

Implementação de Aplicativo para Seleção de Redes em Ambientes Android

Luiz Araújo Carvalho, Claudio de Castro Monteiro, Joaquim Carvalho

luizcarvalho@redrails.com.br, ccm@ifto.edu.br, jjcarvalho@ifto.edu.br

Instituto Federal de Ciência e Tecnologia do Tocantins, Brasil
310 SUL, Avenida LO 05, s/n. Plano Diretor Sul, CEP:77.021-090
Palmas - Tocantins

Resumo: Neste trabalho é abordado sobre a grande popularização de dispositivos móveis como smartphones e tablets, que ano após ano vem ganhando cada vez mais o gosto do consumidor tanto no que diz respeito à necessidade como na comodidade. Com desenvolvimento focado em aparelhos munidos de um dos principais responsáveis por esse avanço, o Sistema Operacional Android. Junto a isso é salientado a inquestionável importância das redes sem fio, e da qualidade que esta deve possuir, para a completa satisfação dos proprietários destes dispositivos móveis. É proposto então a implementação de um algoritmo escrito na linguagem de programação Java, escolhida por diversos fatores, principalmente por ser a linguagem nativa da plataforma. Algoritmo este, que possa ao mesmo tempo em que monitore simultaneamente ambas interfaces de um dispositivo com Android consiga de maneira menos complexa possível e com um eficiente custo computacional qualificar esta rede. Durante o processo de monitoramento o aplicativo deve ficar responsável também pela decisão de troca ou não para uma interface, com base nos dados coletados, com melhor qualidade sempre que necessário. Diante deste necessidade é implementado no algoritmo uma métrica chamada QoV-NS, que tem como característica principal a utilização de dados estatísticos obtidos por disparos de pacotes ICMP como perda, jitter e atraso. E por fim é realizado uma serie de testes ostensivos para a comprovação da eficácia do projeto proposto.

Abstract: This paper discussed about the great popularization of mobile devices like smartphones and tablets, which year after year is becoming increasingly consumer acceptance both in need and the comfort. The development was focused on devices equipped with the Android OS, one of the main responsible for this growth. And still the unquestionable importance of wireless networks, and your quality, to the complete satisfaction of the mobile devices owners. It is proposed then the implementation of the code in java program language, chosen by several factors, mainly because it is the Android native language. This application, should monitor both device interfaces present in Android simultaneously, with a less complex and efficient computational cost, and finally, classify the interface. During the process of monitoring the application is responsible for the decision, and change or not of the preferential network interface, based on data collected. The application uses an implementation of a metric algorithm called QoV-NS, witch use of statistical data obtained by ICMP packets, as loss, jitter and delay. And last is executed a series of ostensible tests to confirm the effectiveness of the proposed project.

Palavras-chave: desenvolvimento, android, rede, qos, telecomunicação.

1. Introdução

Grande parte dos smartphones atuais possuem métodos de escolha de redes com baixa eficiência, isto pode comprometer uma boa experiência de uso por parte dos proprietários de aparelhos. Pode ser citado, por exemplo, o caso do Android que prioriza redes WIFI sem considerar o fator qualidade do sinal em relação a redes móveis. Estudos veem sendo realizados para o aperfeiçoamento de técnicas de escolha de redes, mas devido à quantidade de variáveis existentes, tais técnicas acabam se tornando muito complexas e perdendo a eficiência e a velocidade computacional.

Sendo assim prover um mecanismo de escolha de redes eficiente e com baixo custo computacional para dispositivos móveis é uma necessidade iminente. O sistema operacional Android tanto por ser uma tendência no mercado, quanto por possuir um ambiente de desenvolvimento mais propício à criação de aplicações com baixo custo, foi a plataforma escolhida para realização desse projeto.

A primeira parte deste estudo aborda trabalhos relacionados, onde são descritos trabalhos de conteúdos similares. Na parte seguinte é apresentada a proposta, expondo de maneira sucinta o projeto a ser desenvolvido.

