



1 Descrição do protocolo

1.1 Diagrama de estados para o protocolo RSGama. Diagrama de estados - Protocolo RSGAMA

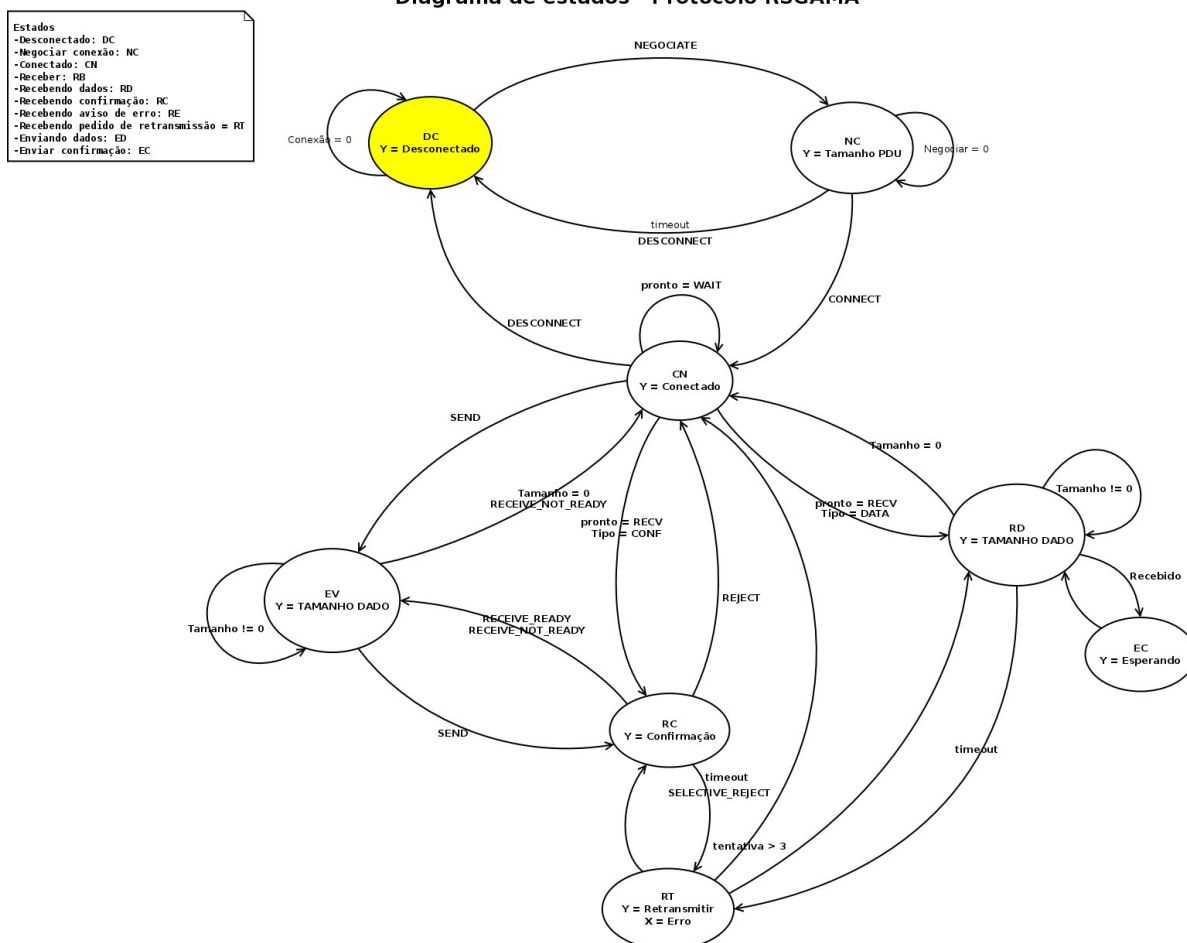


Diagrama de estados. Legenda no canto superior da direita.

1.1.1 Detalhamento do diagrama de estados do protocolo

O estado inicial do protocolo é o de **Desconectado(DC)** e permanece neste até que uma solicitação de conexão seja realizada. O estado de Desconectado é sensível apenas a solicitação de conexão, caso esta ocorra o protocolo transitará para um estado de **Negociação(NC)** da conexão.

No estado de **Negociação(NC)**, busca-se estabelecer um acordo referente ao tamanho da PDU (*Protocol Data Unit*) que será usada. Enquanto esta negociação não for finalizada o protocolo mante-se neste estado, contudo ao término da negociação da conexão o protocolo passará para o estado de **Conectado(CN)**.

