

Desenvolvimento de uma Aplicação para a Configuração do Sistema Embarcado InetD-Mesh

Dalvan Jair Griebler¹, Denis Valdir Benatti¹, Claudio Schepke²

¹Curso Superior de Tecnologia em Redes de Computadores
Faculdade Três de Maio (SETREM)
Av. Santa Rosa, 2405 – 98.910-000 – Três de Maio – RS – Brazil

²Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil
{dalvangriebler,schepke}@gmail.com, benatti@viahz.com.br

Resumo. *Este artigo tem por objetivo apresentar a criação de um protocolo de aplicação para configuração do sistema embarcado de redes wireless InetD-Mesh. O protocolo consiste em um serviço cliente desenvolvido em linguagem JAVA e um serviço servidor desenvolvido em linguagem C. Ambos trocam informações que são determinados pelo protocolo de aplicação criado, usando sockets como meio de comunicação. Com isso é possível fazer a configuração do sistema embarcado através de uma interface gráfica evitando erros de configuração.*

1. Introdução

Sistemas embarcados de redes *wireless* têm sido muito importantes na proliferação da Internet. No entanto, a configuração de muitos deles ainda é feita manualmente através de linhas de comando. Este é o caso de InetD-Mesh, um sistema operacional de código aberto baseado no Kernel Linux conhecido como Wive-ng. A este sistema foi adicionado a tecnologia Mesh, própria para plataformas de hardware RTL8186 de sistemas embarcados.

A manutenção desses sistemas por linha de comando exige do usuário conhecimentos específicos. Além disso, muitos erros de configuração podem ocorrer. Como por exemplo, colocação de configurações impróprias não suportadas pelo sistema, edição de arquivos, execução de comandos, nomes e números inválidos.

Em vista disso, este trabalho propõem a implementação de um mecanismo de configuração do sistema embarcado InetD-Mesh através de uma interface gráfica. Tratando-se de sistemas embarcados, sabe-se que os mesmos são limitados pela plataforma no qual trabalham. Considerando este fato, uma forma de tornar a configuração do sistema embarcado mais fácil e menos suscetível a falhas é a criação de um protocolo de aplicação.

Neste contexto, foram desenvolvidos dois serviços baseados no modelo cliente-servidor, onde o servidor é um programa responsável por servir as requisições feitas pelo cliente. O cliente oferece uma interface gráfica interativa para configurar o sistema embarcado. Os dois serviços trocam informações, que são determinados pelo protocolo de aplicação criado.

A configuração do sistema embarcado é um aspecto importante, quando trata-se da operação do mesmo. Buscou-se através deste artigo, uma forma de apresentar a arquitetura do sistema embarcado na sessão 2, o modelo de programação utilizado na sessão 3, bem como, o seu funcionamento. Descrevendo detalhadamente o protocolo de aplicação criado na sessão 4, os modelos de comunicação e autenticação nas sub-sessões 4.1 e 4.2, como é feita a comunicação entre os serviços de rede ou aplicações na sub-sessão 4.3 e a interface gráfica do serviço cliente na sub-sessão 4.4.

2. Arquitetura do Sistema Embarcado

RTL 8186 é o hardware do sistema embarcado *wireless* InetD-Mesh, produzido pela empresa Realtek, conhecido como *Integrated System-On-Chip (SOC)*. SOC é um sistema em uma única pastilha, contendo processador, memória, interface para periféricos e blocos dedicados. Portanto SOC nada mais é do que um *chip* que possui interligado diversos periféricos, sendo o barramento interno o seu meio de comunicação (SILVA, 2006).

O hardware de RTL 8186 possui arquitetura de 32 *bits*, com controlador RISC, podendo processar até 180 Mhz, além de ter 8k de memória cache (REALTEK, 2009). Este hardware possui um conjunto de instruções semelhantes a arquitetura MIPS. Assim, toda e qualquer aplicação precisa estar compilada para a arquitetura MIP, somente desta forma, a mesma funciona no sistema embarcado InetD-Mesh.

3. Programação Cliente-Servidor

Para o desenvolvimento do trabalho foi utilizada a abordagem Cliente-Servidor. Este é um modelo de programação para sistemas distribuídos em rede, sendo que tanto cliente como servidor são programas distintos neste modelo. O servidor é responsável por receber solicitações dos clientes, já o cliente depende do servidor para poder usar os recursos.