Assuntos relacionados com ambiente de desenvolvimento e teste, requisitos necessários para a implementação e a arquitetura do projeto, estão descritos na terceira seção. Na quarta parte foi descrita a organização metodológica para a realização das experimentações. A quinta parte apresenta os resultados obtidos pelas sequencias de experimentações executadas. E por ultimo foi realizado um apanhado das principais impressões obtidas pela concretização deste trabalho.

2. Trabalhos Relacionados

O Android proporciona varias vantagens a quem deseja desenvolver para a plataforma, atraindo cada vez mais desenvolvedores e usuários de dispositivos que possuam o sistema operacional da Google. Na literatura recente podem ser encontrados artigos que tem como objeto de seu estudo o Android, um exemplo disto é [Cheng10], que propõe a modificação do Kernel do Android de modo a prover ao sistema operacional funções para gestão de serviços de telemática. Pretende-se alcançar, com essa união, um dispositivo amigável para ambientes de comunicação, unindo tráfego seguro, navegação GPS, atividades remotas, entretenimento com áudio e vídeo.

Ao desenvolver para dispositivos móveis uma preocupação constante de programadores é com o

desempenho de suas aplicações, visto que dispositivos embarcados, mesmo com uma grande evolução dos microprocessadores, ainda possuem poder de processamento bem abaixo de máquinas *desktop* e notebooks modernos.

O SDK padrão para desenvolvimento de aplicações no Android utiliza como linguagem de desenvolvimento *Java*. Esta linguagem, considerada lenta e consumidora excessiva de memória, é comumente questionando como linguagem para desenvolvimento mobile. [Lee10] Apresentam uma alternativa para o desenvolvimento na plataforma do *Android* que demonstra a utilização de outras linguagens como ferramenta de construção de aplicações para a plataforma.

A proposta visa à utilização do *Android* NDK, um ambiente de desenvolvimento, no qual utiliza *C/C++* como linguagem de desenvolvimento nativa. Essa alternativa é justificada pelo melhor desempenho alcançado em casos de execução de operações complexas como: simulações físicas e processamento de sinais, transpondo as limitações impostas pelo *Android SDK* e dando maior flexibilidade e liberdade ao desenvolvedor além, da possibilidade de reutilização de códigos legados escritos em *C/C++* para a plataforma móvel. O inconveniente da utilização do *Android NDK* é o fato dele tornar códigos mais complexos e nem sempre apresentam um ganho de eficiência satisfatório.

3. Proposta

Atualmente dispositivos móveis possuem mecanismos de escolha de redes ineficientes. Estudos vêm sendo realizados para melhorar esse fator, mas a complexidade atribuída a cada um deles diminui o interesse da indústria em implanta-las em seus dispositivos. A proposta central de nosso trabalho é desenvolver uma solução eficaz e de pouca complexidade para o sistema móvel *Android* de modo que seja facilmente implantado em qualquer dispositivo. Desta forma, este trabalho descreve a implementação do algoritmo proposto por [Monteiro10] em um ambiente *Android*.

Segundo [Monteiro10] técnicas utilizadas para mensurar a qualidade de vídeo (*Quality of Video* ou *QoV*) podem ser classificadas em subjetivas e objetivas. Técnicas subjetivas necessitam da intervenção do usuário, enquanto técnicas objetivas, por não necessitar da intromissão de um operador, são mais facilmente automatizadas, no entanto são mais ineficientes. Utiliza-se neste projeto, técnicas objetivas por ser de mais simples automação.

Técnicas objetivas de mensuração de redes para transmissão de vídeo podem ser divididas em três tipos de acordo com sua necessidade de informações sobre a mídia a ser transmitida: *Full Reference* (FR) quando necessita de todo o vídeo como parâmetro de avaliação, *Reduced Reference* (RR) quando necessita apenas de algumas informações e *No Reference* (NR) quando não necessita de informações dos vídeos para realizar a métrica. Em nosso estudo utilizou-se métricas do tipo NR tornando o processo menos complexo. Denominou-se então, para esse trabalho, utilizar-se-ia a técnica de mensuração da qualidade de vídeo (*QoV*) sem a necessidade de

informações sobre o vídeo sendo ou a ser transmitido pela rede (NR) de *QoV-NR*. Para implementar o *QoV-NR* [Monteiro12] desenvolveu uma técnica chamada *Network Quality Metric* (NQM). Consiste em, a partir de cálculos baseados em atraso, *jitter* e pacotes perdidos obtidos pelo utilitário *ping*, definir um valor no intervalo de 0 a 5, referente ao grau de qualidade do meio por onde os dados trafegam.

