

Sobre - Arquitetura do componente ATER

Visão Geral

O ATER foi concebido para realizar a comutação de tráfego entre rede de produção (baseada no modelo melhor esforço) e rede de circuitos dinâmicos (baseada em modelos que oferecem qualidade de serviço). Com o ATER, o usuário continuará utilizando a rede de produção normalmente, porém terá a opção de enviar fluxos de aplicações via rede de circuitos dinâmicos. Para tal, o usuário identificará, através de regras de filtragens, os fluxos que deseja enviar via rede de circuitos dinâmicos. O ATER utiliza regras de filtragem para detectar o tráfego do usuário e realizar a injeção desse tráfego na rede de circuitos dinâmicos.

A Figura 1 apresenta a integração entre o ATER e a rede híbrida da RNP, onde estão disponíveis a rede de produção (rede IPÊ) e a rede de circuito dinâmicos (rede CIPÓ). O ATER possui dois componentes principais, CORE e RACE. O RACE é composto por um comutador de pacotes baseado em tecnologia OpenFlow, o qual está conectado às três redes: rede do Cliente, rede IPÊ e rede CIPÓ. Cada RACE possui um controlador dedicado. O CORE é o núcleo de inteligência do serviço e coordena os RACEs aos pares para garantir a comutação dos fluxos de aplicações do usuário.

O usuário insere regras de filtragem através da interface WEB do CORE. O CORE identifica os RACEs que fazem parte da regra de filtragem e os informa que novas regras de filtragem devem ser aplicadas. Os RACEs aplicam as regras de filtragem e passam a identificar o tráfego do usuário. Ao identificar o tráfego do usuário os RACEs informam ao CORE que houve casamento de regra com tráfego e o CORE requisita a criação de um circuito ao OSCARS (componente disponível na rede CIPÓ). Após a criação do circuito com sucesso o CORE informa regras de encaminhamento aos RACEs. Após os RACEs aplicarem as regras de encaminhamento eles estarão aptos a injetar o tráfego do usuário na rede CIPÓ.

