

PRINCÍPIOS E PERSPECTIVAS DA TECNOLOGIA DE RÁDIO DEFINIDO POR SOFTWARE (RDS) E SUA IMPLEMENTAÇÃO VIA GNU RÁDIO

LEANDRO G. B. FERREIRA

*Engenharia de Telecomunicações/Computação
Instituto de Educação Superior de Brasília - IESB*

<http://www.iesb.br>

e-mail: leandro_bsb@hotmail.com

Resumo - Com a evolução das tecnologias de comunicação móvel uma nova gama de produtos e serviços foi disponibilizada no mercado. A crescente necessidade de integração destes produtos e serviços, mais uma vez impulsionada pelo desenvolvimento militar, deu origem à tecnologia de Rádio Definido por Software (RDS), tecnologia essa capaz de atingir altos níveis de integração de hardware, concatenada com a enorme flexibilidade do software. Neste artigo será abordada uma plataforma de desenvolvimento aberta para a implementação de sistemas RDS, com especial foco na implementação do GNU RADIO. Abordar-se-á também as perspectivas e desafios dos sistemas RDS.

Abstract - With the evolution of the technologies, especially those related to mobile communication, a new generation of products and services had been launched. The increasing need of integration of these products and services, promoted by the military application development, has generated a new technology called Software Defined Radio (SDR), capable to reach high levels of integration on hardware, concatenated with the flexibility of software. This paper introduces the implementation of GNU RADIO, a platform of free development capable to offer SDR services. The challenges and perspectives of de SDR systems will also be presented.

Keywords - Software Defined Radio, GNU Radio, DSP, CORBA.

1. INTRODUÇÃO

Durante os últimos vinte anos, são notáveis as formas com que as tecnologias relacionadas à comunicação móvel vêm evoluindo. Como quase todo desenvolvimento tecnológico que inicialmente é destinado a aplicações militares, na área de comunicação móvel não tem sido diferente. Vários são os equipamentos, protocolos e tecnologias voltados ao meio militar, adaptados para serem economicamente viáveis e úteis ao mercado civil. Esta crescente evolução tecnológica inseriu no mercado uma série de produtos e serviços que atualmente são bastante utilizadas por muitas pessoas no dia-a-dia. Não é difícil imaginar alguém que use, simultaneamente, celular e um notebook interligado a uma rede wireless, ou vendo sua localização via o sistema de GPS (Global Positioning System), enquanto continua com o rádio FM ligado. Entretanto, com toda essa gama de tecnologias e serviços, ainda se faz

necessária uma variedade de dispositivos específicos para cada aplicação. Todavia, sob o ponto de vista militar, onde é imprescindível transportar o mínimo possível de equipamentos e ter a maior quantidade de recursos disponíveis, que possam efetuar várias funções simultaneamente, apresenta-se então, em ambos os casos, a necessidade de uma tecnologia integradora, altamente compacta e de grande flexibilidade.

A tecnologia de Rádio Definido por Software (RDS), ainda é uma tecnologia emergente e em desenvolvimento, mas que demonstra um imenso potencial, não só para uso militar, mas também para uso civil. Neste artigo serão abordados aspectos técnicos dos sistemas RDS, dando ênfase à parte do software, e em especial à plataforma GNU RADIO, juntamente com algumas alterações do modelo clássico dos sistemas RDS que se acredita, em um primeiro instante, torná-lo mais eficiente e flexível.

2. CONCEITO RDS

O objetivo da tecnologia RDS é digitalizar o sinal o mais próximo possível da antena, e fazer todo o Processamento Digital do Sinal (PDS) via software. Como a flexibilidade e a independência fornecidas pelo software são enormes, as possibilidades de produtos e serviços que poderão ser oferecidos aos usuários, ou ainda, os níveis de customização que as empresas, ou mesmo os usuários, poderão alcançar, estarão disponíveis apenas substituindo ou selecionando um outro software para operar em seu hardware.

