
Implementação de um jogo distribuído numa Rede Veicular (Rede Oportunista/Tolerante a Atrasos) com Dados Nomeados

1. Introdução

A arquitetura da Internet não foi concebida a pensar em muitos dos requisitos que se colocam à “Rede” hoje em dia, nomeadamente a mobilidade dos seus utilizadores. De acordo com o protocolo IP, o encaminhamento dos pacotes na rede é efetuado tendo por base o endereço IP do destinatário, endereço esse está incluído no cabeçalho do pacote IP. Os endereços IP, por sua vez, são atribuídos aos sistemas terminais de acordo com a rede a que estes se ligam. Por esse motivo, quando um utilizador se desloca de uma rede para a outra, tem que adquirir um novo endereço IP, que seja válido na nova rede. Como é fácil de perceber, esta forma de funcionamento teve de ser adaptada e em muitos casos tiveram que ser desenvolvidas soluções completamente novas e disruptivas com os princípios adotados pela arquitetura original da Internet.

As redes *ad-hoc*, por exemplo, são redes de formação espontânea que não necessitam de infraestruturas e que utilizam uma solução diferente da usada pelas redes IP infraestruturadas, para o encaminhamento de pacotes. Além disso, suportam naturalmente a mobilidade dos seus nós. Ao contrário das redes infraestruturadas, em que existe uma separação clara entre as funções desempenhadas por um sistema terminal e aquelas que são desempenhadas por um router, nas redes *ad-hoc* não há nós com um papel especial e todos precisam de ajudar a fazer tudo. Ou seja, o dispositivo de um utilizador deve ter a capacidade de encaminhar pacotes de outros nós, portando-se como um router. Se os nós que constituem a rede *ad-hoc* forem nós móveis, ou seja, com capacidade para se moverem de forma independente em qualquer direção, a rede designa-se por MANET (ou seja Mobile Ad hoc Network) e nesse caso, a rede apresenta uma topologia dinâmica sendo que, cada nó em movimento irá alterar as suas ligações a outros dispositivos ao longo do tempo. Para lidar de forma eficiente com esta questão, os algoritmos de encaminhamento de tráfego tiveram que ser profundamente adaptados.

As DTNs, redes tolerantes a atrasos ou redes oportunistas, são também redes de formação espontânea (*ad-hoc*), mas além disso são redes que rompem com o modelo de conectividade fim-a-fim, admitindo que a comunicação pode ser interrompida, sofrer atrasos longos e retomada mais tarde de acordo com contatos oportunistas entre os nós da rede. Este conceito de conectividade intermitente, muitas vezes caracterizado como *store-carry-and-forward*, pode ser aplicado no contexto de redes veiculares, por exemplo: um veículo recebe informação que armazena e transporta para posteriormente reenviar para outros veículos com que, oportunisticamente, se cruzem.

As NDNs (Named Data Networks), por outro lado, constituem uma das arquiteturas de Rede propostas para a Internet do futuro, e apresentam uma solução muito interessante para suportar a mobilidade dos utilizadores. Este novo paradigma baseia-se na subscrição e publicação de conteúdos. Os utilizadores (consumidores de informação) apenas têm que saber qual o conteúdo (nome) pelo qual têm interesse, não sendo necessário conhecer a sua localização (endereço). Para obter a informação que deseja, o consumidor propaga um pacote de interesse através da rede. Quando este atinge um nó produtor ou um nó intermédio, que possua o conteúdo em cache, os dados são devolvidos pelo caminho inverso, num pacote de dados.

Com tantos e tão diferentes paradigmas e arquiteturas de rede, muitas vezes, a forma como concebemos e desenvolvemos uma aplicação distribuída também tem de mudar. Além de ser necessário considerar as questões de desempenho que poderão afetar de forma muito significativa o funcionamento das aplicações, o

modelo que serve de base à construção da aplicação sofre, muitas vezes, alterações substanciais. O objetivo deste projeto é avaliar o impacto que os novos paradigmas/arquiteturas de rede podem ter nas aplicações de rede. Para isso propõe-se o desenvolvimento de um jogo simples para vários jogadores ligados através de uma rede de comunicações. Primeiro o jogo deverá ser desenvolvido usando uma rede IPv6 infraestruturada, que se estenderá numa segunda fase para uma rede veicular ad-hoc, em configuração de rede tolerante a atrasos (DTN, Delay Tolerant Network). O paradigma NDN (Named Data Networks) poderá, opcionalmente, ser aplicado numa terceira fase do projeto.