No estado **Conectado(CN)** o protocolo ficará aguardando algum pedido de desconexão, de enviar dados ou de receber dados. Se for detectado que um



determinado quadro chegou, o estado é alterado para **Recebendo Dados(RD)**. Se for solicitado um envio de dados, ocorrerá uma transição para o estado **Enviando Dados(EV)**. Por fim se for detectado um pedido de encerramento da conexão, o protocolo é levado do estado Conectado para o estado **Desconectado(DC)**.

Recebendo Dados é o estado que capta os quadros que estão sendo recebidos.

Enquanto houver dados para ser recebidos, este estado é mantido e além disto a cada quadro recebido um quadro de confirmação é enviado. No término da recepção, o mesmo retorna ao estado de **Conectado**.

Enviando Dados é responsável por captar os dados da camada superior e envia-lo no tamanho previamente negociado com a outra ponta da conexão. Enquanto houver dados para ser enviado este estado é mantido, do contrário o protocolo retorna ao estado de **Conectado**.

1.2 Descrição do cabeçalho.

Bits	8	2	1	3	6		20	8
Nome	FLAG	Protocol	TF	Type	Size	Data	Checksum	FLAG

Cabeçalho do protocolo para o caso de um quadro do tipo Control

1.2.1 **FLAG:** Campo composto pela sequência de bit 01111110 e que tem por única finalidade indicar o começo e o fim de um cabeçalho.

1.2.2 **Protocol(Protocolo):** Indica qual protocolo esta sendo usado e por consequência muda a interpretação dos demais campos, os valores possíveis para este campo são:

Código	Protocolo
00	RS-232
01	RS-422
11	RS-485
10	--

1.2.3 **TF - Type of Frame (Tipo Frame):** Indica o tipo de quadro que esta sendo enviado. Basicamente são possíveis duas combinações:

Código	Type of frame
0	Data
1	Control

1.2.4 **Type(Tipo):** Este campo lida com aspectos referentes a cada tipo de quadro. Para quadros que sejam do tipo *Data*(TF = 0) obtém-se o seguinte arranjo:

Bits	1	1	1
Nome	P(0)/F(1)	Window	Size

O bit P/F significa *Poll* (Captação) e *Final*. Quando este bit é igual a 0, significa dizer que ainda há *frames* para receber, do contrário indica que o referido *frame* é o último. O último bit é acrescido ao campo *Size*, isto faz com que o cabeçalho adquira a seguinte forma:



Universidade de Brasília – Campus Gama (FGA)
Especificação do protocolo RSGama

Bits	8	2	1	1	1	1	6		20	8
Nome	FLAG	Protocol	TF	P/F	Window	-	size	Data	Checksum	FLAG

Cabeçalho para um quadro do tipo Data.

Para quadros do tipo Controle, obtêm-se o arranjo seguinte:

Bits	3
Nome	Control

Estes três bits servem para indicar o tipo de controle, basicamente os seguintes controles são possíveis:

Código	Controle
000	RECEIVE_READY: Envia quadro de confirmação e indica que esta pronto para receber novos quadros.
001	REJECT: Ocorreu um erro no quadro recebido.
010	RECEIVE_NOT_READY: Envia quadro de confirmação e indica ao transmissor para não enviar mais quadros.
011	SELECTIVE_REJECT: Solicita retransmissão do quadro defeituoso.
100	DISCONNECT: Encerra a conexão.
101	CONNECT: Estabelece a conexão.
110	SEND: Envia quadro.
111	NEGOTIATE: Negociar a conexão.

1.2.5 **Size (Tamanho):** Especifica o tamanho máximo em Kb.

1.2.6 **Data (Dados):** Dados efetivos.

1.2.7 **Checksum:** Usado para realizar a verificação da soma de todos os bits do quadro.

1.3 Especificações do protocolo.

1.3.1 Divisão dos quadros

A divisão dos quadros ocorre com base no processo de negociação realizado antes do estabelecimento da conexão. Estes serão divididos e enviados de forma sequencial e terão o tamanho máximo de 4kb. Este tamanho máximo foi pensado baseado na otimização do uso do canal, ele e a quantidade de janelas deslizantes (iguais para a transmissão e recepção) foram calculados com base na formula:

Utilização da linha = $l/(l + bR)$ {Se $l < bR$ a eficiência é menor que 50%}

Sendo:

$l = 32896$ bits (Tamanho máximo)

$b = 160$ Kbits/s

$R = 0,000012207$

Utilização = $32896 \text{ bits} / (32896 \text{ bits} + 160 \text{ Kbits/s} \times 0,000012207 \text{ s}) = 99,99\%$.

