

Implementação de Redes Locais no NS-3 e Análise de Tráfego

1 Topologia do ambiente de rede

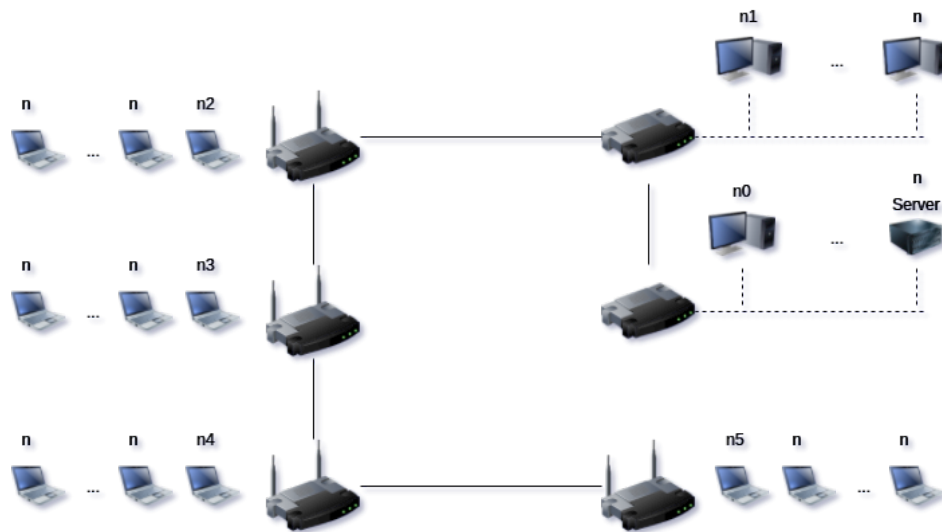


Figura 1: Ilustração da topologia do ambiente de rede, onde os roteadores com antena são as redes sem fio, e os roteadores sem antena são as redes *Ethernet*. As figuras que representam os *notebooks* estão conectadas às suas respectivas redes sem fio, e as figuras que representam os *desktops* estão conectadas às suas respectivas redes *Ethernet*.

A *topologia de rede* é o canal no qual o meio de rede está conectado aos componentes de uma rede. A topologia idealizada e implementada tinha como requisitos: Possuir ao menos 2 redes *Ethernet* 802.3; Possuir ao menos 4 redes sem fio 802.11x; E finalmente o mínimo de 10 clientes em todas as redes a serem implementadas. A topologia implementada se assemelha a uma topologia *linha* como pode ser observado na imagem acima.

Os ambientes das redes *Ethernet* e sem fio implementados são domínios *diferentes de colisão*, ou seja, as redes são diferentes, contando um total de 6 ambientes de redes diferentes, cada uma com 12 clientes e um nó para conexão *peer-to-peer*. Para a realização da simulação, um servidor de aplicação foi implementado no último nó da primeira rede *Ethernet*, sendo este servidor acessado por clientes de todas as outras redes implementadas.

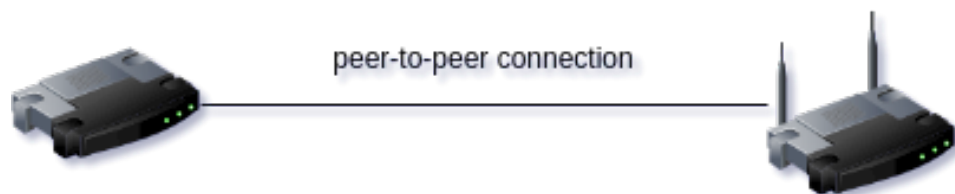


Figura 2: Ilustração de uma conexão *peer-to-peer* utilizada na topologia implementada.

Para implementar a intercomunicação entre estas seis redes, uma conexão *peer-to-peer* (Ilustrada na figura acima) foi concebida para o primeiro nó de cada ambiente de rede, exemplificando, o primeiro nó da primeira rede *Ethernet* foi conectado ao primeiro nó da segunda rede *Ethernet* e assim por diante, como pode ser observado na ilustração acima, que representa a topologia escolhida e implementada. As redes *peer-to-peer* vantagens e desvantagens e ambas serão explicitadas a seguir.

As vantagens das redes *peer-to-peer* são: Facilidade na conexão, pois diferentemente de sistemas que são inteiramente centralizados, não precisa de uma instalação ou administração específica; Conteúdo diversificado, porque conforme vão aumentando as estações interconectadas, as redes *peer-to-peer* também vão crescendo e os recursos que são compartilhados vão se diversificando; Segurança no compartilhamento, porque o tipo de rede *peer-to-peer* está menos sujeita a falhas, erros de compartilhamentos ou ataques intencionais, pois procede de sistemas já distribuídos, que formam um ambiente para trocas de dados ideais para longo prazo; Confiabilidade na rede, pois se tiver problema em algum *peer* o sistema não para totalmente, então, fazendo uso de recursos ou conteúdos existentes os demais podem continuar se mantendo atuantes; E finalmente a disponibilidade no uso dos recursos, se um *peer* não estiver fazendo uso de recurso, este pode deixar disponível os seus recursos para outros *peers* e assim aumentar a capacidade no processamento da rede.