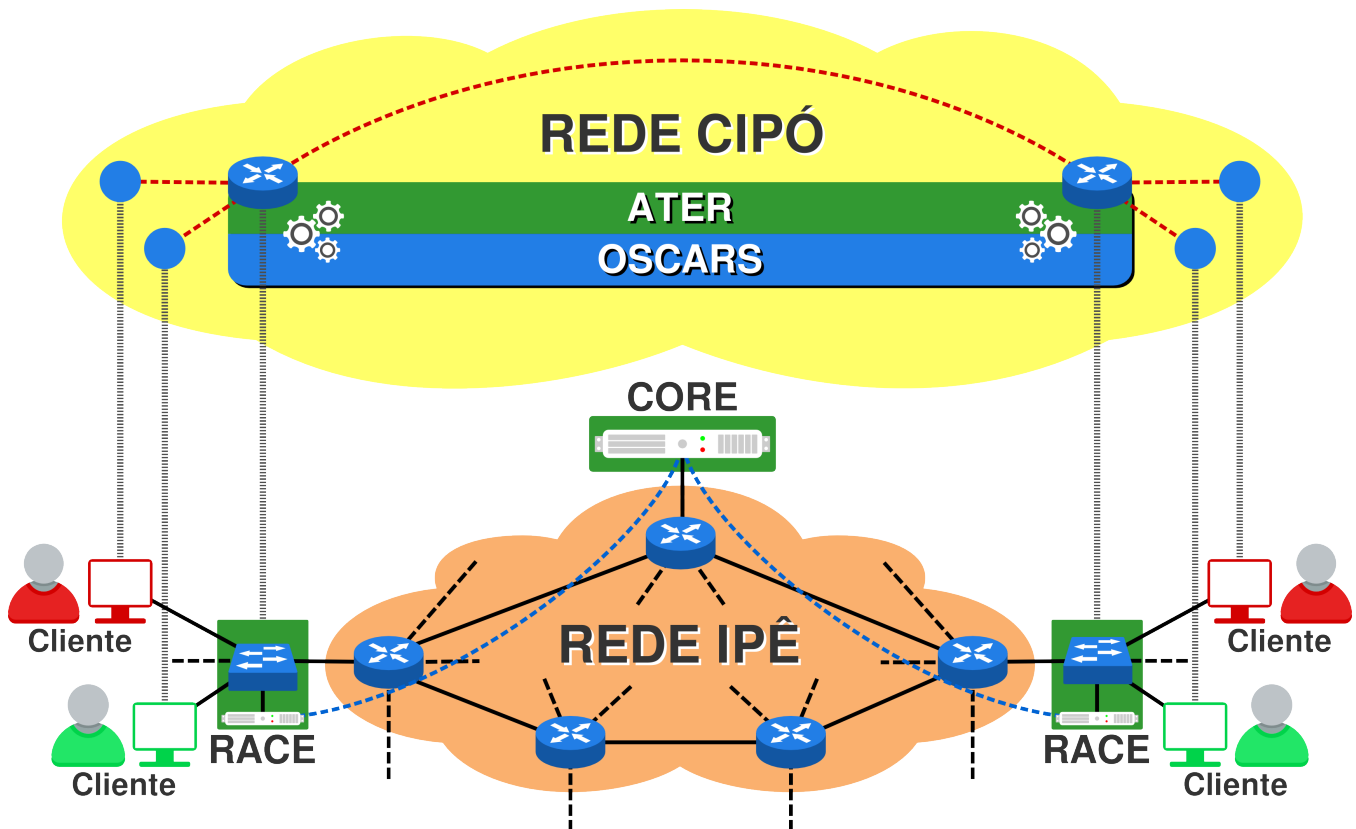


Figura 1: Integração entre ATER e rede híbrida da RNP

Arquitetura

A Figura 2 apresenta os módulos de software dos componentes do ATER. Segue abaixo um resumo de cada módulo:

CORE Web: aplicação Web através da qual o usuário pode criar suas regras de acordo com o perfil de seu tráfego, visualizar suas regras criadas e consultar estatísticas sobre suas regras. Os administradores do ATER também utilizam essa aplicação para gerenciar usuários e regras, monitorar estatísticas de tráfego, verificar o estado dos RACEs e configurar o sistema.

CORE Main: é o núcleo do sistema de gerenciamento de regras e circuitos, o qual é composto por dois principais componentes: o **Circuit Manager**, que é o responsável por coordenar os RACEs no estabelecimento da conectividade entre as extremidades do circuito; e o **RACE Checker** que implementa no lado do CORE, um protocolo do tipo *heartbeat* com os RACEs.

CORE OSCARS Driver (COD): módulo de software responsável pela integração com o OSCARS. O COD funciona como um *driver* para o OSCARS, criando uma abstração mais simples para o restante do sistema e oferecendo um ponto único de acesso à Rede CIPÓ.

CORE Comm: módulo de software instanciado no CORE para implementar um canal seguro, baseado na *libssh2*, para receber requisições dos RACEs.

RACE Comm: módulo de software instanciado no RACE, para implementar um canal seguro, baseado na libssh2, para receber requisições do CORE.

RACE Daemon: módulo de software responsável pela lógica de filtragem e encaminhamento transparente dos fluxos do usuário. O RACE_Daemon é o controlador OpenFlow e foi criado como uma extensão acoplada ao POX.