1.3.2 Entregas

Ao receber um quadro a primeira tarefa realizada pelo receptor consiste em realizar a verificação de erros com base no campo de *checksum*. Caso algum erro seja



detectado, deve-se enviar um quadro de aviso contendo a informação do quadro defeituoso.

1.3.3 Timeout padrão

O *timeout* padrão (tempo de espera para confirmação de recebimento de quadro) é de 1s, este valor é justificado pelo fato de que a velocidade padrão de transmissão de dados via cabo serial é de 160kbps e o *timeslice* de um sistema operacional UNIX é de aproximadamente 100ms. Com estes dados é possível observar que 1s é suficiente para que um quadro chegue ao outro lado, seja processado e respondido.

No caso da desconexão um *host* não conhece o tempo de resposta do outro *host*, por isto utiliza-se o *timeout* padrão.

1.3.4 Timeout calculado

Deve-se calcular o tempo de resposta de um quadro e multiplicar por 2 a fim de se obter certa folga antes de queimar uma tentativa.

1.3.5 Número de tentativas

Por padrão, estabelece-se que o número máximo de tentativas de *timeout* é 3. Isso significa que após o estouro do *timeout* deve-se tentar novamente até que o *timeout* estoure pela 3ª vez quando será considerado que o *host* destino é INACESSÍVEL.

1.3.6 Controle de erro (Checksum 20)

O calculo do *checksum* serve apenas como detecção de erros, não fornecendo correção, em um quadro inteiro (excluindo-se as *flags* de início e final). O calculo é a simples soma de palavras de 20 bits, desconsiderando *overflows*. Um exemplo seria o envio da área de dados com o tamanho máximo (64B) que daria 64x8 bits. A soma seria então dos primeiros 20 bits seguidos dos próximos 20 bits até contemplar todos os bits de dados e do cabeçalho.

1.3.7 Tamanho do payload

Quando um *host* requisita conexão com outro, no primeiro quadro de requisição deve constar também o tamanho máximo sugerido pelo requerente, isto é, o tamanho máximo do *payload*. Este tamanho significa apenas que o *payload* não deve ultrapassar o tamanho combinado, não interferindo em nada se o *payload* for de tamanho menor. O *host* requerido então devolve um quadro de confirmação constando o tamanho do *payload* negociado podendo ser menor ou igual ao tamanho sugerido.

1.3.8 Mensagem

Unidade de dado que será transmitida de um *host* ao outro. Será desmembrada em tamanhos definidos na negociação pelo *host* de origem e remontada no *host* de destino.

1.3.9 Sequenciamento de quadros de dados

Após conectados, ambos *hosts* entram no estado de espera, o requerente em espera de novas chamadas pela camada superior e o requerido em espera de novos quadros do requerente. **O requerente pode enviar um ou mais quadros pertencentes a uma mesma mensagem.** Quando envia quadros em sequencia é feito setado um bit no cabeçalho chamado de P/F, que quando 0 significa que ainda há quadros da mesma mensagem e quando 1 indica que não há mais quadros referentes a mesma mensagem.

1.3.10 Recebimento de quadros

Para garantir que dois quadros iguais não sejam interpretados como quadros diferentes (caso haja repetição de envio) os dois *hosts* fazem um trato onde cada quadro enviado após o outro e diferente do outro deve alternar o bit de janela (bit



após P/F).

Para facilitar o entendimento imagine o envio de 3 quadros de uma mesma mensagem. O emissor do quadro seta os bits de janela(W) como 0,1,0, respectivamente para cada quadro e consequentemente o receptor espera quadros com janelas 0,1,0, respectivamente.

Caso um dos quadros for enviado com erro, o emissor emitirá o um quadro com mesmo bit de janela. Então o receptor saberá quando descartar ou não os quadros enviados. [Havíamos combinado que teríamos somente uma janela].

2 Especificação de ações

2.1 *Estabelecimento de conexão*

O host que deseja estabelecer conexão envia um quadro de pedido de conexão para o outro host (111 - NEGOTIATE, SIZE=tamanho máx sugerido) e espera confirmação.

