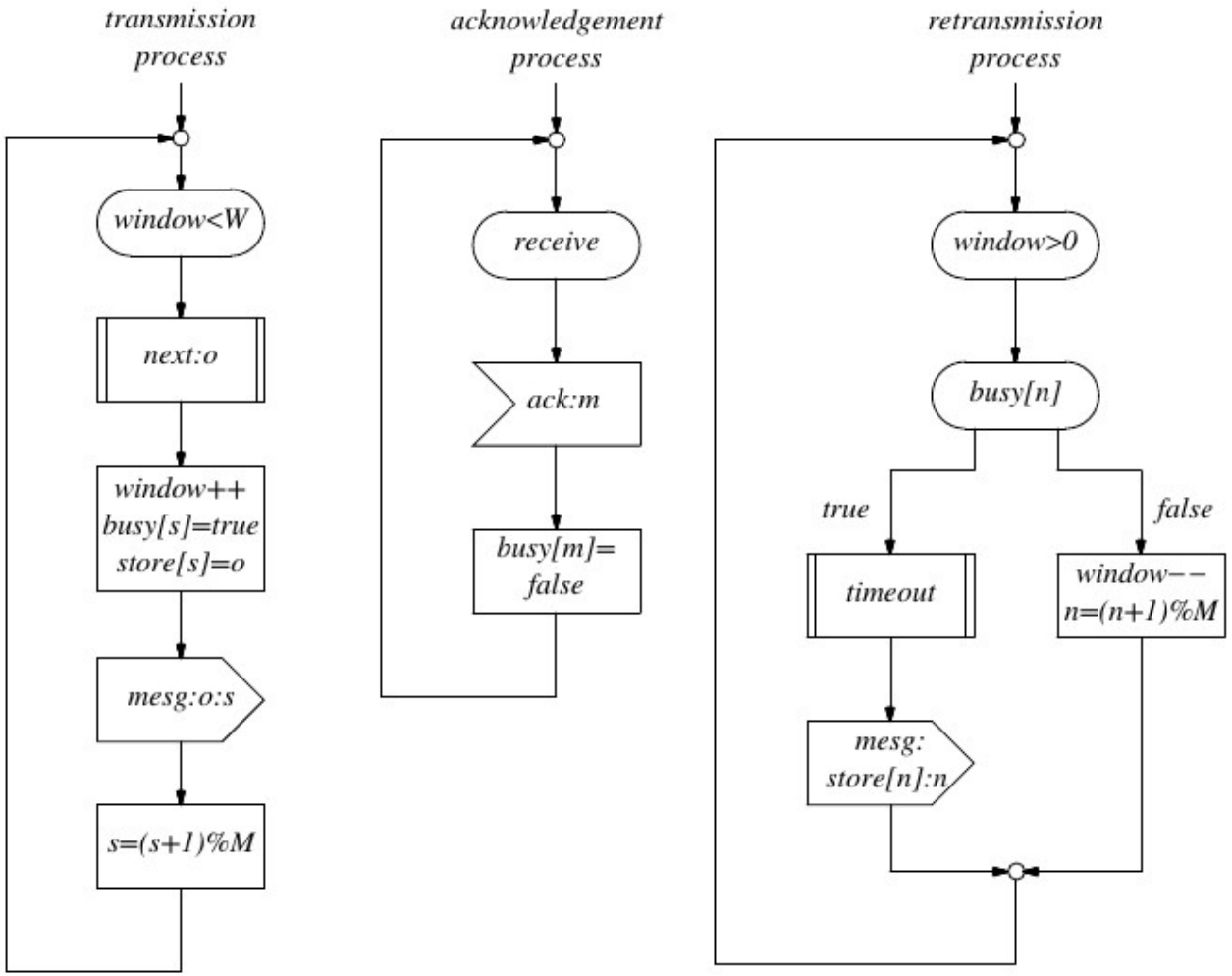
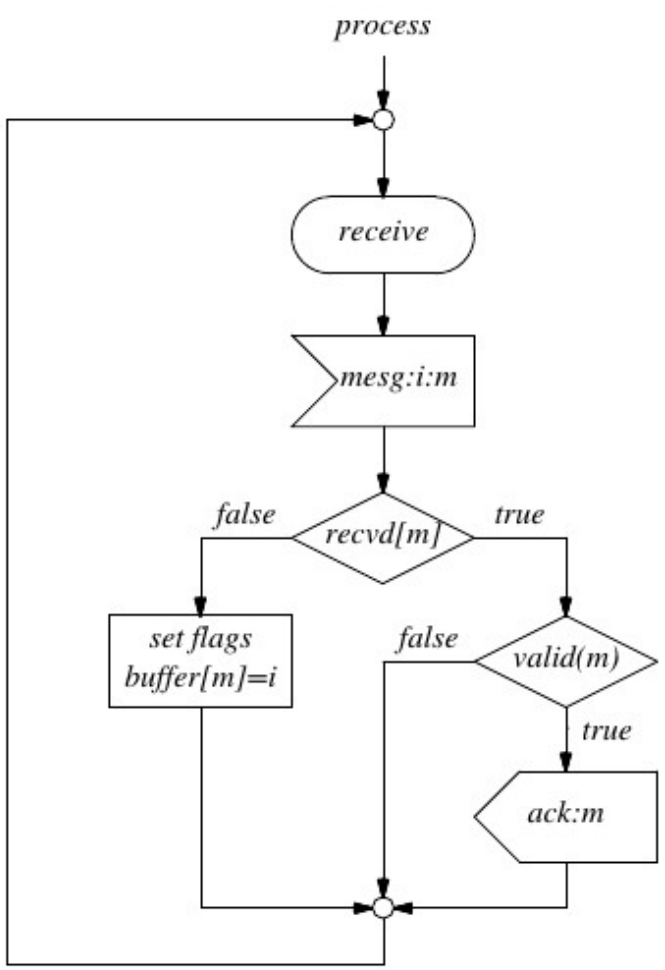