2. Descrição

Apresenta-se na Figura 1 o contexto genérico do projeto que consiste numa rede infraestruturada e numa rede veicular, admitindo que os veículos estão equipados com OBUs (On Board Units) que têm capacidade WiFi.

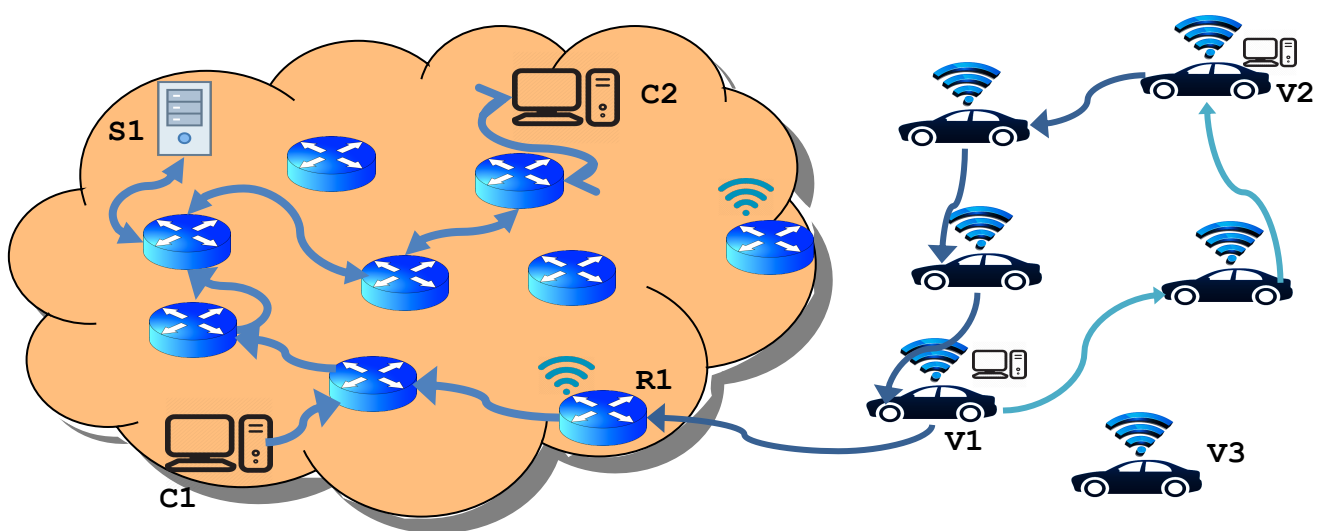


Figura 1. Comunicações na rede veicular e acessos à rede infraestruturada

Neste contexto, os veículos podem comunicar diretamente entre si e com a infraestrutura, sempre que o alcance rádio da interface sem fios o permitir, mas também indiretamente, a mais que um salto de distância, desde que todos se comportem como routers. Para tal devem ser capazes de correr protocolos de encaminhamento e reenviar pacotes que não lhe são destinados. Dado que os veículos se movimentam, também podem ter conectividade intermitente. Assume-se que toda a rede suporta no mínimo conectividade em IPv6, quer na parte de infraestrutura quer na parte da rede veicular. O protocolo IPv6 é pois o denominador comum de toda a infraestrutura.

No cenário ilustrado na Figura 1, podemos observar que o veículo V1 consegue comunicar diretamente a um salto de distância com o router R1 e através dele com toda a rede infraestruturada, ao passo que o veículo V2 precisará de usar dois ou mais veículos como nós intermédios. Por seu lado o veículo V3 está temporariamente sem conectividade, podendo vir a adquiri-la novamente quando se aproximar de V1 ou R1.

