

Join-Me: Uma arquitetura para a integração entre operadores de redes móveis e provedores de serviços

Alexandre Martins G. de Deus¹, Eduardo Pagani Julio¹, Marcelo F. Moreno¹

¹Programa de Pós Graduação em Ciência da Computação
Universidade Federal de Juiz de Fora (UFJF)
Caixa Postal 36036-900 – Juiz de Fora – MG – Brazil

alexandremartins@ice.ufjf.br, eduardo.pagani@ice.ufjf.br, moreno@ice.ufjf.br

Abstract. *The search for Quality of Experience is a constant challenge in mobile network scenarios. However, the increasing complexity of services requires an integration between service providers (SPs) and mobile network operators (MNOs). The present work proposes the Join-Me architecture, allowing components of the services of the SPs, once internalized to the MNOs infrastructure, to be managed autonomously, allowing the delivery of services as close as possible to the mobile user.*

Resumo. *A busca por Qualidade de Experiência é um desafio constante em cenários de redes móveis. No entanto, a crescente complexidade dos serviços demanda uma integração entre provedores de serviços (SPs) e operadores de redes móveis (MNOs). O presente trabalho propõe a arquitetura Join-Me, permitindo que componentes dos serviços dos SPs, uma vez internalizados à infraestrutura dos MNOs, sejam geridos de forma autônoma, permitindo a entrega de serviços o mais próximo possível do usuário móvel.*

1. Introdução

Com o aumento de *smartphones* e de conteúdo multimídia, o tráfego de dados em redes móveis tem crescido consideravelmente, e a Qualidade de Experiência (QoE) do usuário se torna fundamental na entrega de serviços na Internet. Provedores de Serviços (SPs) terceirizam serviços de Redes de Distribuição de Conteúdo (CDNs), para descentralizar a entrega dos dados e oferecê-los mais próximo do usuário.

Operadores de Redes Móveis (MNOs) apostam na utilização de tecnologias como a *Multi-access Edge Computing* (MEC) para descentralizar o processamento das aplicações de rede, trazendo-o para junto das Estações Rádio Base (ERBs) e agilizando a instanciação de funções da própria rede sob demanda.

Este artigo apresenta a *Join-Me*, uma arquitetura integrando MNOs e SPs, objetivando permitir que componentes dos serviços dos SPs, uma vez internalizados, sejam geridos de forma autônoma e dinâmica pelos MNOs, possibilitando a entrega de serviços mais próximo do usuário. O trabalho está organizado como segue: a Seção 2 traz uma discussão sobre os trabalhos relacionados à presente proposta; a Seção 3 apresenta detalhes da arquitetura; Sua aplicabilidade por meio de Prova de Conceito é demonstrada na Seção 4. Os resultados são apresentados na Seção 5, e a 6 dedica-se à conclusão e trabalhos futuros.

2. Trabalhos Relacionados

Tecnologias são criadas para lidar com requisitos de novos segmentos. A *Multi-access Edge Computing* (MEC) [ETSI 2015], possibilita a redução da latência, com recursos computacionais junto às Estações Rádio Base (ERBs). Já o *Network Slicing* [Rost and Mannweiler 2017] no 5G [Gupta and Jha 2015], instancia segmentos virtuais de rede visando Qualidade de Serviço (QoS) para aplicações de diferentes propósitos.

Visto que o uso de *Application Program Interfaces* APIs tem se tornado tendência, o *Common API Framework for 3GPP Northbound APIs* (CAPIF) [3rd Generation Partnership Project 2018] foi desenvolvido para padronizar e hospedar APIs de maneira centralizada.

O uso das NFVs, [Jahromi and Glitho 2018] possibilita o provisionamento de componentes de CDN, criando servidores adicionais à rede CDN nas instalações do *Internet Service Provider* (ISP), direcionando o tráfego para o servidor mais próximo.

O conceito *Follow-Me Cloud* [Taleb et al. 2016], apresenta o modo que serviços em nuvem sejam migrados suavemente entre *datacenters* federados, sem interrupção dos mesmos e à medida que os usuários se movem, visando poupar recursos computacionais e otimizando o desempenho durante a migração. A necessidade de parte do *framework* ser implementada no dispositivo do usuário móvel deixa em aberto uma questão de privacidade do usuário.