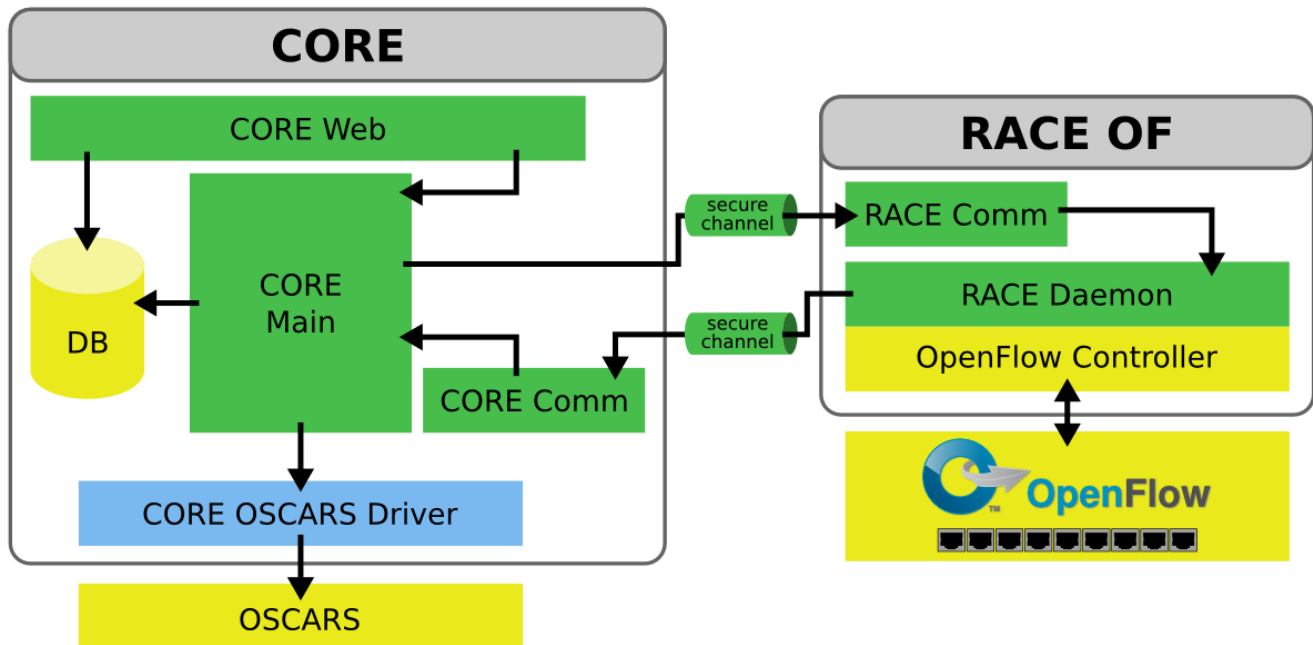


Figura 2: Arquitetura de software

Perfis de Usuário

Os usuários do ATER podem ser classificados em três tipos: **cliente**, **gerente local** e **administrador**. A seguir são descritas as atividades que podem ser realizadas por cada usuário.

- **Cliente:**
 - Solicitação de cadastro
 - Criação de regras
 - Remoção de regras
- **Gerente local:**
 - Todos os recursos do cliente
 - Aprovação de contas locais
 - Aprovação de regras locais
 - Criação/remoção de regras efetivas e de monitoramento
 - Visualização de estatísticas de regras efetivas e de monitoramento
- **Administrador:**
 - Todos os recursos do cliente
 - Todos os recursos do gerente
 - Aprovação de contas de qualquer domínio
 - Aprovação de regras de qualquer domínio
 - Mudar o nível de um usuário para cliente, gerente local ou administrador
 - Realizar configurações do serviço

As regras para identificação do perfil de tráfego podem ser definidas pelos clientes, inicialmente usuários de instituições que tenham necessidade de realizar transporte confiável de grandes fluxos de dados, como também por administradores pertencentes à equipe de operação da RNP. A Figura 3 descreve as principais atividades dos usuários com perfil cliente e gerente local.

