Implementação de um Serviço de Notícias numa Rede Adhoc

1. Introdução

Numa rede infra-estruturada, praticamente todos os nós implementam a pilha protocolar completa, desde os routers aos sistemas terminais, no entanto uns e outros desempenham papeis claramente distintos. Os routers, cujas funções principais são da camada de rede (layer3), quase poderiam dispensar o transporte e as aplicações, não fora dar-se o caso de precisarem ser configurados remotamente, precisamente usando protocolos de aplicação como o telnet ou ssh! Os sistemas terminais, que só precisam de enviar e receber os seus próprios pacotes, prescindem dos protocolos de routing da camada 3, usando apenas rotas estáticas e um default router. Pelo contrário, numa rede Adhoc, de formação espontânea, não há nós com papel especial e todos precisam de ajudar a fazer tudo. Mesmo um simples telemóvel, tem de colaborar no envio e receção de pacotes de outros telemóveis, se quiser participar na formação de uma rede funcional.
Neste trabalho pretende-se implementar um protótipo de uma rede de formação espontânea (AdHoc), em que os nós são capazes de descobrir rotas para outros nós com ajuda dos seus vizinhos. O serviço a fornecer é simplesmente um serviço de difusão de notícias muito simples. Assim, deverão ser concebidos e implementados dois protocolos diferentes: um para descobrir rotas (protocolo de encaminhamento) e outro para enviar e receber as mensagens que contêm as notícias (protocolo aplicacional). O re-envio das mensagens geradas pelo serviço de notícias deverá obrigatoriamente ser feito usando as rotas estabelecidas pelo protocolo de encaminhamento concebido.

2. Descrição

2.1 Protocolo de Encaminhamento

Em cada nó coloca-se em execução uma instância de uma aplicação AdHoc, designada por *adhoc_app*. Ao iniciar, a aplicação *adhoc_app* não conhece nada nem ninguém à sua volta. Fica de imediato à escuta na porta 9999 em TCP e UDP. O TCP vai ser usado para "simular" a interface entre a camada de aplicação e a camada de rede e a porta 9999 em UDP será usada para dialogar com os nós vizinhos de acordo com o protocolo de encaminhamento a implementar. A primeira operação é descobrir vizinhos, sem a qual nada de útil se pode fazer. Constrói-se uma mensagem HELLO, e envia-se por multicast para o endereço IPv6 de grupo FF02::1 ("all hosts"), com TTL=1 (só vizinhos directos). Se todos os nós enviarem e escutarem estas mensagens na porta 9999, ficam a conhecer, de imediato, os vizinhos directos (V=1). Se nas mensagens incluírem como informação adicional a lista dos seus vizinhos conhecidos, todos ficam a conhecer integralmente a vizinhança de raio 2 (V=2). Esta tabela de rotas é uma espécie de "olho do peixe" (analogia com a imagem da figura 1) do nosso protocolo e permitirá resolver a maior parte das comunicações sem mais demoras.

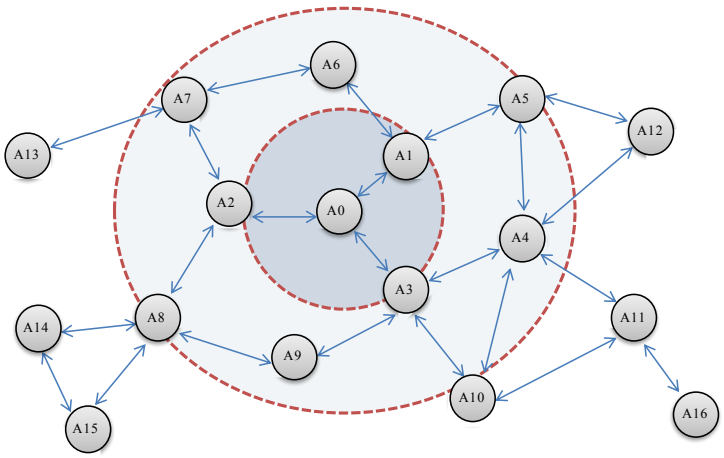


Figura 1

A título de exemplo apresenta-se a tabela de encaminhamento do nó *A0* da figura 1:

Nome	Vizinho	Endereço do vizinho
A1	A1	IPv6(A1)
A2	A2	IPv6(A2)
A3	A3	IPv6(A3)
A4	A3	IPv6(A3)
A5	A1	IPv6(A1)
A6	A1	IPv6(A1)
A7	A2	IPv6(A2)
A8	A2	IPv6(A2)
A9	A3	IPv6(A3)
A10	A3	IPv6(A3)

Mensagens recebidas pelo nó que sejam dirigidas a algum vizinho da vizinhança V=2, podem ser entregues sem dificuldade, direta ou indiretamente. Quanto aos nós fora dessa vizinhança, caso existam e estejam alcançáveis, serão descobertos com pedidos de rota especiais ROUTE REQUEST. Os pedidos devem ser difundidos de todos para todos (*flooding*), até uma vizinhança máxima incluída no pedido de Z=N; Se o pedido atingir um nó que possui uma rota válida, ou o próprio nó de destino, este responde com um ROUTE REPLY. A mensagem de ROUTE REPLY terá de conseguir fazer o percurso inverso à mensagem de ROUTE REQUEST, e as tabelas de encaminhamento nesse percurso deverão ser atualizadas em conformidade. Se ao fim de algum tempo nenhuma resposta for recebida, o nó destino deve ser declarado temporariamente inatingível.