Diversos trabalhos têm a melhoria da QoE do usuário como objetivo. No entanto, não há um trabalho conjunto entre i) garantir que os recursos computacionais sejam devidamente alocados; ii) dar atenção aos serviços SPs e iii) garantir a privacidade do usuário. A *Join-Me* visa harmonizar esses três pontos, a fim de possibilitar o trabalho conjunto dos SPs e MNOs.

3. A arquitetura *Join-Me*

É possível notar uma separação entre SPs e MNOs pois, independente dos esforços realizados por ambas as partes, toda requisição do usuário obrigatoriamente necessita percorrer o caminho pela infraestrutura do MNO até os servidores dos SPs.

Dado esse cenário, a arquitetura "*Join-Me*" permite que os SPs internalizem parte de seus serviços, para dentro da infraestrutura dos MNOs, por meio de uma API. Uma vez dentro da infraestrutura, o MNO se encarregará de suprir as demandas e gerenciará a rede para entregar o serviço o mais próximo possível do usuário, auxiliado por informações já presentes internamente em sua infraestrutura, como o histórico de deslocamento. Por outro lado, os Operadores de Redes Móveis poderão cobrar pelos serviços em execução e pela alocação de recursos.

Visando a privacidade do usuário, a arquitetura *Join-Me* se apoia na técnica de *Network Slicing* e no gerenciamento autônomo dos módulos para isolar aplicações. Uma vez adotado por um MNO, *Join-Me* pode ser suportada pela CAPIF e ter suas APIs descobertas por SPs e MNOs clientes. As APIs *Join-Me* não estão no escopo deste trabalho, o qual se limita à proposição da arquitetura e a sua prova de conceito.

3.1. Componentes da arquitetura

Os componentes da a arquitetura *Join-Me* são apresentados na Figura 1. No *Core* da rede, a *Join-Me* API atua como interface entre SPs e MNOs, o *Service Provider Account Manager* registra novos SPs e o gerencia o controle de acesso de seus operadores. O *Service Provider Authentication* provê autenticação dos SPs permitindo acesso aos componentes de *Join-Me*. Uma vez internalizados, os módulos que compõem serviços dos SPs são armazenados no *Module Repository* e posteriormente o *Module Orchestrator* os coloca em execução. Os módulos passam a utilizar recursos dos MNOs, os quais tem sua utilização contabilizada pelo *Billing Agent*. As APIs da arquitetura são disponibilizadas pelo CAPIF.

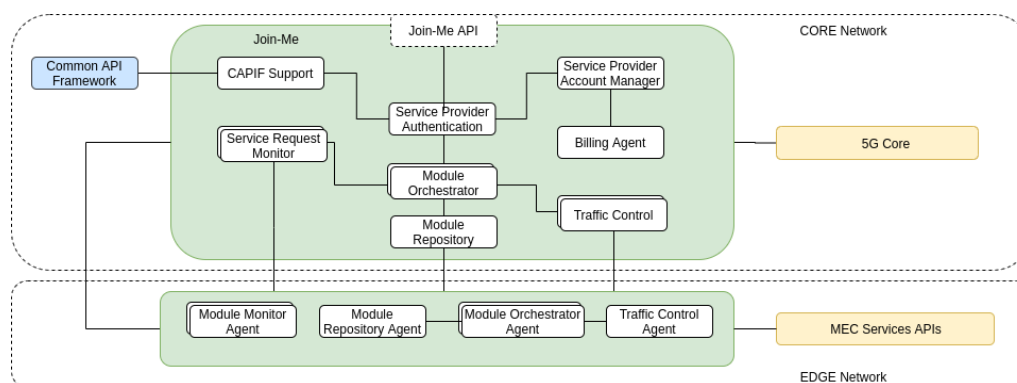


Figura 1. Arquitetura *Join-Me* e seus respectivos módulos

O *Service Request Monitor* recebe requisições dos usuários e informa a borda sobre o módulo do serviço solicitado. Ao receber tal informação, o *Module Repository Agent* copia do *Core* os módulos a serem instanciados e os armazena. O *Module Orchestrator Agent* disponibiliza o serviço em execução e o *Traffic Control* direciona as requisições para a borda. O *Module Monitor Agent* passa a monitorar as requisições ao serviço e, caso os recursos da borda estejam ociosos, o *Traffic Control Agent* redireciona as requisições de serviço de volta ao *Core*, onde estará disponível constantemente.