4. Implementação

4.1 Ambiente

Para desenvolvimento deste projeto, foi construído um ambiente que proporciona uma maior comodidade e agilidade do processo de criação. O Ambiente foi montado utilizando a distribuição *Linux Ubuntu 11.04*. A escolha pelo *Eclipse* como IDE em conjunto com o *Android Development Toolkit* (ADT) se deu pelo fato de ser a ferramenta fornecido e mantido pela própria *Google* como ferramenta oficial de desenvolvimento para *Android*. A versão do *Eclipse* utilizada no experimento foi a *Galileu 3.5.2.r352*, utilizando o *plugin* ADT versão 10.0.1.v201103111512.

O *Android SDK Vevision 7* foi utilizado como base em todo o processo de desenvolvimento, tornando assim o sistema compatível com a grande maioria dos dispositivos existentes. Para operações via console de acesso e modificação do sistema operacional foi usado *Android Debug Bridge* versão 1.0.26. Para concluir o Ambiente de desenvolvimento foi utilizado um o *Samsung Galaxy 5 GT-I5500B* com *Android Gingerbread 2.3.7* e modificação *CyanogenMod-7.2.0.RC0-galaxy5-MADTEAM* com conexões GPRS, EDGE, 3G, Wi-Fi 802.11 b/g e *Bluetooth 2.1*.

4.2 Requisitos

A métrica proposta pelo *QoV-NS* necessita dos dados gerados pela ferramenta *ping*. Para realizarmos um “*ping*” no *Android* precisamos executar chamadas em recursos do sistema através de métodos presentes no *Java*, já que este recurso não está presente na API do sistema da *Google*. O *Java* possui métodos que realizam a mediação entre a linguagem e os recursos do SO, tornando possível executar comandos como se o operador estivesse sobre o console do sistema operacional.

O problema neste procedimento é que o executável *ping* (*/system/bin/ping*), presente no *Android*, possui limitações para usuários normais de maneira que, para estes usuários, o procedimento só pode ser realizado em endereço local. Quando o procedimento é realizado com referência a um servidor de endereço externo, nenhum dos pacotes pode ser entregue. Desta maneira o aplicativo necessita ter privilégios de *super-usuário* (*root*) para que a execução do comando *ping* possa acontecer sem tal restrição. Por questão de segurança, o *Android* não permite que aplicativos executem ações como *super-usuário*. Existem várias técnicas de *rooting* de um dispositivo *Android*, em nosso projeto foi utilizado um método desenvolvido por [Shanks11], que se consiste em introduzir um “*su*” (binário responsável por executar comandos como *super-usuário*) modificado no núcleo do

sistema. Feito isso se pode-se invocar o *su* para realizar tarefas como *root*. Isso Torna possível a execução do *ping* sem restrições.

4.3 Arquitetura

O *Wireless Analyzer* (aplicativo desenvolvido neste trabalho) foi criado com a proposta de desenvolver um sistema que constantemente verifique a qualidade do sinal *wireless* (WIFI e Móvel) de um dispositivo móvel. Para isso foi utilizado a técnica QoVNS [Monteiro10]. O projeto, visando manter uma boa organização e separação funcional, foi então dividido em sete componentes, agrupados em dois pacotes. No pacote principal se encontram o *Inicializador*, *WAService*, *WifiService* e *MobileService*, já no pacote de utilitários temos o *SuDroid*, *PingParser* e o *Logger*, essa estrutura pode ser observada na figura 1.

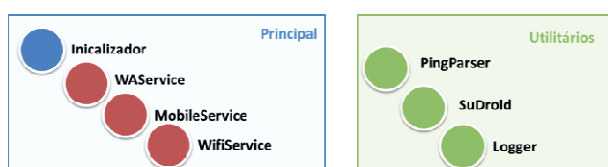


Figura 4. Arquitetura do aplicativo dividida em pacotes.