Figure 4.11 — Sender Processes, Sliding Window Protocol

… 4.4 – Reconhecimento Negativo

- … generalização do método é repetição seletiva contínua.



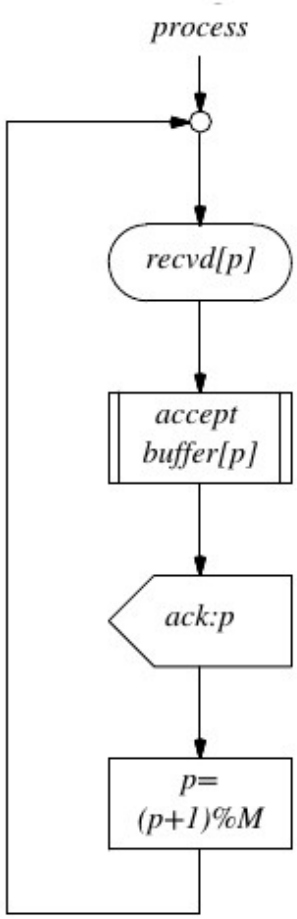


Figure 4.12 — Receiver Processes, Sliding Window Protocol

… 4.4 – Reconhecimento Negativo

- “Go-back-N Continuous ARQ” - pode ser implementado no Protocolo Bit Alternante estendido;
  - … basta que o transmissor retransmita a msg. corrompida e todas as mensagens subsequentes já enviadas.
  - … neste caso o projeto do receptor pode ser simplificado, p.ex., “acceptor” da Fig. 4.12 pode ser removido e o “buffer” torna-se desnecessário.

- “Go-back-N” … receptor recusa aceitar todas as msgs. que chegam fora de ordem e espera por aquelas na sequência correta, ou seja, não se reconhece msgs. fora de ordem.
  - … reconhecimento com nro. de sequência pode ser entendido como reconhecimento de todas as msgs. até àquela incluindo “s” - também chamado de “reconhecimento cumulativo”.