As desvantagens das redes *peer-to-peer* são: Redundância, por causa do tamanho das redes não é comum realizar duas buscas e adquirir o mesmo resultado, mas é possível; Tempo de resposta, que pode variar com o aumento da rede e se houver um longo tempo para o retorno, corre o risco de perder a informação buscada; Perda de conteúdo, porque se um *peer* tem a facilidade de entrar e sair da rede, então, um conteúdo compartilhado pode deixar de existir na rede.

A partir dos pontos discutidos anteriormente a justificativa da escolha desta topologia é dada tanto pela maior simplicidade de implementação da mesma no ambiente do *ns-3* quanto pela proximidade relativa da topologia idealizada com a topologia linha, utilizada em ambientes reais de redes.

2 Descrição dos serviços e clientes em cada rede

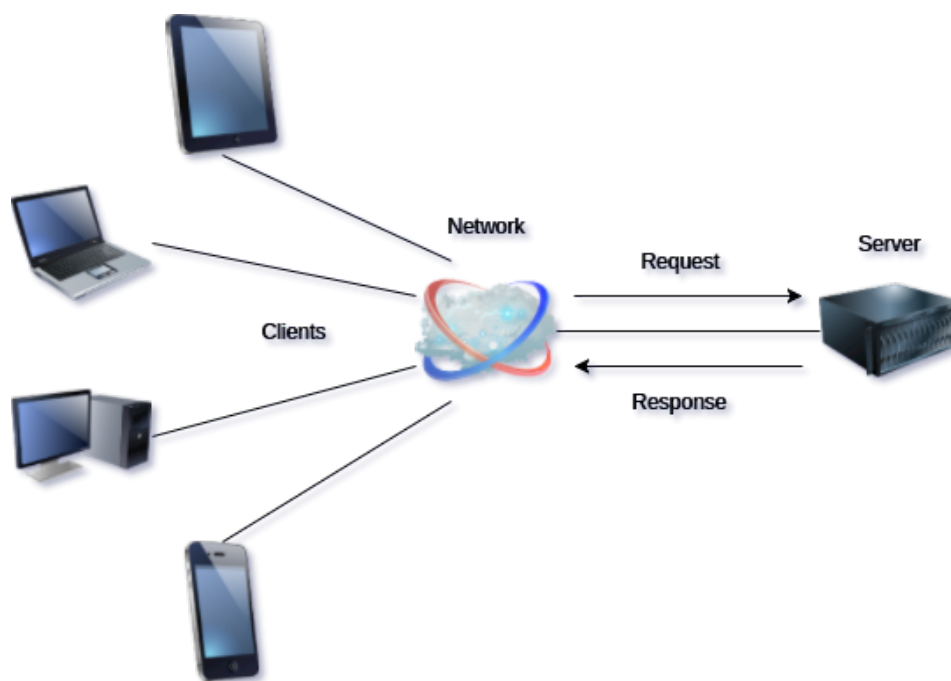


Figura 3: Ilustração de uma conexão *peer-to-peer* utilizada na topologia implementada.

O modelo cliente-servidor é uma estrutura de aplicação que distribui as tarefas e cargas de trabalho entre os fornecedores de um recurso ou serviço, designados como *servidores*, e os requerentes dos serviços, designados como *clientes* (Como

ilustrado na figura acima). Clientes e servidores comunicam-se através de uma rede em dispositivos distintos, mas tanto o cliente quanto o servidor podem residir no mesmo dispositivo. Um servidor é um *host* que está executando um ou mais serviços ou programas que compartilham recursos com os clientes. Um cliente não compartilha qualquer de seus recursos, mas solicita um conteúdo ou função do servidor. Os clientes iniciam sessões de comunicação com os servidores que aguardam requisições de entrada.

Cada rede implementada possui doze nós clientes, que mandam requisições para o servidor no último nó da primeira rede *Ethernet*. O servidor é iniciado com 1 segundo de simulação e para com o tempo de 70 segundos, a implementação do mesmo foi feita por meio da classe *UdpEchoServerHelper* que espera o recebimento de pacotes *UDP* e manda os mesmos pacotes de volta para o remetente original. Os clientes são inicializados com dois segundos de simulação e para com o tempo total da simulação de 70 segundos, estes são um total de $(12 * 6) = 72$ nós utilizaram a classe *UdpEchoClientHelper* que envia um pacote *UDP* para o servidor criado e espera um *eco* do pacote enviado.

A classe *Ipv4GlobalRoutingHelper* foi utilizada para auxiliar a intercomunicação entre os clientes e os servidores. E finalmente a classe *FlowMonitorHelper* gera um arquivo *xml* com as informações de taxas de bytes e de pacotes e também o *throuput* da transação cliente-servidor, também imprimindo estas mesmas informações.

3 Código desenvolvido (clique aqui)

4 Vídeo demonstrando a simulação (clique aqui)

5 Vídeo expondo o principal arquivo de captura gerado pela simulação (clique aqui)

Referências

1. Cliente-servidor - Wikipedia;
2. Vantagens e Desvantagens do modelo P2P - Compartilhamento de arquivos;
3. Topologia de rede - Wikipedia;