Logo que o aplicativo é iniciado o *Inicializador* efetua procedimentos de forma a tentar ativas as duas interfaces de redes (WIFI e 3G) simultaneamente. Pela grande variedade de nomes de interfaces existentes para cada dispositivo, é necessário que nessa etapa de inicialização o usuário classifique as interfaces identificadas como móvel ou como interface WIFI.

O Componente *SuDroid* é responsável pela chamadas de recursos do sistema e tratamento das mesmas. Provê uma interface simplificada e eficiente para executar comandos como *super-usuário*, sem necessidade de tratamento de exceções críticas. Esta classe retorna um objeto com informações sobre estado da requisição, em casos sucesso informa o retorno desejado, em casos de fracasso retorna uma mensagem de erro.

Já o componente *PingParser* pode ser considerado o núcleo do aplicativo, pois nele se encontram as métricas do QoV-NS e todos os métodos de requisição e tratamento do *ping*. Os recursos provenientes do componente *SuDroid* para realizar a comunicação direta com o Sistema Operacional. Para isto, ele informa como parâmetros de entrada o comando *ping* acrescido da quantidade de requisições a serem realizadas (parâmetro “-c”), o host de destino e a interface de saída (parâmetro “-I”). Ao se realizar o *ping* através do *SuDroid* com sucesso, o conteúdo do *ping* é armazenado em uma variável que é então submetida a uma sondagem utilizando expressões regulares, que extraem informações como atraso em cada requisição, pacotes perdidos e menor e maior atraso. De posse destas informações o componente então calcula outras variáveis necessárias como Atraso médio, *jitter*, *jitter médio*, *menor jitter*, *maior jitter*. Desta forma, é possível efetuar o calculo do NQM (Noise Quality Measure) [Monteiro10].

O *WAService* é responsável por orquestrar o funcionamento do *WifiService* e do *MobileService*, após iniciado ele ativa também seus serviços subordinados que realizam, independentemente, chamadas ao *PingParser* de modo que ele efetue o calculo do NQM e retorne o resultado obtido. Com este resultado a cada cinco aferições são calculadas as médias aritméticas e assim definida a qualidade da rede vinculada a cada interface. Quando a qualidade de sinal da interface é alcançada pelo *WifiService* ou pelo *MobileService*. O *WAService* então realiza uma comparação entre a qualidade do sinal obtida pelos dois serviços. Baseado nestes valores é realizada a sinalização, por meio de escrita em arquivo utilizando o componente *Logger*.

Tabela 3. Esquema de Mudanças baseado na Qualidade de Sinal.

WifiService	MobileService	Muda Para
< 3	< 3	Não Muda
< 3	>= 3	<i>MobileService</i>
>= 3	< 3	<i>WifiService</i>
>= 3	>= 3	Não Muda

A sinalização realizada pelo *Logger* é formatada utilizando do padrão 0 (zero) para Não realizar mudanças, 1 (um) para realizar o *handover* para a rede WIFI e 2 (dois) para realizar o *handover* para rede 3G. Tornando assim fácil para que outra ferramenta possa realizar o processo de *handover* baseado na qualidade de sinal calculada pelo *Wireless Analyzer*.

5. Metodologia e Experimentação

Os Experimentos foram realizados utilizando um cenário composto por uma rede WIFI, provida por um AP localizado dentro das dependências da IFTO Palmas, por uma rede de Terceira Geração (3G) provida por uma operadora de telefonia móvel e um dispositivo móvel. O dispositivo móvel em questão possui suporte tanto a interface WIFI quanto a interface 3G e durante o experimento possui conectividade ativa com ambas as redes.

O dispositivo móvel está equipado com o aplicativo proposto, que no decorrer do experimento monitora as duas interfaces. Neste cenário o aplicativo realiza constantes requisições ICMP para o servidor de DNS primário da *Google* no endereço 8.8.8.8, realiza os cálculos referentes à NQM e por fim gera o log de monitoramento e dispara o gatilho de mudança de rede, caso necessário.