Quando são escritos programas que se comunicam por uma rede de computadores, deve-se primeiramente criar um protocolo, um acordo sobre como os programas vão fazer a comunicação. Este é conhecido como protocolo de aplicação, onde ambas as aplicações interagem entre si trocando informações (ESTEVES, FENNER e RUDOFF, 2005).

Formalmente, neste trabalho os programas são chamados de serviços de rede. Estes comunicam-se por um canal de comunicação, criado através da combinação de IP mais a porta do Cliente, e IP mais a porta do Servidor, criando um túnel ou canal, denominado de *Socket* (TORRES, 2001).

4. Protocolo de Aplicação do Sistema Embarcado InetD-Mesh

O Protocolo de aplicação desenvolvido está descrito nas próximas subseções.

4.1 Modelo de Comunicação

O protocolo de aplicação criado consiste no modelo de comunicação cliente-servidor, usando como canal de comunicação o *Socket*. O serviço servidor se encontra no sistema embarcado e o serviço cliente está em um computador conectado através da rede. Este acessa o servidor para configurar o sistema embarcado. A representação do modelo está descrita na Figura 1.

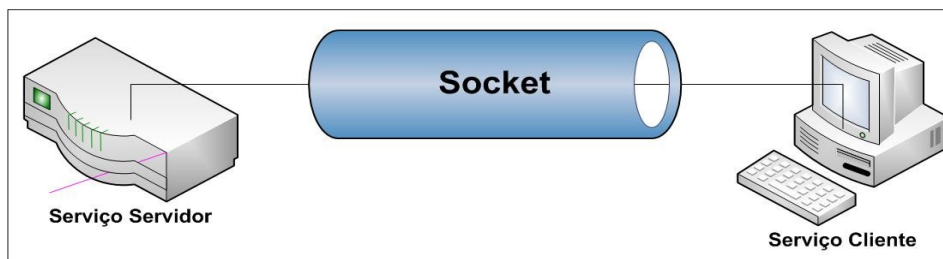


Figura1: Modelo de comunicação.

A grande maioria dos serviços de rede são programados para aceitar várias conexões. Porém, o serviço de rede criado, aceita somente uma conexão ao servidor. Com a restrição das conexões, a configuração do sistema embarcado é menos suscetível a falhas, visto que somente uma única configuração do sistema poderá ser feita por vez. Quando existe a possibilidade de mais de uma conexão, dois usuários poderiam estar configurando casualmente o sistema embarcado ao mesmo tempo, ocorrendo inconsistências durante este processo (GRIEBLER, 2009).

4.2 Modelo de Autenticação

O modelo de autenticação está representado em camadas, conforme a Figura 2. No serviço servidor na camada de hardware é o sistema embarcado. Na camada de sistema operacional está o InetD-Mesh e na camada de serviços está a aplicação servidor desenvolvida.

Na camada de hardware do lado cliente, concentra-se um computador *desktop* ou outro tipo de *Personal Computer*. A camada do Sistema Operacional não depende de um sistema específico, pois na camada superior é executado uma Máquina Virtual Java (JVM), que opera a aplicação cliente desenvolvida em Java (DEITEL, 2001).

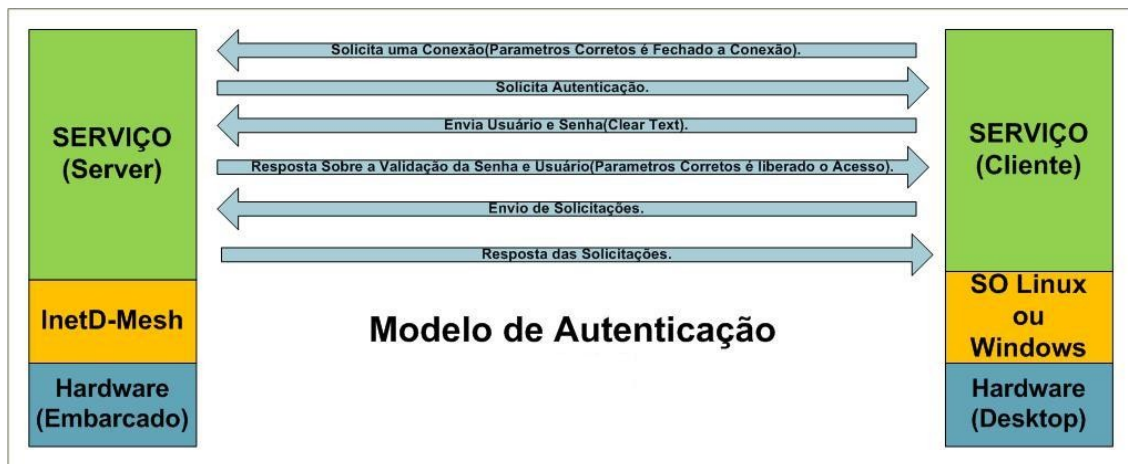


Figura 2: Modelo de Autenticação.