Ainda assim, é necessário explicar a diferença entre os sistemas RDS e os sistemas baseados em software. No caso do último, são sistemas que utilizam uma implementação via software, assemelhando-se muito com o conceito de RDS, exceto pelo ponto que, qualquer alteração na padronização de algum aspecto da comunicação, implicará na substituição e/ou ajuste no hardware, pelo mesmo não ser capaz de se adaptar de forma dinâmica, tornando este, o principal ponto de diferença entre os sistemas RDS e os sistemas de rádio baseados em software [1]. Abaixo, verifica-se um modelo híbrido de várias literaturas que auxiliará na conceituação e entendimento dos sistemas RDS.

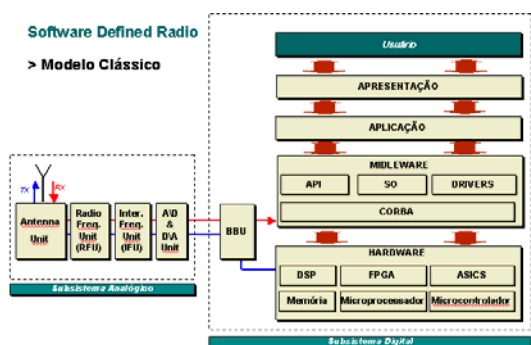


Figura 1 - Modelo RDS Clássico [2] [3]

O subsistema analógico é inicialmente composto pela AU (Antena Unit), que é um sistema de antenas inteligentes e multibanda. A RFU (Radio Frequency Unit), na transmissão, eleva os sinais recebidos à frequência necessária para a transmissão. Na recepção o sinal passa por um downconverter de forma a levá-lo a uma FI (Frequência Intermediária) ou para banda base [2].

A IFU (Intermedite Frequency Unit), na transmissão, é responsável por elevar o sinal recebido do ADC/DAC (Analog Digital Converters, Digital Analog Converters) e elevá-lo a uma frequência intermediária para que possa ser identificado e transmitido pela RFU e AU. Na recepção, o IF é responsável por filtrar os sinais para que eles possam ser tratados pelo DAC. Quando na transmissão, o sinal recebido da BBU (Base Band Unit) é convertido em sinal analógico via DAC, que dá seqüência à transmissão. Na recepção, após receber um sinal da IFU, ele é amostrado e transformado em um sinal digital, e posteriormente enviado a BBU [2]. Atualmente as limitações tecnológicas permitem ter um nível de reconfigurabilidade apenas a partir do ADC/DAC, o que de forma alguma limita a tecnologia. O subsistema analógico tem como foco a transmissão RF (Radio Frequency). A tecnologia de hardware empregada neste subsistema será descrita mais à frente, juntamente com a tecnologia de hardware usada na implementação do subsistema digital.

O subsistema digital é onde atualmente já existe uma total abstração do hardware, assegurando, assim, a flexibilidade conceitual existente no modelo clássico dos sistemas RDS.

O primeiro bloco do subsistema digital é a BBU. Na transmissão, o sinal é modulado, digitalmente, e enviado ao ADC, onde é preparado para a transmissão. Na recepção, a BBU demodula e decodifica o sinal, além de auxiliar na redução da interferência, dando maior confiabilidade ao sinal recebido [2].

3. O HARDWARE DO RDS

Não é tarefa simples definir as características mínimas do hardware para um sistema RDS. Como todo projeto, a definição de uma estrutura básica vai depender das necessidades específicas de cada caso. Apesar de estar intimamente vinculado à definição do projeto, há características que são recomendadas para a implementação de um sistema RDS. Os sistemas de PDS, que são uma das bases dos sistemas RDS, necessitam de uma grande eficiência de operações em tempo real, velocidade superior aos microprocessadores em operações genéricas, e uma alta capacidade de processar números em tempo reduzido,

sendo necessária ainda uma alta capacidade de reconfigurabilidade para facilitar a sua implementação de sistemas RDS [1].

Sensível a este mercado potencial, as indústrias buscam desenvolver dispositivos capazes de atender a esses princípios básicos. A seguir, serão descritos superficialmente os três principais tipos de dispositivos voltados para esta necessidade.