Os experimentos foram realizados baseados em noventa pre-amostras iniciais coletadas em três pontos (trinta em cada), o primeiro foi próximo ao AP, o segundo, aproximadamente, 10 metros ao norte do AP e o segundo, aproximadamente, 10 metros ao sul do AP. Com base nos dados coletados e baseado em um nível de confiança de 95%, foi calculado o desvio padrão, onde o erro máximo admitido para a construção da foi de cerca de trinta por cento desse desvio padrão. Efetuando o calculo probabilístico de amostras foi determinada à quantidade mínima de coletas a serem realizadas e nos casos em que os valores de amostras para 3G e para WIFI forem

diferentes, assume-se a maior dentre eles para definir a quantidade testes a ser realizado naquele ponto.

6. Experimentos e Resultados

Após a bateria de teste realizada, com cerca de três mil *pings*, o resultado dos cálculos NQM e das trocas realizadas foram processados e representados em forma dos gráficos apresentados a seguir. O gráfico exibido na Figura 2 representa a variação dos NQM's em relação às iterações a cada 10 *pings* para as coletas realizadas no ponto central. Os pequenos picos apresentados pela linha amarela (muda 3G) e verde (muda WIFI) representam o teste de qualidade realizado a cada cinco iterações de NQM. Caso a mudança seja necessária para alguma das interfaces o pico apresentado por estes será bem maior, como pode ser visto na Figura 3.

Em ambas as interfaces houve poucas variações relevantes. Considerando, ainda, que para que haja a sinalização para troca de interface preferencial é necessário que a média de cinco iterações seja menor que três, o que não chegou a acontecer em nenhuma das interfaces.

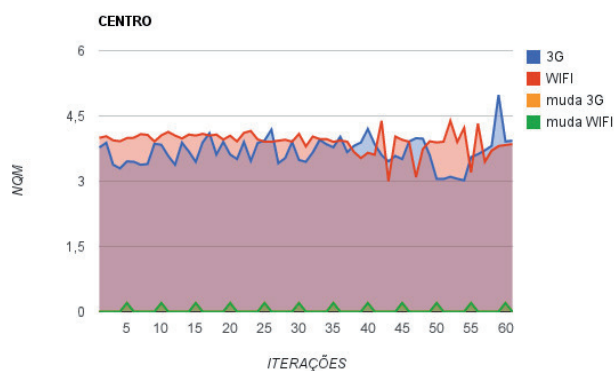


Figura 5. Coleta de NQM junto ao AP.

Na Figura 3 que representa as coletas realizadas ao norte do AP pode ser observado o melhor funcionamento do aplicativo, pois são verificadas maiores variações relevantes dos dados coletados. No início da coleta a rede WIFI apresenta um início com valores abaixo dos dois pontos de NQM e logo em seguida chega à zero por sete iterações o que, obviamente, resulta em uma média na primeira aferição de menos de três pontos. Diante desse resultado o aplicativo sinaliza a mudança para rede preferencial 3G.

No sexto cálculo de qualidade a rede 3G apresenta uma queda brusca o que resulta em troca para rede WIFI, mesmo com certa instabilidade nesta rede. Isso ocorre, pois mesmo em tal circunstância sua média consegue alcançar um valor a cima dos três pontos da escala NQM.

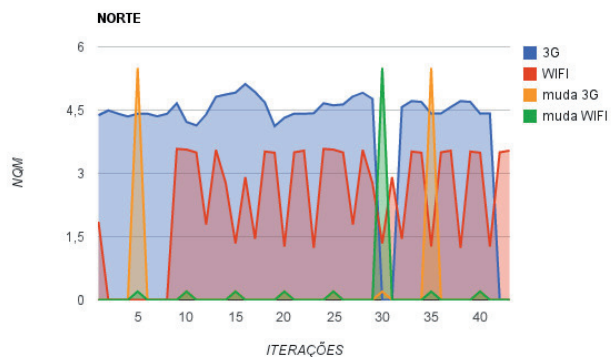


Figura 6. Coleta realizada a 10m ao Norte do AP.