FLAG	Protocol	TF	Type	Size	Data	Checksum	FLAG
01111110	00	1	111	11111	-	00000000000000001000111	01111110

Cabeçalho do protocolo para o caso de um quadro do tipo Control

Exemplo do uso do checksum:

$$\begin{array}{rcl} & 00 & \text{Protocol} \\ & 1 & \text{TF} \\ & 111 & \text{Type} \\ + & & \\ & 11111 & \text{Size} \\ \hline & 1000111 & \text{Checksum} \end{array}$$

2.1.1 *O host destino respondeu*

O *host* destino recebeu o quadro, processou e enviou um quadro de confirmação (000 - RECEIVE_READY, SIZE = escolhido) e espera confirmação.

2.1.1.1 *O host de origem respondeu*

O *host* de origem recebeu o quadro, processou, aceitou o tamanho do *payload*, calculou o novo *timeout* e respondeu com outro quadro (101 - CONNECT) entretanto NÃO ESPERA CONFIRMAÇÃO! O *host* de origem passa para o estado de CONECTADO.

2.1.1.1.1 *O host destino recebeu*

O *host* destino recebeu o quadro, processou, calculou o novo *timeout* e passou para o estado CONECTADO.

2.1.1.2 *O host de origem não respondeu*

O *host* destino (consequentemente o de origem), permanecem no estado DESCONECTADO.

2.1.2 *O host destino não respondeu*

É gerada a mensagem de erro "Host destino inacessível".



2.2 Envio de dado

Para exemplificar o esquema de envio de dados considere o envio da mensagem "Ola Mundo!", que contém 11 Bytes (contando com o carácter “\0”), e que o *payload* negociado seja de apenas 5 Bytes. Isso gera 2 quadros de 5 bytes e mais um de 1 byte.

FLAG	Protocol	TF	P/F W	Size	Data	Checksum	FLAG
01111110	00	1	111	000101	'O'+ 'l'+ 'a'+ ' ' + 'M'	xxxxxxxxxxxxxxxxxxxxxxxx	01111110

Cabeçalho para exemplo

00 Protocol
1 TF
000 Type
+ 'O'+ 'l'+ 'a'+ ' ' + 'M' (soma em porções de 20 bits)

xxxxxxxx Checksum

FLAG	Protocol	TF	P/F W	Size	Data	Checksum	FLAG
01111110	00	1	111	000101	'u'+ 'n'+ 'd'+ 'o'+ 'l'	xxxxxxxxxxxxxxxxxxxxxxxx	01111110

Cabeçalho para exemplo

FLAG	Protocol	TF	P/F W	Size	Data	Checksum	FLAG
01111110	00	1	111	000101	"\0"	xxxxxxxxxxxxxxxxxxxxxxxx	01111110

Cabeçalho para exemplo

Perceba a lógica dos bits P/F W:

- 1º quadro: P/F = 0 indica que ainda há mais quadros da mensagem;
 - W = 0 indica que é a janela 0 e que a próxima é 1;
- 2º quadro: P/F = 0 indica que ainda há mais quadros da mensagem;
 - W = 1 indica que é a janela 1 e que a próxima é 0;
- 3º quadro: P/F = 1 indica que NÃO há mais quadros da mensagem;
 - W = 0 indica que é a janela 0 e que a próxima é 1.

Cada quadro é enviado individualmente e o próximo é enviado SOMENTE quando houver confirmação de recebimento do quadro anterior (também conhecido por ACK). Um quadro de confirmação (com sucesso) tem a seguinte forma:

FLAG	Protocol	TF	P/F W	Size	Data	Checksum	FLAG
01111110	00	1	000	000000	-	xxxxxxxxxxxxxxxxxxxxxxxx	01111110

Cabeçalho para exemplo



Em que o TYPE indica que o receptor está pronto para receber novamente. Caso houver erro no quadro, o quadro de confirmação pode conter os seguintes valores para TYPE: 001 - REJECT; 010 - RECEIVE_NOT_READY e 011 - SELECTIVE_REJECT. Para esses casos o emissor deverá saber se comportar para o reenvio ou pausa nos quadros. Ao final da transmissão dos quadros da mensagem ambos *hosts* voltam para o estado de espera inicial de cada um.

2.3 *Encerramento de conexão*

Ambos hosts podem se desconectar a qualquer momento. O quadro para pedido de desconexão é semelhante ao quadro de confirmação de dados com exceção dos bits de TYPE. O host requisitante envia um quadro com TYPE = 100 - DISCONNECT e espera confirmação, dessa vez de desconexão. O host requisitado então responde com um quadro TYPE = 110 - DISCONNECT_OK e NÃO ESPERA CONFIRMAÇÃO! Ambos hosts entram em status de desconectado.