Os dispositivos DSP (Digital Signal Processing) tornam a solução flexível e com um bom desempenho, mas traz, em contra partida, o alto consumo de potência. A tecnologia ASIC (Application Specific Integrated Circuits) oferece um baixo consumo e uma boa capacidade de miniaturização. Porém, a ausência da possibilidade de reconfiguração torna seu uso bem restrito em sistemas RDS. Como uma solução intermediária temos a tecnologia FPGA (Field-Programmable Gate Array), que utiliza VHDL (VHSIC Hardware Description Language), que é menos popular que a linguagem C e que tem como um dos problemas o fato da reconfigurabilidade não ser tão dinâmica quando em um DSP [1].

O fato de esses dispositivos serem exemplos para a implementação de sistemas RDS, não exclui a possibilidade da utilização de qualquer outro tipo de processador, desde que eles atendam aos requisitos de projeto, ou à utilização de projetos híbridos dessas tecnologias, auxiliados ainda, por processadores clássicos e memórias. Vale lembrar que devido às características apontadas, estes processadores, DSP, FPGA e ASICS, também são empregados no subsistema analógico para a execução de operações específicas.

4. O SOFTWARE DO RDS

Considera-se aqui, para fins didáticos, que o sistema de software do RDS seja composto apenas pelo subsistema digital, descrito no modelo de referência (Figura 1). A porta de entrada para os sinais tratados pela BBU é o bloco Middleware. Este bloco é responsável por toda a interface entre a aplicação e o hardware, onde se tem: a função de API (Application Programming Interface), responsável pela interface entre as aplicações e o Middleware, o SO (Sistema Operacional), que pode ser desde

um Kernel Linux¹ até algo proprietário, os “DRIVERS”, responsáveis pelos parâmetros de configuração do sistema RDS, e o CORBA2 (Common Object Request Broker Architecture) que é uma arquitetura que permite às aplicações fazerem solicitações a objetos [3].

O bloco de aplicação é a camada responsável pela comunicação com o Middleware, via API. Entretanto, neste bloco não existe a preocupação com o hardware, o que facilita o desenvolvimento de aplicações de alto desempenho voltadas às mais diversas necessidades [3].

O bloco de apresentação/usuário é onde efetivamente ocorre a interação com o usuário, neste ponto todo o trabalho de processamento digital do sinal é convertido em algo palpável para o usuário, desde uma ligação “voz” até a visualização de HDTV (High Definition Television), passando ainda pela atualização de um documento na rede via 802.11.

O bloco de hardware, como já visto, pode ser composto por DSP’s, FPGA’s, memórias, processadores e etc.

5. O GNU RADIO

O potencial da tecnologia RDS fomentou, inicialmente, o interesse de instituições militares e, conseqüentemente, o interesse da comunidade acadêmica e de empresas privadas. Com o intuito de fortalecer e estudar esta tecnologia fora do meio militar, vários grupos de estudos foram criados ao redor do mundo, unindo esses interesses em prol da maturação da tecnologia em questão.

Partindo deste princípio, surgiu o GNU RADIO, ou seja, uma plataforma modular e aberta de desenvolvimento, para a criação e implementação de blocos, para o processamento de sinais focado em sistemas RDS.

A grande vantagem do GNU RADIO é o fato de se tratar de uma tecnologia aberta, o que permite o desenvolvimento modular e descentralizado, que conseqüentemente facilita sua implementação em uma mesma base (Kernel Linux), o que ajuda em muito nos testes e na melhoria contínua dos sistemas. Os blocos de processamento são desenvolvidos para as mais variadas aplicações dentro do subsistema digital, desde o tratamento de sinais na entrada,

¹ Caso do GNU RADIO.

² Será tratado mais à frente item seis.

decodificação, demodulação até sua saída em forma de HDTV, FM, GPS e voz. Outro fator que amplia e acelera o desenvolvimento dos módulos é o fato de seu desenvolvimento ser em C++, uma linguagem bem popular e já amplamente utilizada em outras aplicações.