Além de se comportarem como routers sem fios, os veículos também executam aplicações. As aplicações veiculares podem ser classificadas em três grupos de acordo com seu objetivo principal: (i) segurança rodoviária, (ii) eficiência rodoviária e (iii) conforto/entretenimento de tripulantes e passageiros. As aplicações de segurança têm como principal objetivo aumentar a proteção das pessoas dentro do veículo, bem como do próprio veículo. São aplicações que contribuem para salvar vidas, evitando ou minimizando os efeitos dos

acidentes. Recorrem normalmente apenas a comunicações diretas, a um salto de distância, e têm restrições de tempo de atraso. As aplicações que se preocupam com a eficiência rodoviária têm como principal propósito melhorar os fluxos de tráfego, evitando situações de congestionamento de tráfego e contribuindo para um menor consumo de combustível. Por fim, as aplicações de conforto e entretenimento estão mais focadas em garantir que a viagem seja mais agradável para o condutor e/ou os passageiros. As aplicações deste tipo podem geralmente tolerar atrasos maiores do que as aplicações de segurança e eficiência.

Neste projeto pretende-se desenhar, implementar e testar uma aplicação dita de entretenimento, tendo em conta os novos paradigmas de rede, partindo de uma versão tradicional simples. O objetivo principal é fazer evoluir a versão inicial de forma adequada e justificada, face aos desafios da mobilidade e da conectividade intermitente.

A aplicação fica em aberto e pode ser proposta por cada grupo, mas sugere-se, por exemplo, um jogo simples interativo que envolva dois ou mais jogadores. O jogo deve ser desenvolvido com base num servidor que poderá ser usado apenas como “ponto de encontro” ou ir mais além e mediar as interações entre os diferentes jogadores. O jogo não terá que ser usado exclusivamente numa rede veicular, podendo ser um jogo genérico (simples), que pode ser executado a partir de nós na rede veicular ou a partir de nós na rede infraestruturada. No entanto, o servidor do jogo deverá ser sempre executado a partir de um nó da rede infraestruturada.

Embora seja tolerante a atrasos, o jogo deve ainda assim ter alguns requisitos temporais. Cada grupo de trabalho tem a liberdade de definir o jogo que pretende implementar, porque isso pode facilitar a forma como pensam no problema. No entanto o foco da avaliação não é o jogo em si, mas a forma como o mesmo é adaptado aos novos paradigmas. A estética gráfica e o grau de dificuldade do jogo em si são elementos acessórios.

3. Planeamento e descrição das várias tarefas

3.1 Conceção e desenvolvimento da aplicação numa rede IPv6 infraestruturada.

Nesta primeira tarefa pretende-se que os alunos concebam e desenvolvam uma aplicação de entretenimento, usando um paradigma tradicional. A aplicação deve ser testada e demonstrada na rede infraestruturada sobre IPv6.

A rede infraestruturada deve suportar IPv6 e deve incluir alguns routers também equipados com interfaces WiFi (para comunicação com os veículos). Considere que a gama de endereçamento IPv6 disponível para toda a rede é 2001:0690:2280:0820::0/60 e que a topologia de rede infraestruturada tem 8 (ou mais) routers IPv6, sendo que 2 (ou mais) deles estão equipados com ethernet e WiFi. Convirá notar que em IPv6 não existem endereços de broadcast mas existem endereços multicast que podem ser utilizados no âmbito do projeto para atingir todos os nós e todos os routers (recorde que o endereço de grupo FF02::1 tem a abrangência de todos os nós atingíveis em link local).

Resultado esperado: uma demonstração do jogo na rede infraestruturada, com um mínimo de dois jogadores.

Etapas recomendadas: i) configuração da rede; ii) teste de conectividade IPv6 (ping/traceroute); iii) execução do servidor de jogo; iv) execução da aplicação do jogo (jogador 1); v) execução da aplicação do jogo (jogador 2);