… 4.4 – Reconhecimento Negativo

- “Go-back-N” ou “Selective Repeat” com variações podem reduzir o nro. de msgs. de reconhecimentos individuais que são enviadas do receptor para o transmissor - “reconhecimento em bloco”
- … neste caso, reconhecimento pode especificar a faixa de nros. de sequência que foram recebidas corretamente;
- “block acknowlegment” - pode enviar periodicamente pedido do transmissor e, assim, pode ser visto como uma forma estendida do “reconhecimento cumulativo”.

4.5 – Evitando Congestionamento

- Há 02 razões principais para incluirmos controle de fluxo nos protocolos: sincronização e controle de congestionamento.
  - … até este momento, praticamente ignoramos como evitar o congestionamento e nos concentramos na sincronização fim-a-fim!

- “ponto não discutido” - para um dado enlace, qual o tamanho da janela bem como a correspondente faixa de nrs. de sequência que se deve escolher ?
  - … é relativamente fácil estabelecer limites superiores no tamanho da janela, pois aumentar a partir de um ponto pode não mais melhorar a vazão caso o canal já esteja completamente saturado.

… 4.5 – Evitando Congestionamento

- Considere t_{prog} = 0.5 seg. do transmissor para o receptor e t_{prog} = 0.5 seg. do receptor para o transmissor, ou seja, o canal será saturado pelo transmissor se a transmissão perdurar por 1 seg.
  - … é relativamente fácil estabelecer limites superiores no tamanho da janela, pois aumentar a partir de um ponto pode não mais melhorar a vazão caso o canal já esteja completamente saturado.

… 4.5 – Evitando Congestionamento

- Considere t_{trans} = S bps, ou seja, transmissor deve ser capaz de transmitir S bits antes que seja necessário verificar os “acks”;
  - … se cada msg. contém M bits, o melhor tamanho de janela é “S / M”, naturalmente, podemos fazer algo como “M < S” de modo a transmitir mais mensagens de uma única vez.

- “tamanho da janela > S / M” → perda de tempo ?!
  - … é uma perda de tempo pois no momento que a última msg. da janela é transmitida, o “ack” para msg. válida mais velha deve ter chegado, ou se não chegou, é o momento de considerar a retransmissão da mensagem.

… 4.5 – Evitando Congestionamento

- … necessário considerar o tipo de cálculo proposto, pois o mesmo reduz o problema do controle de fluxo para uma questão no nível do enlace, enquanto ignora a rede que contém o enlace de dados;
- Tendo por base um rede com 02 enlaces, há 02 caminhos na definição de um protocolo de controle de fluxo:
  - “Hop-by-Hop” ou também chamdo “Node-to-Node”;
  - “End-to-End” - não há participação dos nós intermediários.

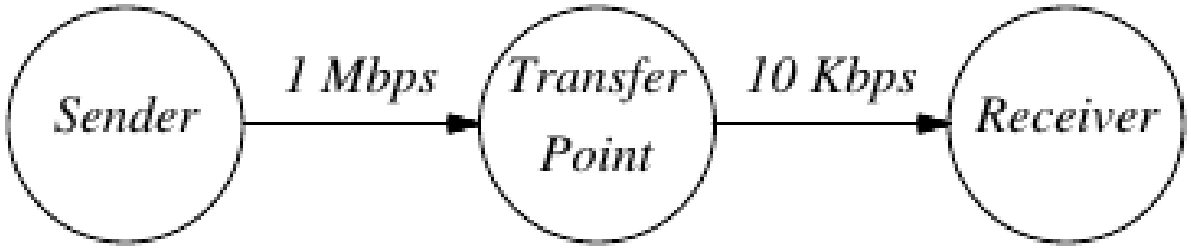


Figure 4.15 — Two-Hop Link

… 4.5 – Evitando Congestionamento

- “Hop-to-Hop Protocol” - tamanho da janela é calculado separada-mente para cada enlace de modo que possa ser saturado;
- … primeiro enlace é 100 vezes mais rápido que o segundo, assim, se bem sucedidos na saturação dos 02 canais, ter-se-á um grande problema “versus” desempenho no uso de recursos;
- … dados chegam 100 vezes mais rápido do que podem ser repassados para o receptor e, independente do espaço em “buffer”, pode ser sobrecarregado → perda de mensagens;
- … única forma do “transfer point” controlar o transmissor é recusar as mensagens de “acks”, não obstante o transmissor irá saturar o canal e ser bem sucedido (retransmissões ou dados novos).