Apenas para ilustrar o potencial do que pode ser desenvolvido com certa facilidade, usando como base alguns blocos, classes e subclasses já existentes, padronizadas e disponíveis pelo GNU RADIO (www.gnuradio.org), será exibido as linhas necessárias para se criar duas ondas senoidais de 350Hz e 440Hz [4].

```
Src0 = gr.sig.source_f (sampling_freq, gr
GR_SIN_WAVE, 350, ampi)
Src1 = gr.sig.source_f (sampling_freq, gr
GR_SIN_WAVE, 440, ampi)
```

É evidente que a amostra acima sozinha, sem os demais parâmetros necessários, não tem utilidade. A implementação de todas as funções realizadas analógicamente é feita a partir do desenvolvimento de modelos matemáticos dos sistemas analógicos que se deseja construir e estas funções devem representar, da forma mais próxima possível, a realidade dos sistemas analógicos. Estes modelos permitem o desenvolvimento de softwares capazes de desempenhar digitalmente, com muita fidelidade, qualquer tipo de sistema de sinal analógico.

A composição desses modelos, no caso específica do GNU RADIO, gera como resultado a criação de classes, subclasses, pacotes e blocos que agilizam o desenvolvimento. Por exemplo, existem classes que fazem processamentos específicos, como criar uma onda senoidal com uma amplitude específica, poupando tempo e, de alguma forma, garantindo um padrão de tratamento da informação e conseqüente compatibilidade.

Outro fato importante que facilita a implementação de sistemas RDS baseado no GNU RADIO é o fato dele operar sobre um Kernel Linux, deixando todo o sistema bem “leve” e ainda com a possibilidade de ser implementado fora de PCs. O GNU RADIO ainda possui uma plataforma de desenvolvimento de interfaces gráficas, o Python. O site do GNU, além de disponibilizar um toolkit do Python, possui vários processos de comunicação interna, diversos blocos e classes, além de uma boa documentação que pode ser usada ou

aprimoradas durante o desenvolvimento destas interfaces [4].

Encaixando ao modelo apresentado no item cinco, o GNU RADIO é uma ferramenta que pode representar todos os blocos do subsistema digital, com exceção ao bloco de hardware. Essa capacidade de representar todas as etapas deste subsistema, juntamente com as facilidades de implementação, tornam o GNU RADIO hoje uma importante ferramenta de simulação.

6. PERSPECTIVAS

Com todas as estruturas que compõem um sistema RDS já descritas, seria relativamente simples descrever o seu potencial, perspectivas e benefícios de sua implantação. Todavia, ao longo do desenvolvimento deste trabalho, notou-se um ponto hipotético de relativa falta de flexibilidade no modelo clássico, ponto esse que em uma análise preliminar, poderia comprometer a total independência do software em relação ao hardware, que é base do conceito que se tem de RDS. Assim, serão descritas, brevemente, as possibilidades de interfaceamento entre as camadas do sistema RDS clássico, dando especial realce no ponto onde se entende que pode ser implementada uma melhoria. A interação do usuário com o sistema é feita através da camada de apresentação/usuário, que atua da mesma forma que em um PC, ou qualquer outro sistema, convertendo as informações processadas em algo palpável, ou encaminhando as informações recebidas para o processamento. A interface entre a camada de apresentação e aplicação funciona da forma tradicional, coordenada para a obtenção dos resultados desejados. Como já foi dito anteriormente, a camada de aplicação se comunica com o middleware via uma API. Neste ponto, já se nota que poderá existir uma possível falta de compatibilidade entre a aplicação e o middleware. Imaginando que cada fabricante tende a desenvolver um middleware de maneira a potencializar seus recursos de hardware e que ainda há a possibilidade do desenvolvimento independente, como o GNU RADIO, há uma grande possibilidade de problemas. Assim, não seria lógico e economicamente viável desenvolver uma aplicação com várias versões para os vários middlewares. No sentido de reduzir este problema, pode-se adotar a implementação de MV's (Máquinas

Virtuais), amplamente utilizadas atualmente nos PCs. Um bom exemplo é a máquina virtual Java JVM (Java Virtual Machine), utilizada em muitos PCs portáteis e outros vários tipos de equipamentos e sistemas. A principal vantagem de uma aplicação baseada em uma MV é a liberdade que se ganha no desenvolvimento de aplicações, tornando-se possível, então, o desenvolvimento das aplicações sem a necessidade de se preocupar com a eventual compatibilidade, ou não do middleware, tornando como único requisito que o middleware suporte a MV convencional.