Exemplo meramente ilustrativo (quiz): O jogo é um quiz de perguntas e ganha o jogador que responder a mais perguntas certas, numa série de N perguntas. Em cada pergunta, de escolha múltipla, apenas uma das opções que está correta. A pergunta é um texto. A resposta um número. Só o jogador mais rápido a escolher a resposta certa é que ganha os pontos relativos a essa pergunta. A lógica do jogo e a comunicação é totalmente gerida pelo servidor do jogo. Os jogadores correm uma aplicação cliente muito simples, que se liga ao servidor indicando, por exemplo, o nome do jogador. Quando o jogo começa, o servidor prepara uma série de N questões, e envia a primeira pergunta numa mensagem para a todos os jogadores que estejam online. Depois fica à espera das respostas dos jogadores, até receber pelo menos uma resposta certa, ou até acabar o tempo dessa pergunta. Quando receber uma resposta certa de um jogador, acumula pontos para esse jogador, e envia uma notificação para todos. De seguida envia a próxima pergunta e repete o processo. No final das N perguntas declara vencedor o jogador com mais pontos. O servidor do jogo (ex: quiz-server) fica à escuta de ligações de clientes num socket predefinido e os clientes do jogo (ex: quiz) ligam-se a esse socket por IPv6.

3.2 Estabelecimento de uma rede veicular ad-hoc, em configuração de rede tolerante a atrasos

Nesta segunda tarefa, pretende-se que os alunos preparem o jogo para poder funcionar agora também na rede veicular, num cenário de rede ad-hoc com conectividade intermitente. A conectividade intermitente deve acontecer pelo movimento dos veículos, embora possa ser simulada manualmente aproximando ou afastando os veículos. O servidor do jogo mantém-se instalado e a funcionar na rede fixa. Os jogadores ou estão na rede fixa ou nos veículos.

Note-se que a versão desenvolvida na etapa anterior pode funcionar sem modificações na rede veicular, com IPv6, desde que todos os veículos tenham sempre conectividade com o servidor. Bastará configurá-los para modo ad-hoc IPv6 e mantê-los em alcance rádio. No entanto o objetivo é que se possa jogar mesmo que durante períodos de tempo não tenham nenhuma conectividade. E para isso já é necessário implementar uma lógica adicional DTN em todos os nós móveis.

As DTNs pretendem resolver os problemas associados a contextos de comunicações com longos atrasos e em que as quebras de conectividade são frequentes. Um exemplo típico será um conjunto de nós wireless em mobilidade (por exemplo veículos) que, em consequência da sua dinâmica, frequentemente estabelecem, ou quebram, as comunicações com outros nós vizinhos (sejam estes outros nós móveis do mesmo tipo, ou infra-estruturados). Muitas vezes os contatos entre nós são esporádicos e de duração limitada, pelo que a estratégia habitual de um *router* (armazena-e-envia, *store-&forward*) em redes IP(v6) não é uma solução satisfatória; preferencialmente um nó DTN utiliza uma outra estratégia, que genericamente, designaremos por *armazena-transporta-e-envia*, *store-carry-&forward*, em que os dados são armazenados num nó e por ele transportados para, eventualmente, serem reenviados num instante futuro.

O protocolo de encaminhamento DTN irá servir para tornar possível a comunicação entre os vários jogadores que estão na rede veicular, bem como a comunicação entre estes e o servidor e outros jogadores que estão na rede infra-estruturada. O protocolo de encaminhamento para DTNs mais simples é o protocolo epidémico. Este protocolo permite a entrega de pacotes ao destino sem qualquer conhecimento acerca da topologia e mesmo que não exista um caminho direto entre a fonte e o destino. Na ausência de contacto, os pacotes são armazenados numa cache dos routers. O processo de troca de pacotes é repetido sempre que existe contacto com outro router. Cada mensagem contém um identificador global e único para determinar se esta já está armazenada ou se já foi encaminhada. Quantas mais mensagens forem distribuídas pela rede maior será a probabilidade de entrega e menor será o seu atraso. Este protocolo apresenta custos muito elevados em termos de número de transmissões e espaço de memória necessário. A partir deste protocolo foram propostos muitos outros que pretendem de alguma forma contribuir para minimizar as limitações do protocolo epidémico mantendo níveis elevados de taxa de entrega.

O principal objetivo desta fase é conceber e implementar uma estratégia de encaminhamento DTN que permita o funcionamento do jogo desenvolvido na fase anterior, na rede veicular.

Resultado esperado: uma demonstração do jogo na rede veicular, com um mínimo de dois jogadores nos veículos, com conectividade intermitente.