… 4.5 – Evitando Congestionamento

- “solução” - controle de fluxo que retire o máximo do uso dos recursos em separado, ou seja
  - “buffer space” nos nós da rede;
  - “bandwidth” do enlaces que conectam os nós da rede.
- … esta abordagem falha nos 02 pontos, uma vez que gasta espaço de “buffer” no nó “transfer point”, potencialmente bloqueando outros tráfegos que passem pelo nó;
- … também gasta “bandwidth” pois gatilha um dilúvio de retrans-missões no enlace do transmissor para o “transfer point”;
- … uso ótimo dos 02 enlaces pode ser atingido se o transmissor encaminha dados na velocidade do enlace de menor capacidade.

… 4.5 – Evitando Congestionamento

- “End-to-End Protocol” - neste caso, a capacidade fim-a-fim da rede é igual a capacidade do menor enlace e, assim, o tamanho da janela pode ser escolhido apropriadamente.

- Obs.: Em Redes mais complexa, não há esperança que o transmissor possa facilmente predizer onde se encontra o enlace de menor capacidade no caminho até o receptor.
- …. em redes maiores, a capacidade do enlace de dados depende não apenas do “hardware”, mas também do nro. de usuários;

… 4.5 – Evitando Congestionamento

- Em redes maiores, a capacidade do enlace de dados depende não apenas do “hardware”, mas também do nro. de usuários;
- e.g., Se 10 usuários iniciam a transf. de arquivos no mesmo enlace da rede, este enlace pode subtamente se transformar no enlace de menor capacidade para todos eles.
- “mais seguro a fazer” - derivar o tamanho máximo da janela para toda a rede considerando o enlace de menor capacidade.

… 4.5 – Evitando Congestionamento

- “dynamic window” - controle de fluxo com janela dinâmica permite que o protocolo seja auto-adaptável.
- … método simples e comum de se usar é forçar o transmissor a diminuir o tamanho da janela toda vez que ocorrer “timeout” (e.g. caso de retransmissão automática);
- … considerando que o “timeout” não mais ocorra, o transmissor pode aumentar gradualmente o tamanho da janela até o seu valor máximo (busca-se o máximo de transferência a todo instante).

… 4.5 – Evitando Congestionamento

- Há diferentes filosofias quanto a precisão dos parâmetros de modo que esta técnica seja utilizada, dentre elas:
- … diminuir a janela de 1 toda vez que “timeout” ocorre e, aumentar de 1 toda vez que se recebe um “ack” positivo;
- … diminuir a janela pela metade do tamanho corrente a cada “timeout” e aumentar de 1 msg. quando da ocorrência de “N” “acks” positivos recebidos;
- … diminuir a janela para o seu valor mínimo de 1 na ocorrência do “timeout” e, aumentar a janela de 1 quando da ocorrência de “N” “acks” positivos recebidos.

… 4.5 – Evitando Congestionamento

- Suponha que uma primitiva leva 0,5 segundos para ir do transmissor ao receptor e o mesmo tempo para voltar o Ack:
  - … transmissor satura o canal se ele tem algo para trasmitir que toma 1s.

- Se a taxa do canal é de S bps, o transmissor pode transmitir S bits para então se preocupar com reconhecimento.
- Se cada primitiva tem “N” bits, a melhor janela é “S/N”, ressaltando-se a importância de garantir que “N < S”
  - … janelas maiores do que isto representam desperdício!