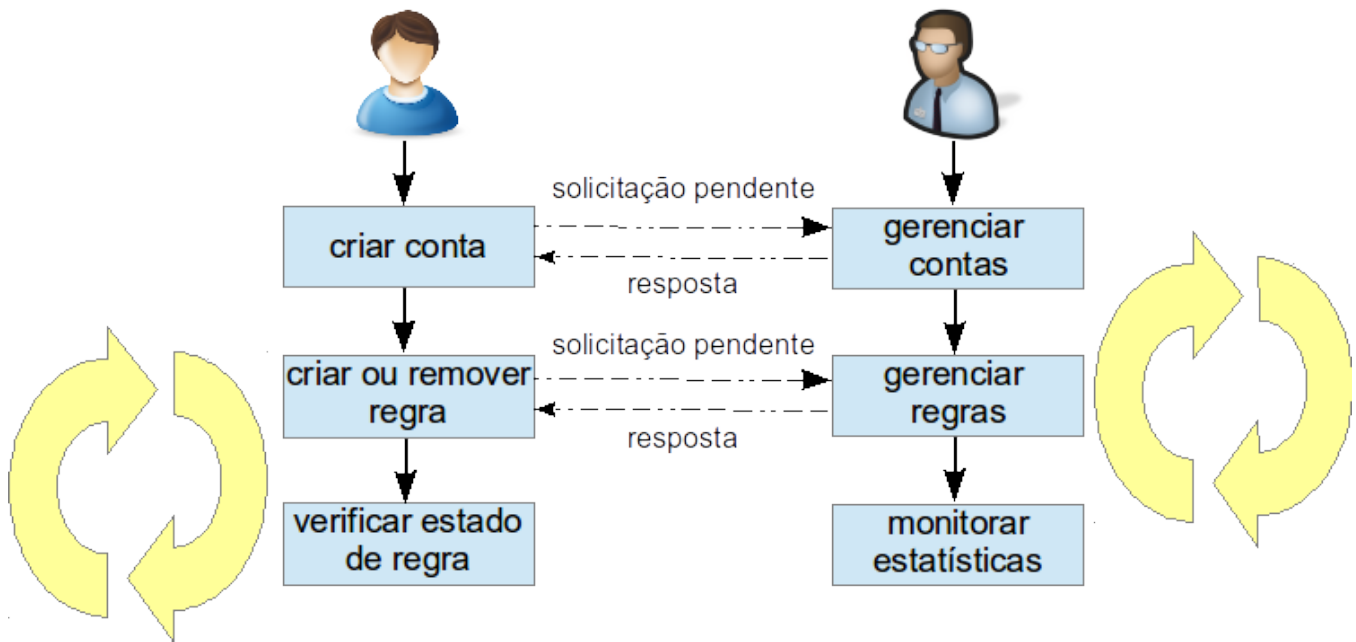


Figura 3: Principais atividades do cliente e gerente local

Tipos de Regra

No ATER, existem dois tipos de regras:

- **Efetiva:** permite solicitar um circuito dinâmico quando um tráfego que casa com ela é identificado.
- **Monitoramento:** não permite solicitar circuito quando um tráfego que casa com ela é identificado. Entretanto, todo tráfego que casa com a regra é contabilizado.

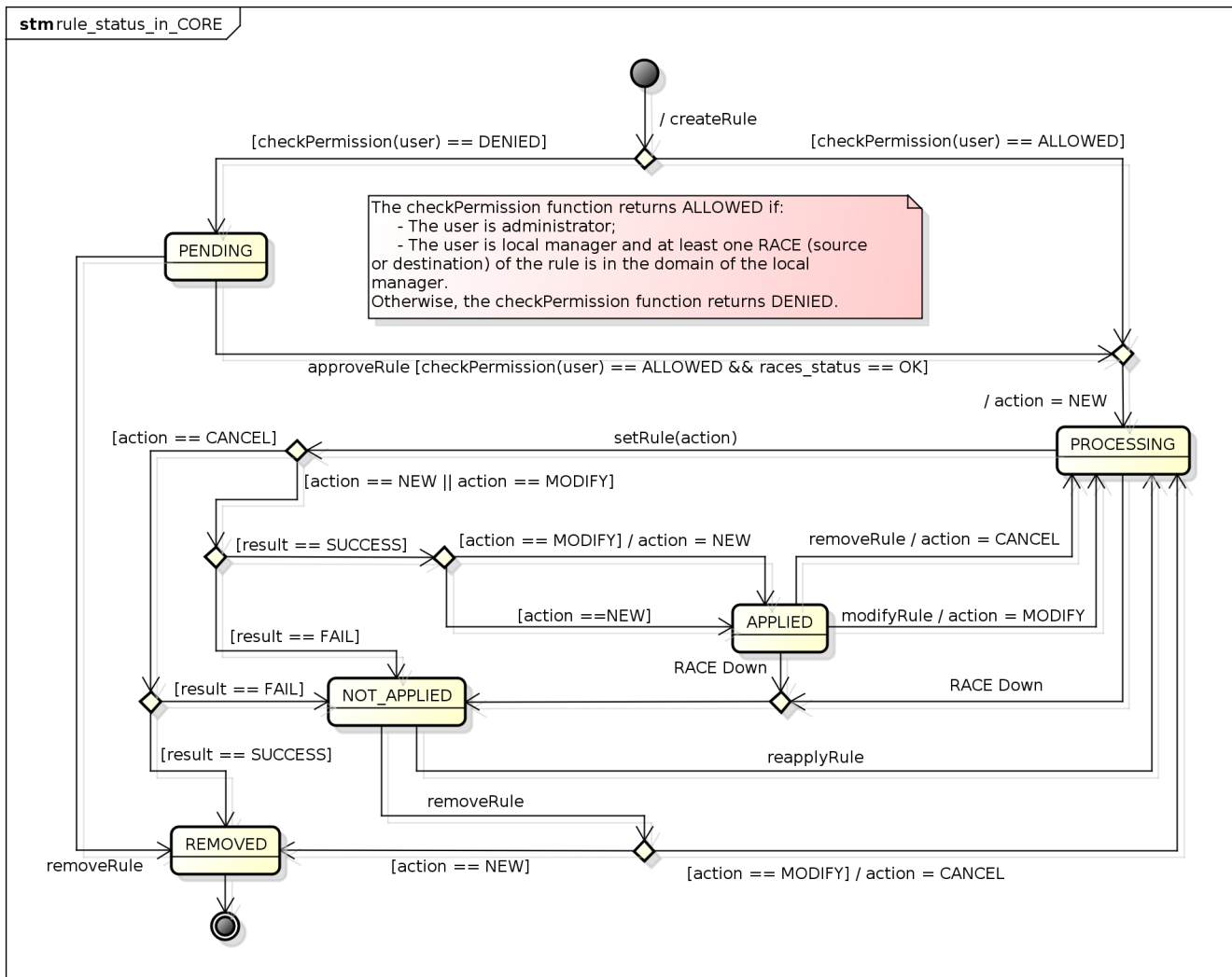
Independente do tipo da regra, todo o tráfego que casa com ela é contabilizado.