Etapas recomendadas: i) configuração da rede para incluir os nós móveis (veículos) com suporte ad-hoc nativo em IPv6; ii) configuração da rede para incluir o movimento dos veículos; iii) testar a conectividade em IPv6 nos nós móveis; iv) definir e implementar uma estratégia de encaminhamento DTN para os nós móveis, capaz de detectar ausência de conectividade ou novas conexões, e de decidir se um novo vizinho é ou não um bom encaminhador para as mensagens que tem guardadas para enviar; v) implementar, instalar e executar esse novo módulo em todos os veículos (ex: dtn-node); vi) adaptar a aplicação cliente do jogo de modo a que use o suporte DTN (as mensagens passam a ser geridas pelo módulo DTN, que as recebe e envia; mas mantém-se a lógica da comunicação, centrada no nome e/ou endereço do servidor); vii) execução do servidor de jogo; viii) execução da aplicação do jogo no veículo 1 e no veículo 2;

Exemplo meramente ilustrativo: no caso do quiz, a lógica do jogo já implementada no cliente e no servidor mantêm-se inalterada. Não são espectáveis alterações. As mensagens usadas na versão anterior do jogo também podem ser as mesmas, exatamente com a mesma sintaxe e semântica. O que muda é a aplicação cliente e o servidor. A aplicação cliente do jogo em vez de tentar enviar uma mensagem diretamente via socket para o servidor, entrega-a localmente no seu módulo DTN, e é o módulo DTN que decide o que fazer de acordo com a estratégia DTN definida. Pode por exemplo enviá-la para todos os nós que encontrar, de forma epidémica, que algum vai conseguir entregar no destino. A estratégia pode ser mais inteligente e eficiente. Para facilitar a implementação, o servidor de jogo pode ser implementado num nó de fronteira entre a rede infraestruturada e a rede dtn veicular, o que permitirá que a adaptação a fazer do lado do servidor seja semelhante à do lado do cliente.

3.3 NDN (Opcional)

Nesta última fase do projeto, o objetivo é adaptar o jogo desenvolvido na primeira fase, de forma a que funcione numa rede de dados nomeados que é ao mesmo tempo, a rede oportunista que foi implementada na fase anterior.

No contexto deste projeto, propôs-se a implementação de um jogo para múltiplos jogadores. Agora vai ser necessário adaptá-lo para um novo paradigma, centrado nos nomes, e com uma lógica de comunicação totalmente diferente. Não interessa qual o endereço ou o nome do “servidor” e/ou dos “clientes”, mas sim que informação é produzida e que informação é consumida. É preciso pensar em primeiro lugar nos nomes a dar à informação. Dependendo do jogo, a forma de o implementar numa rede de dados nomeados irá mudar de forma mais ou menos significativa. Partindo do princípio que os vários jogadores terão que partilhar entre si as várias jogadas que executam, ou seja, que existe um estado que é preciso partilhar, numa rede de dados nomeados já não se trata de enviar essa informação para os vários jogadores. O jogador ao jogar, transforma-se num produtor de informação, enquanto que os outros jogadores serão os consumidores que demonstram interesse na informação gerada.

Para implementar a rede NDN cada grupo terá que implementar, ainda que de forma simplificada, as estruturas de dados que esta arquitetura prevê, ou seja, cada nó deverá possuir:

- uma *Content Store* (CS), para armazenamento dos pacotes de dados, quer os relativos aos dados produzidos, caso se trate de um produtor, quer os relativos aos pacotes de dados reenviados, já que a CS funciona também como uma cache;
- uma *Pending Information Table* (PIT), para armazenamento dos pacotes de interesse pendentes;
- eventualmente uma *Forwarding Information Base* (FIB);

Assim, os pedidos de interesse são gerados sempre que um utilizador (consumidor) exprime interesse em obter informações acerca do jogo. Esses pedidos deverão ser espalhados pela rede veicular usando *flooding* ou eventualmente consultando a informação de encaminhamento da FIB. Um nó ao receber um pedido de interesse, deverá antes de mais verificar se não é capaz de o satisfazer. Se for, envia de volta pacotes de dados com as informações correspondentes, se não for, deverá reencaminhar o pacote de interesse. Os pacotes de interesse deverão ser inseridos na PIT sempre que forem reencaminhados de forma a ser possível reencaminhar os pacotes de dados através do mesmo caminho que o pacote de interesse efetuou.

