

Protocolo Sistema Simples de Notificação Remota (SSNR)

Jonathas Augusto de Oliveira Conceição¹

¹ Curso de Bacharelado em Ciências da Computação
Universidade Federal de Pelotas (UFPel)

jadoliveira@inf.ufpel.edu.br

Resumo. *O Sistema Simples de Notificação Remota é um protocolo de notificações distribuída em modelo cliente-servidor. Nele há dois tipos de possíveis clientes, emissores e receptores de notificações. No sistema os clientes se comunicam por meio de um servidor que contém a identificação de casa receptor e, sempre que recebe uma notificação, a encaminha ao destinatário correspondente. Um cliente pode ser apenas emissor ou apenas receptor, simplificando assim a implementação do protocolo em um dispositivo.*

1. Introdução

O Sistema Simples de Notificação Remota (SSNR) é um protocolo de notificações distribuída em modelo cliente-servidor. No sistema a comunicação é feita por meio de um servidor que contém a identificação de receptores, e quando requisitado, repassa a notificação ao destinatário correspondente.

O protocolo define dois tipos de clientes, emissores e receptores de notificações. Receptores podem se conectar à um servidor enviando uma mensagem de conexão junto à um nome de identificação. Uma vez conectados eles passam a receber notificações de qualquer remetente com acesso aquele servidor que deseje contactá-los. Emissores podem enviar mensagens ao servidor requisitando uma lista de clientes conectados, e podem também enviar notificações à clientes conectados ao servidor.

O servidor então tem três funcionalidades principais: (1) indexar receptores conectados; (2) enviar lista dos receptores disponíveis à emissores; (3) receber e entregar notificações vinda de um emissor para um receptor válido.

O artigo é organizado da seguinte forma: A Seção 2 apresenta as máquinas de estados do cliente emissor, cliente receptor e do servidor; na Seção 3 todos os tipos de mensagens e seus cabeçalhos são apresentados; por fim, a Seção 4 conclui apresentado um caso de uso do SSNR.

2. Máquina de Estados

A seguir são apresentados as máquinas de estado para o Cliente Emissor, o Cliente Receptor e o Servidor. Os círculos representam estados do sistema e as transações são dadas por dois componentes: um evento que pode acontecer; e uma ação a ser tomada. Os componentes da transição são separados por um traço vertical, o evento sempre se encontra acima do traço e a ação a ser tomada se encontra abaixo.

O Cliente Emissor funciona através de trocas de mensagens com uma conexão TCP não-persistente com o servidor. Ele pode solicitar a lista atual de usuários conectados