Além de criar e remover uma regra, ainda é possível modificá-la. Entretanto, atualmente o serviço só permite a modificação do tipo da regra, de monitoramento para efetiva.

Durante seu ciclo de vida, uma regra pode passar por diferentes estados, dependendo do seu tipo e do componente onde a regra está instanciada. Os componentes que trabalham com as regras atualmente são o CORE e o RACE.

Estado da regra no CORE

A Figura 4 apresenta a máquina de estado da regra mantida internamente no CORE.



powered by Astah

Figura 4: Máquina de estado de regra no CORE

Pode-se dividir as transições da regra no CORE entre os processos de criação, aplicação e remoção da regra. O processo de criação da regra contém alguns detalhes que dependem do tipo da regra e do usuário que está criando a regra.

- **Regra Efetiva**
 - **Usuário Administrador ou Gerente Local:** quando uma regra efetiva é criada por um Gerente Local (pelo menos um dos RACEs da regra deve estar sobre o domínio dele), ou por um Administrador, ela é considerada aprovada automaticamente.
 - **Cliente:** quando uma regra efetiva é criada por um usuário que não tem permissões de aprovação direta, como um cliente, a regra automaticamente assume o estado "pendente" (PENDING), indicando que ela deverá aguardar a aprovação do Gerente Local, dependendo da configuração da regra, ou de um Administrador.
- **Regra de Monitoramento**
 - Quando uma regra de monitoramento é criada, independente do tipo do usuário que a criou, ela é considerada aprovada automaticamente.

Quando uma regra é considerada aprovada, imediatamente ela assume o estado "processando" (PROCESSING) e o processo para aplicá-la nos RACEs é iniciado. O processo de aplicação tem dois resultados possíveis:

- **SUCESSO:** se o processo de aplicação da regra obtiver sucesso, ou seja, a regra for aplicada com sucesso em ambos os RACEs, a regra assume o estado "aplicada" (APPLIED).
- **FALHA:** se o processo de aplicação da regra falha, ou seja, a regra não pode ser aplicada em um ou mais RACEs, a regra assume o estado "não-aplicada" (NOT_APPLIED).

Uma regra pode ser removida quando ela está em qualquer estado, exceto "processando". Nesse processo, algumas verificações acontecem sobre a regra para determinar qual o procedimento correto para sua exclusão.