Teoricamente resolvido os problemas com a camada de middleware e de aplicação, a próxima parte é como compatibilizar as inúmeras possibilidades de middleware com a grande variedade de hardware em que se pode implementar um sistema RDS. Inicialmente, observa-se que algumas vertentes acadêmicas e da literatura sugerem para esta interface a arquitetura CORBA, definida pela OMG (Object Management Group). A OMG é uma organização internacional que fornece diretrizes e especificações para o desenvolvimento de aplicações. Por definição, a arquitetura CORBA permite que aplicações façam solicitações a objetos de uma forma transparente, estando este objeto local ou não. Baseado nesta definição pode-se dizer que o foco da arquitetura CORBA é o processamento distribuído, onde os objetos se comunicam e são desenvolvidos em uma linguagem própria, independente da estrutura de software e hardware. Essa linguagem é o IDL (Interface Definition Language).

Simplificadamente, o processo do CORBA funciona da seguinte maneira: o cliente requisita um serviço de implementações de objetos (Servant) através de um ORB (Object Request Broker). Este, por sua vez, encaminha ao objeto responsável pela execução do processo, estando ele local ou remoto. Após a conclusão do processo, a ORB entrega ao cliente o resultado de forma transparente, independente de onde ele foi obtido [5].

Outro aspecto importante a ser notado, é que imagina-se poder obter uma implantação mais eficiente se fosse optado pelo desenvolvimento de uma MV, cujo ambiente de execução simula uma certa arquitetura computacional que seja passível de ser implantada nos vários tipos de hardware e que forneça uma saída uniforme para os desenvolvedores de

middleware. Desta forma, estaria garantida a independência total do software em relação ao hardware [6].

Um questionamento apropriadamente levantado é definir qual a necessidade de criar uma interface baseada em MV se é possível fazer essa interface usando CORBA, que já está bem mais desenvolvido? Argumenta-se então que o CORBA é muito eficiente em seu propósito de processamento distribuído no caso de usá-lo como interface com o middleware, e é difícil de imaginar que ao solicitar um processamento, por mais limitado que seja o seu hardware, ele não consiga fazê-lo. No caso desta incapacidade, o repasse nesse nível levaria este processo de hardware específico para rodar em outro lugar, tornando o sistema mais lento e aumentando a possibilidade de erros e falhas, pois essa comunicação depende de rede. No caso de uma indisponibilidade da rede poderia se inviabilizar a execução de algum objeto fundamental para o middleware. Isso paralisaria o sistema, limitando a sua operacionalidade. Mesmo no caso de todos os objetos estarem localmente no equipamento, acredita-se que a complexidade do CORBA torna o processamento mais lento do que se fosse feito em uma MV.

Já com a implementação de uma MV, acredita-se que todo o processamento seria feito de forma mais eficiente. Porém, seria necessário um grande esforço inicial para a padronização. Este esforço deve ser de forma conjunta, como a que foi feita no desenvolvimento do CORBA e da Java JVM, onde todos os principais fabricantes opinaram e encontraram um padrão, que é referência no mercado atualmente. Como consequência, este padrão seria tomado como base para o desenvolvimento de middlewares independente do desenvolvedor ou da plataforma.

Ainda, em relação ao CORBA, seria muito interessante sua adoção no desenvolvimento do middleware. Imagina-se um cenário em que seja necessário executar algum algoritmo, incubado em um objeto IDL, que não esteja local ou que seja dinâmico, como uma chave de criptográfica. Seria possível, de forma transparente e simultânea, executá-lo e/ou gerá-lo remotamente. Com a utilização da arquitetura CORBA, pode-se então compor uma informação que seu sistema tenha capacidade de transmitir, mas não poderia gerá-la localmente, pelos mais diferentes motivos. A seguir verifica-se o modelo de

um sistema RDS com as observações e ajustes que foram evidenciadas neste item.