4. Prova de Conceito

Para demonstrar a viabilidade da arquitetura proposta, foi elaborada uma Prova de Conceito (PoC - *Proof of Concept*) para simular casos de uso relacionados à acomodação de serviços terceiros dentro da infraestrutura dos MNOs para as próximas gerações de redes móveis. O objetivo é demonstrar o comportamento da rede e o consumo de recursos na arquitetura. O cenário (Figura 2) consiste em um servidor para o *Core* da rede, duas ERBs (cada uma composta por um servidor se comportando como MEC) e por um Roteador *Wi-Fi* com OpenWRT¹, a fim de fornecer um ponto de acesso para usuários móveis.

4.1. Funcionamento

Considera-se ERB um conjunto de um servidor MEC e um *Access Point* (AP). Assim, o servidor MEC será referenciado como ERB-MEC, enquanto o AP será ERB-AP. Assume-se que os módulos dos SPs já estão em execução no servidor *Core*. O diagrama de sequência da Figura 3 ilustra as atividades realizadas na PoC.

¹OpenWRT Project - <https://openwrt.org/docs/start>

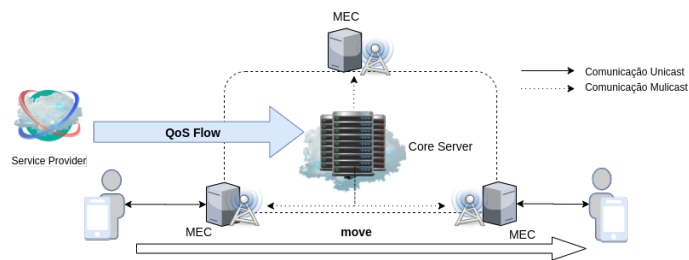


Figura 2. Cenário Prova de Conceito - Join-Me

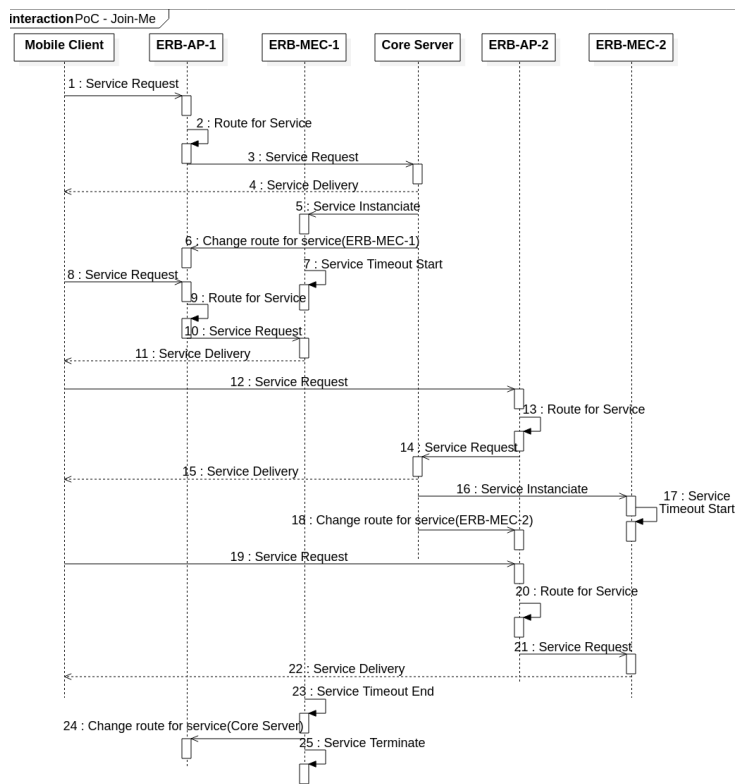


Figura 3. Diagrama de Sequência PoC - Join-Me

Considerando o "Mobile Client" conectado à ERB-AP-1, inicia-se a requisição do serviço que se encontra disponível apenas no core, conforme o passo 1. Sendo assim, a requisição segue até o mesmo (passo 3) e a resposta é devolvida ao usuário no passo 4. O *Service Request Monitor* aciona o *Module Orchestrator Agent* para instanciar uma cópia do(s) módulo(s) na ERB-MEC-1 (passo 5). Assim que o(s) módulo(s) já esteja(m) instanciado(s) na ERB-MEC-1 (passo 6), o *Traffic Control* altera a rota presente na ERB-AP-1 referente ao serviço requisitado para a ERB-MEC-1. Observa-se que o passo 7 é iniciado pelo *Module Monitor Agent* como um processo de *timeout* para o(s) módulo(s) instanciado(s) em ERB-MEC-1. Nos passos 8 a 11 as requisições originadas na ERB-AP-1 passam a ser atendidas pela ERB-MEC-1 e não mais pelo Core.

Ao realizar o *handover* da ERB-AP-1 para a ERB-AP-2, o processo de requisição de serviço se repete nos passos 12 ao 22 para a sua respectiva ERB (ERB-MEC-2). Quanto a economia de recursos, percebe-se o fim do processo de *timeout* (passo 23) ocorrido devido à ociosidade do(s) módulo(s), uma vez que não há mais requisições originadas