- **Exclusão de regra que nunca foi aplicada:** quando o usuário solicita a exclusão de uma regra que foi criada e nunca foi aplicada com sucesso nos RACEs, seja por não aprovação ou por algum problema durante a aplicação nos RACEs, a regra é excluída diretamente e o ciclo de vida dela termina, sem que os RACEs precisem ser contatados.
- **Exclusão de regras que já foram aplicadas:** quando o usuário solicita a exclusão de uma regra que foi criada e já foi aplicada pelo menos uma vez nos RACEs, a regra assume o estado "processando" e os RACEs são contatados para que também excluam a regra. Caso a exclusão obtenha sucesso, a regra é excluída do CORE e o ciclo de vida dela termina. Caso a exclusão não obtenha sucesso, o próprio CORE vai, automaticamente, tentar excluir a regra em outro momento. Neste último caso, o usuário não pode mais tentar excluir a regra.

Quando o usuário solicita a modificação uma regra, a regra assume o estado "processando" e então a modificação é enviada para os RACEs. Caso essa modificação aconteça com sucesso nos RACEs, a regra assume o estado "aplicada" e ação "NEW". Caso contrário, a regra assume o estado "não-aplicada". Entretanto, uma regra só pode ser modificada se ela já tiver sido aplicada, ou seja, quando ela está no estado "aplicada".

Periodicamente, um componente do serviço tenta reaplicar regras cujas aplicações anteriores foram mal sucedidas. As tentativas são repetidas até que a regra seja aplicada com sucesso. Essa reaplicação acontece com base na ação que foi mal sucedida para a regra, ou seja, a reaplicação pode acontecer para uma regra não aplicada, não modificada ou não removida nos RACEs.

Estado da regra nos RACEs

A Figura 5 apresenta a máquina de estado da regra mantida internamente nos RACEs.

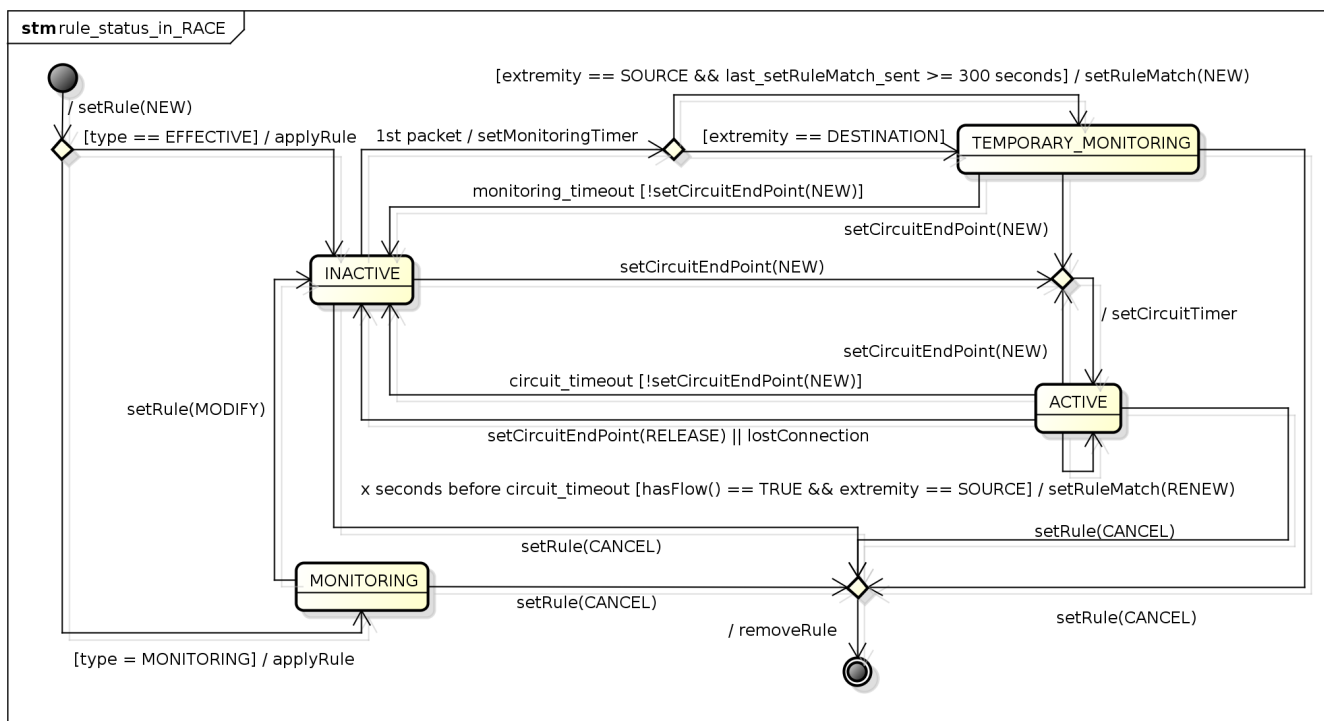


Figura 5: Máquina de estado de regra no CORE

No RACE as transições de estado da regra estão condicionadas ao tipo dela.