O processo de autenticação e comunicação se encontra na camada dos Serviços. No processo de autenticação, primeiramente o usuário conecta-se ao servidor, quando conectado é solicitado a autenticação do cliente. Para tanto, o cliente manda o usuário e a senha em *Clear Text* para o servidor. O servidor fará a comparação do usuário e da senha. Depois da comparação será enviado a confirmação, para identificar se o processo foi executado com sucesso ou não.

Se o usuário e senha forem válidos, começa o processo de configuração do sistema embarcado. Os comandos ou requisições feitas ao servidor, são respondidas,

devolvendo ao cliente o que foi requisitado. A configuração do sistema embarcado só será possível quando o processo de autenticação for válido.

No modelo desenvolvido, após o processo de autenticação, o servidor é responsável por:

- Ler o conteúdo dos arquivos do sistema embarcado.
- Enviar o conteúdo dos arquivos via *socket*.
- Receber o conteúdo dos arquivos gravados via *socket*.
- Gravar o conteúdo dos arquivos lidos do *socket* no sistema local.
- Executar alguns comandos do sistema operacional embarcado.

O Cliente, depois do processo de autenticação, tem a responsabilidade de:

- Ler o conteúdo do *socket*.
- Enviar comandos para o servidor.
- Fornecer uma interface gráfica para conexão.
- Fornecer uma interface gráfica para a configuração do sistema embarcado.

Além destes atributos, o sistema cliente é capaz de manipular os dados enviados pelo servidor, caracterizados como uma grande *string* de caracteres, pois todo o conteúdo do arquivo é enviado via *socket* e não apenas parte dele (GRIEBLER, 2009). O fim de cada arquivo é determinado pela *string* “#fim”, sendo esta interpretada por ambos os serviços.

4.3 Descrição do Protocolo

Para que cliente e servidor possam se comunicar é necessário que os mesmos estejam programados para interpretar um padrão de comandos ou mensagens que determinam a ação, que um ou outro devem realizar ao receber determinada mensagem. Para isso, é estabelecido um padrão de comunicação entre os dois serviços, sendo este determinado por mensagens do tipo *strings*, onde cada uma das mensagens utilizadas, são descritas pelo protocolo de aplicação.

A Tabela 1 possui na sua primeira coluna o comando e ao lado a respectiva descrição do que se trata o mesmo. Estes comandos são trocados durante o processo de autenticação. Os dois primeiros comandos são enviados do servidor para o cliente, confirmando o estado do processo de autenticação. O último é enviado ao servidor, quando o usuário desistir de *logar* ao fechar a janela de *login*.

Tabela 1: Processo de Autenticação.

Sistema de Autenticação	
Comando	Descrição do Comando
n	Falha na Autenticação
y	Sucesso na Autenticação
\$c\$a\$i\$f\$o\$r\$a\$	Cai Fora do Socket Quando Fechar Janela de Login

Após o processo de autenticação, são trocados outros comandos, onde somente o cliente envia os mesmos, correspondente com a ação efetuada pelo usuário na aplicação gráfica do cliente. O servidor apenas responde as solicitações do cliente. Este protocolo funciona com a combinação de ação e o tipo de ação, onde a ação é o que o usuário quer

fazer e o tipo de ação é uma espécie de especificação do que se quer realizar. A ação e o tipo de ação são separados por um ponto. A Tabela 2 demonstra o formato do comando ou mensagem enviada pelo cliente ao servidor.

Tabela 2: Funcionamento do Protocolo.

Funcionamento do Protocolo	
"<Ação>.<Tipo de Ação> "	EX: "g.g"(busca arquivo General).

O serviço servidor interpreta 4 ações, sendo estas representadas na primeira coluna da Tabela 3. A segunda coluna desta tabela, demonstra respectivamente a descrição. As ações podem ser de busca do conteúdo do arquivo, gravação do conteúdo do arquivo, execução de um comando e o término da conexão, onde os mesmos estão diretamente ligados ao servidor e executados sobre o sistema operacional embarcado.