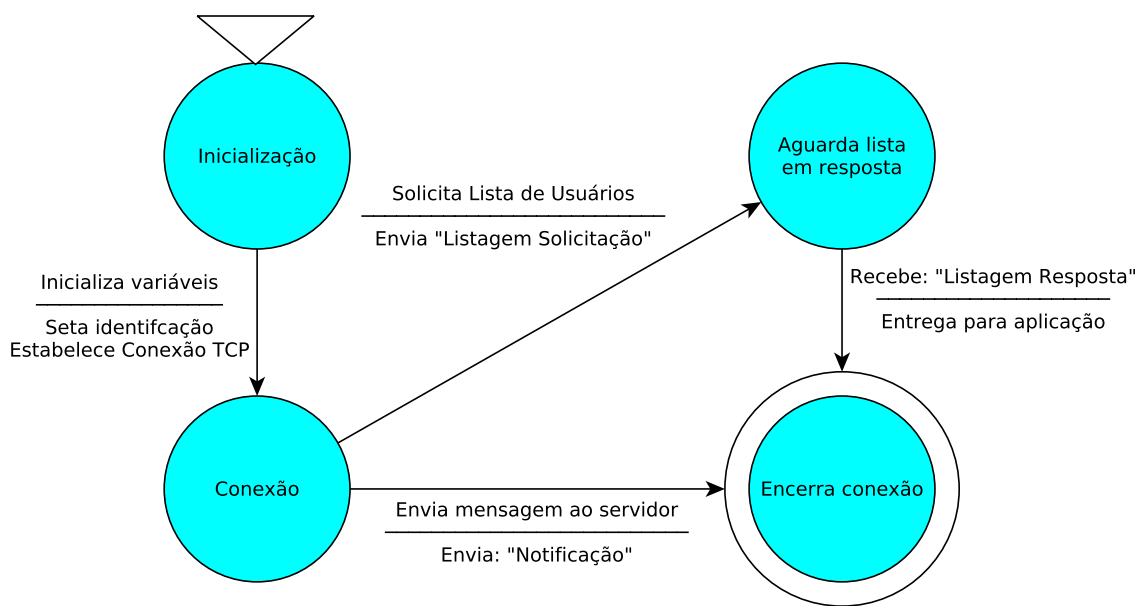


Figura 1. Máquina de estados do Cliente Emissor.

ou enviar uma mensagem para um usuário. A Figura 1 apresenta a máquina de estados do cliente emissor.

O receptor, por sua vez, tenta se conectar a um servidor usando uma conexão TCP persistente e, se bem sucedido, fica à escuta de notificações. Sua máquina de estados é apresentada na Figura 2.

O servidor, uma vez iniciado, aloca uma tabela para indexar os receptores e fica à espera de mensagens. Caso receba uma mensagem de notificação ele encontra o receptor na tabela a encaminha, se o receptor não for encontrado a mensagem é descartada; ele também deve enviar a tabela atual a qualquer receptor que faça a solicitação; e cadastra novos receptores de acordo com a requisição e a disponibilidade. Todas os tratamentos do servidor a requisições são feitos em *Threads* separadas que compartilham a tabela de receptores, a Figura 3 apresenta sua máquina de estados.

3. Mensagens

O SSNR define quatro tipos de mensagens, *Notificação*, *Listagem*, *Registro* e *Ping*. Cada grupo de mensagens são usadas dentro de um mesmo contexto e compartilham o padrão de cabeçalho, todo cabeçalho começa com um campo de **Código** de 1 byte usado para identificar o tipo da mensagem. Também é padronizado o tamanho dos "nomes", utilizados identificar usuários, sendo estes uma sequência de 16 caracteres de 8 bits (16 bytes no total).

Mensagens do tipo *Notificação* representam o principal tipo de mensagens enviadas no sistema. Elas possuem um cabeçalho de 31 bytes e corpo variável:

- Cabeçalho:
 - **Código** - 1 byte;
 - **Tamanho** - 8 bytes;
 - **Código Receptor** - 2 bytes;