- **Regra Efetiva:**
 - Quando o RACE recebe uma regra efetiva, ela assume o estado "inativa" (INACTIVE), que é o estado inicial da regra, de onde ela pode mudar de estado com base nos eventos que acontecem.
 - Quando a regra está no estado "inativa", dois eventos podem ocasionar a mudança de estado da regra:
 - **Tráfego que casa com a regra é identificado:** neste caso, quando o RACE identifica algum tráfego que casa com as configurações da regra, ela assume o estado "monitoramento-temporário" (TEMPORARY_MONITORING) e, se o RACE for o de origem da regra, uma mensagem `setRuleMatch(NEW)` é enviada para o CORE. Além disso, um temporizador é disparado para que o RACE possa voltar a regra para o estado "inativa", caso uma mensagem `setCircuitEndPoint(NEW)` não seja recebida em um intervalo de tempo pré-definido.
 - **Uma mensagem `setCircuitEndPoint(NEW)` é recebida:** Quando o RACE recebe uma mensagem `setCircuitEndPoint(NEW)` com as informações de um circuito para uma determinada regra, essa regra assume o estado "ativa" (ACTIVE).
 - Quando a regra está no estado "monitoramento-temporário" e o RACE recebe uma mensagem `setCircuitEndPoint(NEW)` com as informações de um circuito para uma determinada regra, essa regra assume o estado "ativa".
 - Quando a regra assume o estado "ativa", um temporizador é disparado para que o RACE possa controlar as seguintes operações:
 - **Solicitação de renovação de circuito:** caso o tempo para expiração do circuito recebido esteja próximo do fim e ainda haja tráfego casando com a regra, o RACE solicitará a renovação do circuito para o CORE, através do envio de uma mensagem `setRuleMatch(RENEW)`.
 - **Voltar a regra para o estado "inativa":** caso o RACE não receba uma mensagem `setCircuitEndPoint(NEW)`, mesmo depois de enviar uma mensagem `setRuleMatch(RENEW)` para o CORE, a regra então assume o estado "inativa" novamente.
 - Caso o RACE receba uma mensagem `setCircuitEndPoint(NEW)`, um novo temporizador é disparado e a regra permanece no estado "ativa".
 - Além disso, a regra pode sair do estado "ativa" para "inativa" caso o RACE receba uma mensagem `setCircuitEndPoint(RELEASE)`, que significa que o RACE deve deixar de transmitir dados pelo circuito atual, e também caso o RACE identifique que perdeu a conexão com o CORE.
- **Regra de Monitoramento:**
 - Quando o RACE recebe uma regra de monitoramento, ela assume o estado "monitoramento" (MONITORING). Nesse estado, todas as estatísticas são colhidas, mas caso algum tráfego case com a regra, nenhuma mudança de estado acontece.
 - Quando o RACE recebe uma solicitação de modificação da regra, a regra assume o estado "inativa" (INACTIVE). A partir desse ponto, a regra pode assumir as mesmas transições descritas no item acima, para regra efetiva.

Independentemente do estado que uma regra se encontra, caso o RACE receba uma solicitação de exclusão da regra, a regra é removida do RACE e o ciclo de vida dela termina.

Conexão de Experimentação

A Figura 6 apresenta o modelo de conexão que é utilizado na fase experimental pelo ATER nos PoPs.

A infraestrutura do ATER nessas localidades é composta de um Switch OpenFlow Datacom DM 4001, de uma VM para o Controlador OpenFlow e de uma VM cliente de testes. O RACE será executado na VM Controlador OpenFlow que possui dois IPs: um IP público para acesso remoto (gerência) e um IP privado para envio de comandos ao Switch OpenFlow. A VM cliente de testes possui dois IPs: um IP público para acesso remoto (gerência) e um IP público para testes isolados.

