**NAT VS PAT:**

**Network Adddress Translation:**

É usado para mapear endereços públicos IP para privados, pode usar a filosofia one-to-one ou a filosofia many-to-one

A screenshot of a cell phone

Description automatically generated

* Principal razão de utilização: diminuir a taxa de esgotamento do espaço de endereçamento.
* Dá segurança uma vez que o processo de tradução é transparente.

**Port Address Translation:**

É usado para mapear múltiplos endereços privados para um único endereço público IP. Usando portos. Usa a filosofia many-to-one.

* O PAT deriva do NAT.

**Tipos:**

* **Static** **NAT:** Relação one-to-one, endereço privado mapeado para endereço publico. Útil quando o host requer um endereço consistente, acessível na internet. Por exemplo: Servidores. Configurado manualmente.
* **Dynamic NAT:** Configurando dinamicamente, o dynamic NAT permite que enderecos privados IP não registados sejam traduzidos em endereços IP públicos registados através de uma pool de endereços IPs Públicos.
* **NAT / PAT Overloading:** Variante de Dynamic NAT mas que mapea múltiplos endereços privados IP para um único IP público endereço usando portas. Se um grupo de portas não estiver disponível ou estiverem ocupados e se existir mais algum endereço IPV4 externo disponível, o PAT vai tentar alocar nesse endereço.

More: <https://techdifferences.com/difference-between-nat-and-pat.html>

**O NAT é controverso:**

Routers devem apenas processar na camada 3 (IPs)... O NAT faz isso, mas o PAT pode ser visto como um modo de operar na camada 4 (de transporte), porque envolve portos....

**Port (ou porto ou porta):**

Por exemplo, faz de conta que o IP address é a rua, o port vai ser, neste caso, uma casa ou apartamento. O número da porta da casa. A casa neste caso, é a aplicação (no caso dos protocolos UDP TCP) ou computador (no caso dos protocolos NAT PAT).

**End-to-end principle:**

Core/centro da internet simples e reliable. Não fazer mais do que precisamos / do que é necessário. A inovação fica por conta dos dispositivos terminais. Por exemplo, quero implementar uma nova maneira de transmitir dados na internet. Eu não preciso de fazer isso no core da internet, basta ter os dois dispositivos no end node a rodar o mesmo protocolo.

**O que o E2E promove é também a transparência de enderecos, coisa que o NAT não promove assim tanto.**

<https://devopedia.org/end-to-end-principle>

**NAT Traversal Problem:**

O NAT possui um problema: deixa a rede privada interna inadequada para sistemas de servidores de hospedagem, pois o dispositivo NAT não possui um método automático de determinar o host interno para qual os pacotes recebidos são destinados.

Solicitações de entrada (incoming requests) podem não ser facilmente ou correctamente correspondidos com o host interno. Além disso, podem possuir os IPs e portos no meio da application data, requerendo que se faça uma inspeção profunda aos pacotes para uma substituição dos IPs e portos (NAT) .

Isto não é um problema para acessamento geral de web e de email. Mas nas aplicações peer-to-peer file sharing e nas VoIP e nos jogos de consolas, requere que os clientes também sejam servidores. As ligações P2P requerem que o Peer A seja capaz de iniciar uma ligação TCP com qualquer outro participante Peer B. O problema é que o Peer B pode estar por detrás de uma NAT, o que não pode agir como server e aceitar o TCP connection.

**As NAT Traversal** são técnicas que permitem estabelecer conexões IPs através de Gateways que implementam o NAT. É um requisito para muitas das aplicações como peer-to-peer file sharing e Voice over IP.

Um exemplo de NAT Traversal usado para conexões P2P é usar um terceiro Peer C, que não esta por detrás de uma rede NAT e que tem ligação TCP com o Peer B, assim o Peer A e o Peer B já se podem comunicar. É um hack chamado de connection reversal. Se ambos os peers tiverem em NATs, é mais difícil, mas pode ser resolvido como no caso do Skype.

**Solução 1: Estaticamente configurar NAT para encaminhar o pedido de conexão para um determinado porto do servidor:**

Exemplo: (138.76.29.7, port 2500) always forwarded to 10.0.0.1 port 25000

**solução 2: Universal Plug and Play (UPnP):**

Protocolo que permite que o host descubra e configure um NAT por perto. UPnP requer que o host e o NAT sejam UPnP compatíveis.

Dito isto, uma aplicação a correr no host pode pedir um mapeamento do NAT entre o seu endereço e porto privados e um endereço e porto públicos para um qualquer pedido de número publico do porto. Se o NAT aceitar o pedido e criar o mapping, então os nós lá de fora (terminais noutras redes, nós de ISP, etc.....) podem iniciar conexões TCP com o endereço e porto públicos. O UPnP deixa a aplicação saber o valor desse endereço e porto, então ela (a aplicação) pode usar isso no mundo lá de fora.

Exemplo: **automate** static NAT port map configuration

A screenshot of a cell phone

Description automatically generated

Um BitTorrent tracker (ou apenas "tracker") é um servidor que auxilia na comunicação entre dois computadores que utilizam o protocolo P2P BitTorrent. Para que se proceda a uma partilha de arquivos por BitTorrent é necessário que dois PCs se comuniquem com o servidor. De qualquer forma, o mesmo servidor, não aloja conteúdos e são os próprios peers que os partilham entre si as partes dos arquivos. Os clientes que estão a proceder downloads BitTorrent, comunicam esse Servidor de torrents, periodicamente, para dar informação sobre novos peers, como também, de estatísticas usualmente relacionadas com os arquivos compartilhados. No entanto, logo após o início da partilha, a comunicação entre os dois PCs continuará sem a interligação com o "tracker". Assim sendo, um tracker não é mais que um meio de contacto entre duas outras entidades - os peers.

Relaying (usado em Skype):