Os dados coletados no ponto sul do AP estão representados no gráfico exibido na Figura 4, que por sua vez demonstra um comportamento diferente das demais coletas. Neste terceiro experimento a rede 3G obteve resultados irregulares e na grande maioria das iterações, abaixo de três pontos na escala NQM. Isso resultaria em mudança para rede WIFI se ela não estivesse, por convenção, iniciada como rede preferencial. Mesmo quando em três momentos a rede WIFI atinge valores muitos baixos de NQM a troca não existe, pois os valores NQM obtidos da interface 3G não são suficientes para que está possa vir a ser a rede preferencial, ou seja, não atinge valores médios acima de três pontos.

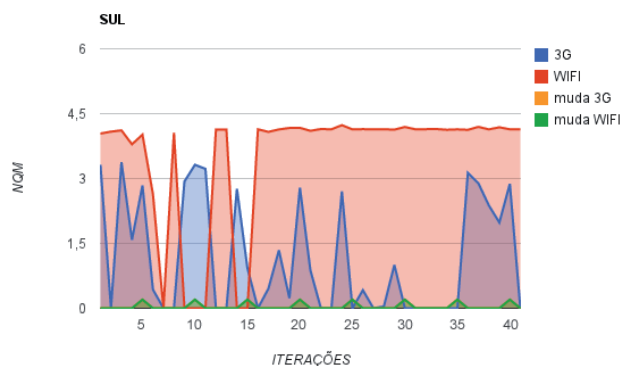


Figura 7. Coleta Realizada 10m ao Sul do AP.

Finalizada a bateria de experimentos observamos um bom funcionamento do aplicativo em diversas situações possíveis dentro de um cenário onde o tráfego de fundo não foi controlado, assim uma serie de adversidades reais puderam ser estudadas e observadas durante os testes. Resultado disso foi uma sinalização apropriada para a realização trocas de redes sempre que o valor NQM atingia o limiar de *handover* proposto.

7. Conclusões e trabalhos futuros

Este artigo apresenta um estudo sobre o desenvolvimento de uma aplicação para o sistema operacional *Android*, que possa a partir da métrica de QoS NQM, avaliar e decidir pela troca de, maneira eficiente e com baixo custo computacional, dentre duas redes disponíveis. O projeto teve início com o estudo para a escolha da melhor tecnologia para implementação da equação para obtenção de valores NQM. Seguido então pela implementação de um algoritmo para que pudesse monitorar múltiplas interfaces simultaneamente. Com obtenção de uma versão

estável do *software*, foi construído um ambiente para realização de experimentos, apoiados por cálculos estatísticos, que por final comprovaram a eficiência do aplicativo em diversas situações de adversidades reais presentes em um meio *wireless*.

Como aperfeiçoamento deste projeto é planejado a realização do *handover* embutida no próprio aplicativo, que oferece a principio, o módulo de classificação e decisão, mas não o executor. A execução do *handover* fica, então, a cargo de um protocolo externo que ofereça todas as garantias necessárias para que não haja perdas de conexão durante este processo.

Referências bibliográficas

[Cheng10] Cheng Y.H.; Kuo W.K.; Su S.L. (2010). An Android System Design and Implementation for Telematics Services, Intelligent Computing and Intelligent Systems (ICIS),IEEE International Conference on, vol. 10, pp. 206-210.

[Lee10] Lee, S.; Jeon J.W. (2010) Evaluating Performance of Android Platform Using C for Embedded Systems, International Conference on Control, Automation and Systems, vol. 8, pp. 1160 - 1163.

[Monteiro12] Monteiro, C. C. (2012). Um ambiente para apoio à integração de redes sem fio heterogêneas. 186 f. Tese (Doutorado em Engenharia Elétrica) - Universidade de Brasília, Brasília, Brasil.

[Monteiro10] Monteiro, C. C.; Gondin, P. R. L. (2010). An Alternative QoS Architecture for Integrating WLAN/3G Networks. Fed. Inst. of Educ. of Tocantins, Palmas, Brazil. in: Wireless and Mobile Communications (ICWMC), 2010 6th International Conference on, pp 68-77.

[Shanks11] Shanks, A. (2011). su-binary. <http://github.com/ChainsDD/su-binary>