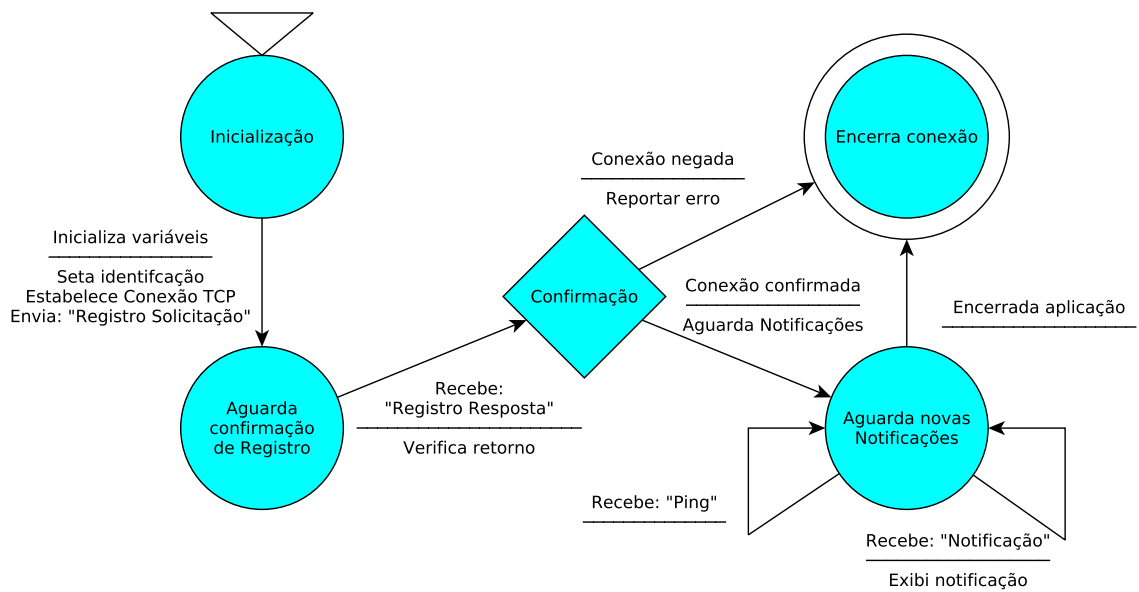


Figura 2. Máquina de estados do Cliente Receptor.

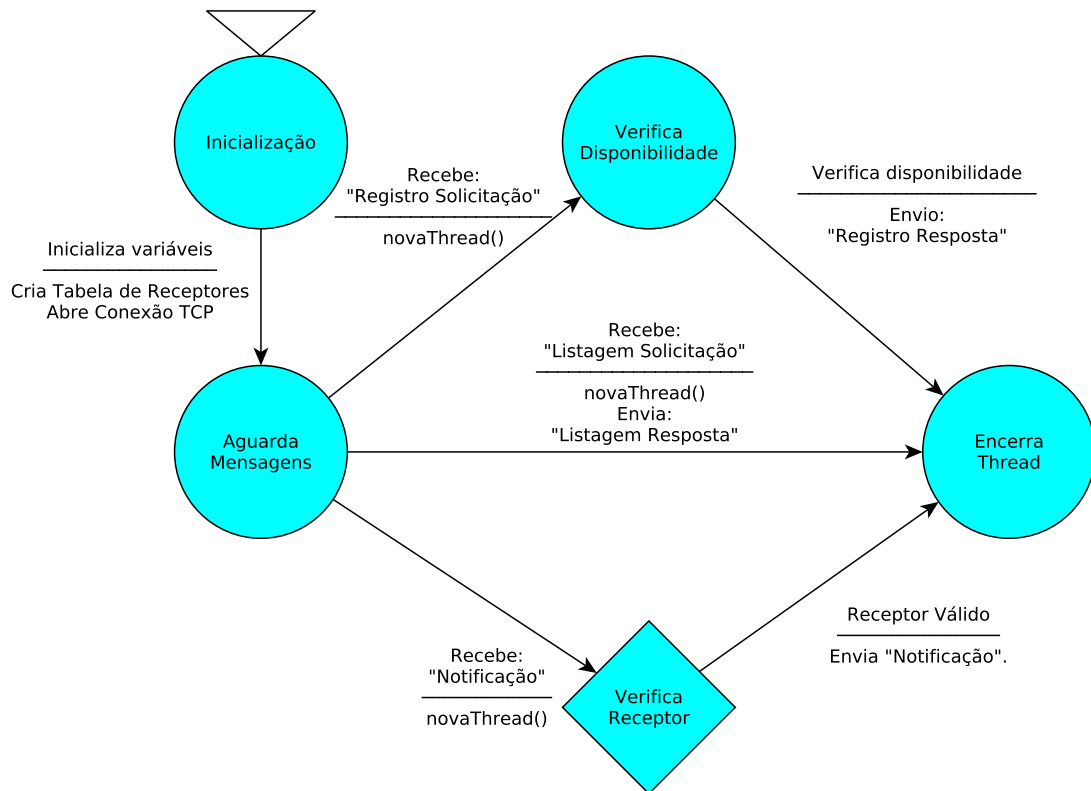


Figura 3. Máquina de estados do Servidor.

- **Timestamp** - 4 bytes;
 - **Nome do Emissor** - 16 byte.
- **Corpo:**
 - **Texto** - Variável.