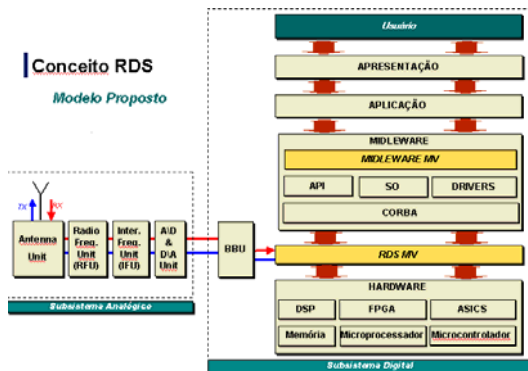


Figura 2 - Modelo sistema RDS - Proposto

Ficam registradas assim, as considerações ao modelo clássico, não esgotando, portanto, o assunto que, evidentemente necessita de mais pesquisas, simulações e testes para que as considerações feitas aqui sejam comprovadas.

7. CONCLUSÃO

Este artigo abordou de maneira superficial, as principais características e técnicas dos sistemas RDS e alguns modelos para sua implementação. Este trabalho teve como proposta contribuir para a maturação desta tecnologia que se entende tornar-se uma revolução nos sistemas de comunicação móvel.

A consolidação dos sistemas RDS não está vinculada apenas à evolução das tecnologias de hardware e software, necessárias para sua implementação, mas também aos aspectos regulatórios internacionais e de segurança, que são de extrema importância e devem ser estudados com maior profundidade, para não comprometer a imagem da tecnologia em questão. Neste ponto, a tecnologia GNU RADIO será de grande utilidade para simulações e testes.

A tecnologia RDS, sob outra perspectiva, servirá também para o nivelamento das tecnologias entre os países com maior desenvolvimento tecnológico e os países subdesenvolvidos, uma vez que, a única limitação desta tecnologia é o desenvolvimento de software.

Deve-se ressaltar, também, que toda tecnologia integratória é recebida com certa resistência no mercado civil, cabendo assim à indústria, buscar a melhor forma de abordagem em seus produtos, de maneira

a não “depreciar” a tecnologia junto ao público civil.

Admitindo-se um cenário ideal, no auge da maturação, se teria um sistema RDS capaz de oferecer qualquer informação a qualquer tempo em qualquer lugar.

AGRADECIMENTOS

Ao professor Eduardo Wolski pela grande ajuda na transição das abstrações sobre o tema em idéias palpáveis, aplicáveis à tecnologia em questão e ao amigo Denard Costa Soares pela ajuda no entendimento do universo das máquinas virtuais.

REFERÊNCIAS

- [1] LIMA, André G. M. Rádio Definido por Software: o próximo salto no mundo das telecomunicações e computação. Disponível em: www.revdigonline.com. Acesso em: junho, 2004.
- [2] HARADA, Hiroshi; PRASAD, Ramjee. Simulation and software radio communications. Boston: Artech House, 2002.
- [3] Corrêa Júnior, Rivaldo G.; Oliveira Jansen C. A tecnologia do Processador Digital de Sinal (PDS) aplicada ao Rádio Definido por Software (RDS) - 3G. Disponível em: www.revdigonline.com. Acesso em: junho, 2004.
- [4] BLOSSOM, Eric. Exploring GNU Radio. Disponível em: www.gnu.org/software/gnuradio/exploring-gnuradio.html. Acesso em: junho, 2005.
- [5] ARQUITETURA CORBA. 2001. Disponível em: www.ic.unicamp.br/ra07293/corba/corba.htm. Acesso em: maio, 2005.
- [6] MÁQUINA Virtual Java (JVM). Disponível em: www.infowester.com/jvm.php. Acesso em: junho, 2005.

AUTOR



Leandro Ferreira, cursando o 7º semestre de Engenharia de Telecomunicações no IESB, Brasília-DF, Administrador de Redes da CNI, Brasília-DF.