Naturalmente que pedidos muito frequentes de rota para fora da vizinhança V=2 podem originar muito tráfego na rede, eventualmente desnecessário. Recomenda-se que idealizem e proponham melhorias a este processo de modo a torná-lo mais simples e eficiente.

Relativamente aos protocolos a desenvolver incluem-se abaixo algumas sugestões concretas. Os grupos têm liberdade para decidir e justificar toda e qualquer escolha que façam no design do seu serviço.

Protocolo HELLO

Parâmetros de configuração recomendados:

- *hello interval* - período de tempo entre dois HELLO consecutivos do mesmo nó;
- *dead interval* – período de tempo de silêncio de um nó a partir do qual ele é declarado inalcançável

As mensagens de HELLO devem ter um formato bem definido e simples. Nos dados podem incluir a lista de vizinhos de vizinhança V=1 que conhecem (com o cuidado de evitar ou lidar bem com a fragmentação); O endereço IPv6 de destino é FF02::1, porta 9999 (UDP), TTL=1.

Protocolo ROUTE_REQUEST

Parâmetros de configuração sugeridos:

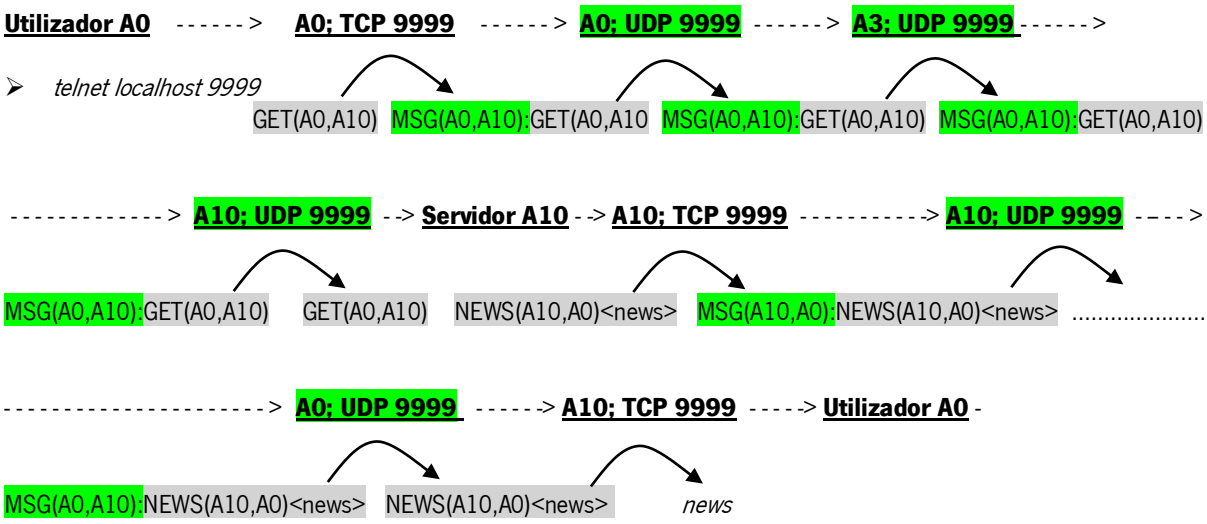
- *timeout* – tempo máximo de espera por uma resposta a um pedido de rota...
- *radius* – raio máximo para procurar por uma rota (em numero de saltos)

As mensagens ROUTE_REQUEST despoletam um processo de descoberta de rota para um determinado destino. O nó de destino, ou um nó com uma rota válida na tabela de vizinhança V=2 pode responder ao pedido com uma mensagem ROUTE_REPLY. Estas mensagens também devem ser enviadas e recebidas na porta 9999 (UDP).

2.2 Protocolo de Aplicacional

Este protocolo é o equivalente da camada de aplicação, sendo em formato de texto, legível para o utilizador. O servidor mantém as notícias (desejavelmente numa base de dados, mas para já pode ser em mesmo em memória), e o cliente recebe como argumento a identificação de um servidor e tenta obter as notícias lá armazenadas. Tanto o cliente como o servidor devem manter uma ligação com a porta TCP 9999 para enviar e receber estas mensagens (esta ligação simula a "ligação entre o nível aplicacional e o nível de rede").

A título de exemplo, suponhamos que o cliente A0 quer fazer o download de notícias a partir do servidor A10. Nesse caso o cliente A0 deverá enviar uma mensagem com o pedido de notícias (por exemplo GET_NEWS_FROM A10) através da porta TCP 9999. A "camada de rede" do *localhost* recebe esta mensagem do utilizador no formato definido, acrescenta-lhes um cabeçalho MSG com o nó de origem (o seu próprio nome!) e nó destino (o nome escrito logo após GET_NEWS_FROM, A10 no exemplo) e entrega-as na porta UDP 9999 em *localhost*. As mensagens são depois reencaminhadas pelos agentes UDP, nó a nó até ao destino. O agente UDP do nó destino retira o cabeçalho MESSAGE e entrega a mensagem na porta TCP 9999 local.



3. 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 e descoberta das rotas fora da vizinhança.

4. 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 na aula de 5/Abril/2018. 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 2/Abril/2018 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 do protocolo (primitivas de comunicação, formato das mensagens protocolares (PDU), interações); Implementação (detalhes, parâmetros, bibliotecas de funções, etc); Testes e resultados; Conclusões e trabalho futuro.