Uma conexão nível 2 é realizada utilizando a VLAN Cliente_ATER desde o equipamento de testes da instituição do usuário até o Switch OpenFlow. Quando o tráfego do usuário chega no Switch OpenFlow a classificação do tráfego é realizada. Se houver regra de encaminhamento o tráfego do usuário será direcionado para a rede CIPO, caso contrário o tráfego será direcionado para a rede IPÊ.

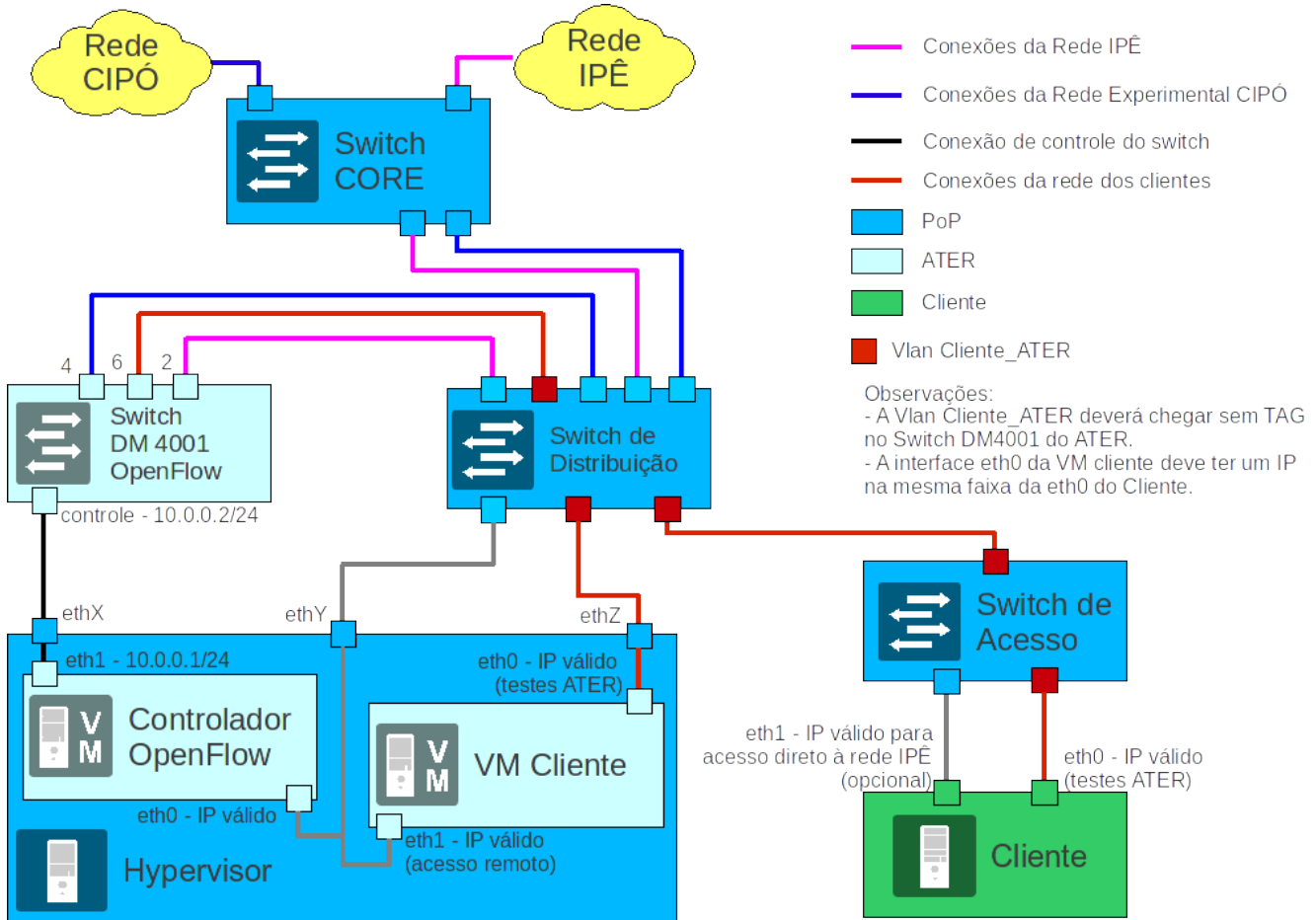


Figura 6: Modelo para implantação dos RACEs nos PoPs da RNP