Resultado esperado: tal como na etapa anterior, uma demonstração do jogo na rede veicular, com um mínimo de dois jogadores nos veículos, com conectividade intermitente.

Etapas recomendadas: i) redefinir a comunicação para o envio de pacotes de interesse e recepção de pacotes de dados, dando nomes aos dados; ii) adaptar a lógica de comunicação no jogo a este novo paradigma (enviar interesses, receber dados); iii) reimplementar a aplicação do jogo e a aplicação do jogador para suportar este paradigma; iv) implementar, instalar e executar um suporte mínimo NDN em todos os nós da rede, fixos ou móveis (ex: ndn-node); v) execução do servidor de jogo; vi) execução da aplicação do jogo no veículo 1 e no veículo 2;

Exemplo meramente ilustrativo (quiz): no caso do quiz, os nomes podem ser, por exemplo, */quiz*, para os dados do jogo, */quiz/pergunta/<N>* para as perguntas do jogo, e */quiz/resposta/<N>/<jogador>* ... para as respostas dos vários jogadores. Para consumir um novo jogo, a aplicação cliente envia pacotes de interesse com o prefixo */quiz*. O jogo produz dados que fazem match com esse prefixo. O jogo produz perguntas do quiz, em sequência, enquanto a aplicação do jogador consome perguntas do quiz, na mesma sequência. A aplicação do jogador é um produtor de respostas a perguntas, o jogo por sua vez é um consumidor de respostas a perguntas do quiz. Para cada nova pergunta do jogo, a aplicação do jogador gera um pacote de interesses com o prefixo */quiz/pergunta/<N>*, e a aplicação do jogo produz um pacote de dados, com esse nome, a cada nova pergunta. Só produz uma nova, depois de ter consumido uma resposta certa na jogada anterior. E assim sucessivamente.

4. Ambiente de Desenvolvimento e Teste

A implementação e teste desta Rede Adhoc, bem como dos protocolos de encaminhamento e aplicação propostos, deverá ser efetuada na plataforma de emulação CORE (Common Open Research Emulator); esta mesma plataforma CORE deverá utilizada para a implementação dos nós móveis, incluindo, a utilização e programação do seu modelo de mobilidade. Como já foi referido, todos os nós da rede devem estar configurados em IPv6, e deverão utilizar multicast para implementar a descoberta dos vizinhos.

5. Entrega do trabalho

O trabalho deve ser realizado em grupo (de três elementos) sendo a constituição dos grupos da inteira responsabilidade dos alunos. O trabalho deverá ser demonstrado em duas fases (*demonstração intermédia* a 8/4 e *demonstração final* a 20/05). Na demonstração final, além da demo, os alunos deverão elaborar um

relatório escrito que descreva o trabalho efetuado. Este relatório e o respetivo código deverão ser submetidos até ao dia 18/05 na plataforma *elearning.uminho.pt*.

O relatório deve ser escrito em formato de artigo com um máximo de 8 páginas (recomenda-se o uso do formato LNCS - *Lecture Notes in Computer Science*). Deve descrever o essencial do desenho e implementação com a seguinte estrutura recomendada: Introdução; Especificação dos protocolos (primitivas de comunicação, formato das mensagens protocolares (PDU), interações); Implementação (detalhes, parâmetros, bibliotecas de funções, etc); Testes e resultados; Conclusões e trabalho futuro.

Referências

- Benamar, N., Singh, K. D., Benamar, M., El Ouadghiri, D., & Bonnin, J. M. (2014). Routing protocols in vehicular delay tolerant networks: A comprehensive survey. *Computer Communications*, 48, 141-158
- Saxena, D., Raychoudhury, V., & Becker, C. (2017, January). Implementation and performance evaluation of name-based forwarding schemes in V-NDN. In *Proceedings of the 18th International Conference on Distributed Computing and Networking* (pp. 1-4).