Tabela 3: Primeira Ação.

Primeira Ação	
Ação	Descrição da Ação
g	Busca do Conteúdo do Arquivo
s	Gravação do Conteúdo do Arquivo
c	Execução de Comando no Sistema
t	Terminar Conexão

Para completar um comando, existe o segundo parâmetro, logo após o ponto. O mesmo consiste no tipo de ação, sendo esta a concordância com o primeiro parâmetro. Desta forma, a Tabela 4, demonstra a concordância das ações "g" e "s". Na primeira coluna estão os tipo de ação, na segunda coluna está a descrição e na terceira coluna esta o caminho, demonstrando os arquivos de configuração do sistema embarcado.

Tabela 4: Concordância com Ação "g" e "s".

Concordância com Ação "g" e "s"		
Tipo de Ação	Descrição do Tipo de Ação	Local
g	Arquivo General	/etc/network/wifi/general
l	Arquivo Interfaces	/etc/network/interfaces
b	Arquivo Boot	/etc/boot
w	Arquivo Wpa	/etc/wpa.conf
m	Arquivo Olsrd	/etc/olsrd.conf
d	Arquivo Dnsmasq	/etc/dnsmasq.conf
r	Arquivo Group	/etc/group
h	Arquivo Hostname	/etc/hostname
o	Arquivo Hosts	/etc/hosts
p	Arquivo DnsCli	/etc/resolv.conf
s	Arquivo Service	/etc/services
u	Arquivo Udhcpd	/etc/udhcpd.conf
f	Arquivo Iptables	/etc/network/iptables
c	Arquivo Routes	/etc/network/routes
t	Arquivo TcStart	/etc/network/tc.start
v	Arquivo Vlan	/etc/network/Vlan
n	Arquivo Unchecked	/etc/network/wifi/unchecked
a	Arquivo Root	/etc/crontabs/root

A Concordância com a ação “c”, possui sua representação na Tabela 5. Na primeira coluna está descrito o tipo de ação, na coluna ao lado está a sua respectiva descrição e na última coluna estão os locais onde são executados os comandos do sistema operacional embarcado.

Tabela 5: Concordância com Ação “c”.

Concordância com Ação “c”		
Tipo de Ação	Descrição do Tipo de Ação	Local
s	Comando Salvar no Disco	fs save
r	Comando Reiniciar Router	reboot

Através do conjunto de comandos é especificado o protocolo de aplicação. Meio pelo qual os serviços de rede se comunicam, possibilitando a interação com o sistema embarcado. Estas mensagens são enviados sem que o usuário perceba, pois os mesmos são enviados quando o usuário interage com a interface gráfica do cliente.

Toda a inteligência em manipular o sistema embarcado encontra-se no servidor. Ao interpretar as mensagens enviados pelo cliente, o mesmo pratica uma ação respondendo ao cliente o que lhe foi requisitado. Geralmente são processos de leitura e gravação em arquivos, contidos no sistema embarcado e executados os principais comandos do mesmo.

A parte de configuração e alteração dos arquivos de configuração é realizado no cliente. O conteúdo do arquivo é enviado por inteiro e tratado na memória do cliente, que por sua vez prepara o conteúdo na interface gráfica. Quando o usuário terminar a configuração, as mesmas são alteradas na memória, sendo assim, todo o conteúdo do arquivo é enviado ao servidor para ser gravado no sistema embarcado. Embora pareça simples, o cliente trabalha com instruções mais complexas, concentradas na parte de manipulação do conteúdo dos arquivos (GRIEBLER, 2009).

O conteúdo dos arquivos é enviado conforme a mensagem é enviada ou requisitada. A interface gráfica abstrai a camada do protocolo de aplicação, onde existe a troca de mensagens. Toda ação praticada na interface gráfica, gera o envio de uma mensagem ao servidor, automaticamente enviando uma requisição, onde a mesma é interpretada pelo servidor e retornada para o cliente.

4.4 Interface Gráfica do Serviço Cliente

O serviço cliente tem por objetivo fornecer uma interface gráfica para se conectar ao servidor e também efetuar as configurações do sistema embarcado (GRIEBLER, 2009). A Figura 3 demonstra a tela de conexão com o sistema embarcado. Por padrão os dois serviços operam na mesma porta, sendo que, para conectar-se ao servidor, é necessário especificar qual é o endereço de IP do servidor, usuário e a senha.