na mesma. Sendo assim, o *Traffic Control Agent* altera novamente a rota presente na ERB-AP-1 para o servidor *core* e então o *Module Monitor Agent* aciona o *Module Orchestrator Agent*, que finaliza a execução do(s) módulo(s), (passos 24 e 25).

5. Resultados

Para a experimentação da PoC, foi elaborado um caso de uso utilizando um *Streaming* de vídeo ao vivo como serviço, o qual possui três módulos, o *sp-live-streaming* o qual envia o vídeo em um único fluxo com suas características originais por meio de *Real Time Protocol (RTP) (multicast)*. A segmentação do vídeo é realizada pelos *sp-segmentation* nas resoluções 360p, 480p e 720p. Já o *sp-delivery* disponibiliza os segmentos ao usuário.

No experimento, o fluxo RTP *multicast* gerado externamente é recebido com a QoS necessária pela arquitetura, replicado-o para os segmentadores. A entrega em HTTP é realizada pelo *Core* da rede. Não há movimentação do usuário, observando apenas o consumo de recursos pelos módulos. Os segmentadores de resoluções superiores consomem mais recursos, uma vez que não há perda de detalhes dos *frames*. Focou-se apenas no tráfego HTTP, o qual está relacionado à entrega do vídeo. O *sp-delivery* teve seu consumo ínfimo. Como sua comunicação é *Unicast*, seu consumo tende a crescer linearmente em função do número de usuários, além do aumento do tráfego HTTP (Figura 4).

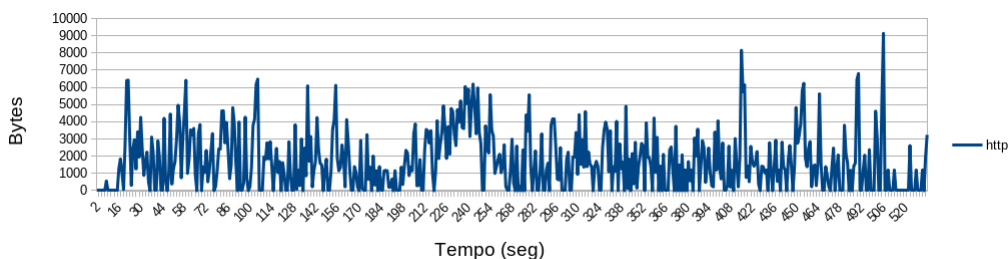


Figura 4. Tráfego HTTP - Core Join-Me

Posteriormente o serviço continuou no *core* da rede mas com a movimentação do usuário. Ignorou-se pequenas interrupções e observou-se que, devido a alteração no sinal *wireless* durante a movimentação, a perda de qualidade dos segmentos. Utilizando a PoC de Join-Me, a entrega do serviço se inicia no *Core* da rede e posteriormente segue todo o *workflow* apresentado na Figura 3. Na Figura 5 é possível observar o serviço acompanhando o usuário. Observa-se o alívio de tráfego no *core* da rede em comparação com o tráfego inicial do experimento (Figura 4).