O campo **Código** representa o tipo da mensagem e deve sempre conter 078 (ASCII correspondente à 'N'). **Tamanho** indica o tamanho total do **Texto** da notificação. **Código Receptor** é o identificador do destinatário da notificação no servidor, cada servidor pode então indexar até 65536 (2^{16}) receptores. **Timestamp** é o horário de envio da notificação no padrão Unix. Por fim, o pacote contém o **Nome do Emissor** seguido então pelo **Texto** da notificação.

Mensagens do tipo **Listagem** são usadas para informar emissores dos receptores disponíveis. Possuem cabeçalho de 5 bytes e dois subtipos: **Solicitação**, que são enviadas por emissores; e **Resposta**, enviada por servidores. Mensagens de **Listagem** tem a seguinte organização:

- **Cabeçalho:**
 - **Código** - 1 byte;
 - **Número de usuários** - 2 bytes;
 - **Offset** - 2 bytes.
- **Corpo em Solicitação:**
 - **Nome Emissor** - 16 bytes.
- **Corpo em Resposta:**
 - **Lista de usuários** (Código e nome) - 18 bytes por par.

O campo **Código** deve sempre conter 076 (ASCII correspondente à 'L'). Na **Solicitação**, o **Número de usuários** contém o número máximo de usuários que se deseja receber; já **Resposta**, o campo contém o número de usuários retornados na mensagem. O **Offset** é usado na **Solicitação** para controlar quais usuários pegar, se for maior ou igual ao número total de usuários no servidor a **Resposta** é enviada sem nenhum usuário.

Mensagens de **Registro** são usados no início de uma nova conexão de um receptor ao servidor. Estas mensagens possuem um tamanho total de 20 bytes e 2 subtipos: **Solicitação** e **Resposta**.

- **Código** - 1 byte.
- **Código Receptor** - 2 bytes.
- **Retorno** - 1 bytes.
- **Nome Receptor** - 16 bytes.

O campo **Código** deve sempre conter 082 (ASCII correspondente à 'R'). Na **Solicitação** o campo **Código Receptor** é usado como índice preferível, e na **Resposta** o índice ao qual o Receptor foi efetivamente alocado é informado no **Código Receptor**. O campo **Retorno** na **Confirmação** contem o resultado da operação, **Retorno** pode assumir os seguintes valores:

- 0 - Conexão bem sucedida no endereço requerido.
- 1 - Conexão bem sucedida no endereço retornado em **Código Receptor**
- 2 - Conexão recusada, servidor cheio.
- 4 - Conexão recusada, servidor lista negra.
- 128 - Erro não informado.

Por fim, a mensagem de `Ping` consiste apenas da repetição de dois bytes com valor 80 (código ASCII para 'P'). Mensagens de `Ping` são enviadas pelo servidor a todos os Receptores conectados sempre que há necessidade de conferir se a conexão ainda é válida. `Ping` é enviado sempre que um novo registro é requerido para verificar disponibilidade de índices; ele também é enviado quando uma listagem é requerida para evitar a listagem de receptores desconectados.

4. Conclusão

O Sistema Simples de Notificação Remota é um protocolo de notificações distribuídas que permite uma implementação simples de clientes, especializando-os nas funções de emissor ou receptor. Aplicações que disponham de mais robustez podem implementar ambos os protocolos de cliente se assim desejarem.

Para um exemplo de uso deste protocolo então propomos três aplicações diferentes: (1) O Servidor, de uso como descrito no protocolo. (2) Uma aplicação de terminal para *Desktops* que pode enviar notificações. (3) E um aplicativo para Celular que apenas recebe notificações. Tomando como cenário então vários *Desktops* que possuem a implementação mencionada e que computam programas que levam tempo arbitrário para terminar. É possível então, com um pouco de habilidade em *Shell Script*, fazer com que sempre que um programa terminar a execução, uma notificação seja enviada para o Celular com tal aplicativo indicando: a máquina que enviou a notificação (no campo **Nome Emissor** da Notificação), e parte da saída padrão ou saída de erro do programa (no campo **Texto**).