Nesta tela também são fornecidas as opções de adicionar e editar roteadores, os quais ficarão armazenados em um arquivo, sendo que o mesmos serão listados no campo roteador. Para a conexão com um sistema embarcado, primeiramente é necessário cadastrá-lo no serviço cliente.



Figura 3: Tela de Acesso.

A tela principal de configuração do sistema embarcado, está representada na Figura 4. Ela somente ficará disponível quando o usuário e a senha foram validados. Após isso, começará o processo de configuração do sistema embarcado. A interface cliente fornece um menu de opções de configuração e botões de comandos que são executados no sistema embarcado.

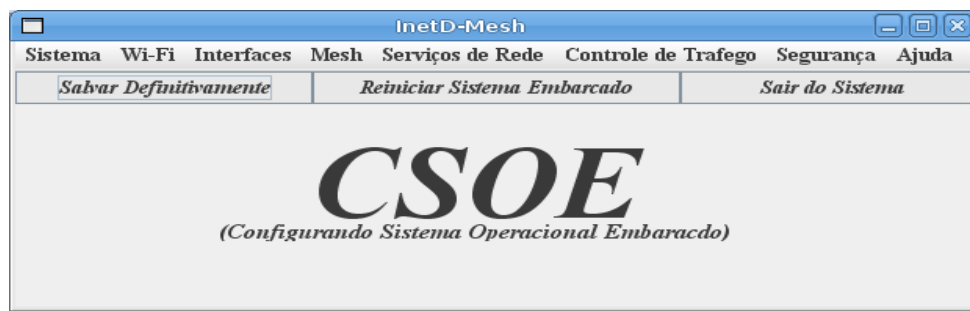


Figura 4: Tela de Configuração Principal.

Através da tela principal de configuração, são chamadas as telas de configuração do sistema embarcado, carregadas pelo menu de opções, tornando o processo de configuração mais amigável e menos suscetível a falhas.

A aplicação cliente, tem por finalidade fornecer uma interface gráfica para configurar todos os sistemas embarcados InetD-Mesh de uma rede, desde que os mesmos possuam o serviço servidor operando com o sistema InetD-Mesh, para o qual foi desenvolvido esta aplicação (GRIEBLER, 2009).

5. Conclusão

Esta aplicação teve forte contribuição na facilidade de se configurar o sistema embarcado, simplificando este processo. Assim, é possível elevar a quantidade de usuários do sistema embarcado. Com a interface gráfica da aplicação desenvolvida, também são evitados os erros de configuração do sistema embarcado.

O modelo de programação cliente-servidor é uma boa alternativa para trabalhar com sistemas bastante limitados. Este modelo de programação possibilita a criação de um protocolo de aplicação, fazendo com que dois programas de linguagens distintas conversem entre si, trocando informações através de mensagens, sendo que estas, possibilitam a configuração do sistema embarcado.

Para trabalhos futuros com base neste sistema, podem ser implantados um sistema de confirmação de envio do arquivo para garantir a integridade do mesmo, implementar um algoritmo de criptografia no cliente para que usuário e senha possam estar criptografados no envio da mesma e aumentar o número de opções de configuração no sistema cliente.

Referências

- DEITEL, M. H, DEITEL, J. P. **JAVA Como Programar**. Porto Alegre: Bookman, 2001.
- ESTEVES, W. Ricardo, FENNER, Bill, RUDOFF, M. Andrew. **Programação de Redes Unix**. São Paulo: Bookman, 2005.
- GRIEBLER, J. Dalvan. Orientadores: BENATTI V. Denis, SCHEPKE Cláudio. **Desenvolvimento de uma Aplicação Cliente-Servidor Para a Configuração de um Sistema Embarcado de Redes Wireless**. Três de Maio: SETREM, Faculdade Três de Maio, Conclusão de Curso, 2009.
- REALTEK. **RTL8186**. Disponível em: <<http://www.realtek.com>>. Acesso em: 15 de março de 2009.
- SILVA, D. C. Daniella, Orientadores: MELCHER, Elmar Uwe Kurt, FECHINE, Joseana Macêdo. **Desenvolvimento de IP Corre de Pré-Processamento Digital de Sinais de Voz para Aplicações de Sistemas Embutidos**. Campina Grande: Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, Dissertação de Mestrado, 2006.
- TORRES, Gabriel. **Rede de Computadores**. Rio de Janeiro: Axcel Books, 2001.