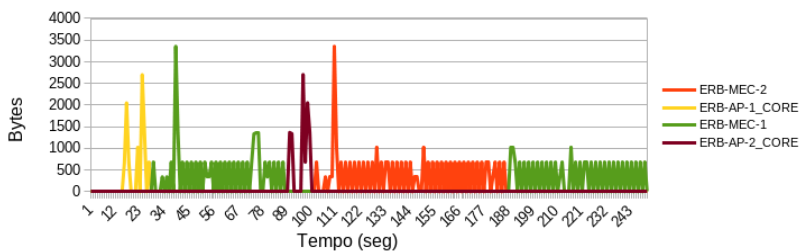


Figura 5. Tráfego HTTP e migração do serviço - Arquitetura Join-Me

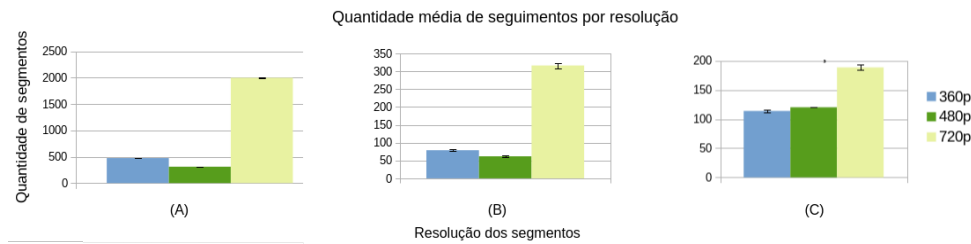


Figura 6. Análise de QoE - (A) Core Join-Me - (B) ERBs Join-Me - (C) Fora da arquitetura Join-Me

A QoE do usuário (Figura 6A), o experimento não sofreu mudanças relevantes, obedecendo o mesmo comportamento do *core* (Figura 6B). Se faz necessário a realização de experimentos com um maior número de usuários, para observar alterações com a elevação do tráfego HTTP. Por fim, o serviço com seus respectivos módulos foi analisado sem o suporte de Join-Me, (Figura 6C) para confrontar com os resultados da PoC. Observou-se um número maior de segmentos de baixa e média resolução em comparação a arquitetura *Join-Me*, devido não apenas a fatores relacionados ao sinal *wireless*, mas também às condições externas à rede local.

6. Conclusão

Este trabalho apresentou a arquitetura *Join-Me*, que gerencia módulos de serviços de Provedores de Serviços internalizados à infraestrutura dos Operadores de Redes Móveis. Verificou-se um ganho de QoE utilizando a arquitetura, zelando pela economia de recursos e respeitando a privacidade do usuário. Uma prova de conceito foi apresentada para exemplificar a sua utilização. Como trabalhos futuros, serão desenvolvidos os demais componentes apresentados neste trabalho e as APIs voltadas para a interação entre SPs e MNOs, incluindo o monitoramento dos serviços e recursos alocados.

7. Agradecimentos

Este trabalho contou com o apoio da Rede Integrada de Pesquisa em Alta Velocidade (RePesq) <https://repesq.ufjf.br> da Universidade Federal de Juiz de Fora.

Referências

- 3rd Generation Partnership Project (2018). Functional architecture and information flows to support common api framework for 3gpp northbound apis. pages 1–189.
- ETSI, G. (2015). 004, mobile edge computing (mec) service scenarios v1. 1.1,(2015).
- Gupta, A. and Jha, R. K. (2015). A Survey of 5G Network: Architecture and Emerging Technologies. *IEEE Access*, 3:1206–1232.
- Jahromi, N. T. and Glitho, Roch H, e. a. (2018). An NFV and microservice based architecture for on-the-fly component provisioning in content delivery networks. In *15th IEEE Annual Consumer Communications and Networking Conference*, pages 1–7.
- Rost, P. and Mannweiler, Christian, e. a. (2017). Network Slicing to Enable Scalability and Flexibility in 5G Mobile Networks. *IEEE Communications Magazine*, 55:72–79.
- Taleb, T., Ksentini, A., and Frangoudis, P. (2016). Follow-Me Cloud: When Cloud Services Follow Mobile Users. *IEEE Transactions on Cloud Computing*.