



## **GT-ATER: Aceleração do Transporte de Dados com o Emprego de Redes de Circuitos Dinâmicos**

RT3 - Avaliação dos resultados do protótipo

Sand Luz Corrêa  
Kleber Vieira Cardoso

**03/08/2013**

## 1. Apresentação do protótipo desenvolvido

Nesta seção, apresentamos o protótipo desenvolvido como resultado do trabalho realizado pelo GT-ATER entre novembro de 2012 e julho de 2013. O protótipo implementado seguiu a arquitetura proposta e detalhada no relatório RT2, a qual é novamente apresentada na Figura 1.

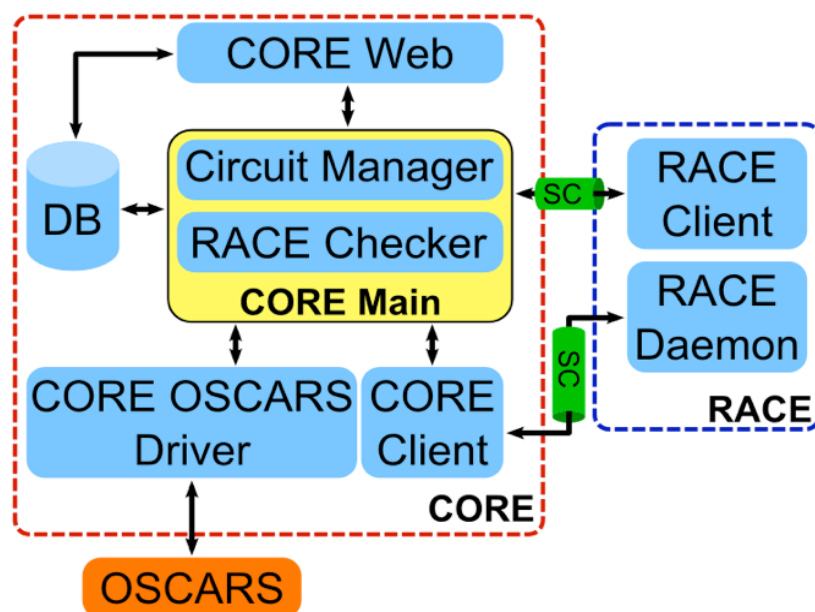


Figura1: Arquitetura do serviço ATER

É importante destacar, nesta figura, que o ATER segue uma arquitetura cliente-servidor. O papel do servidor é desempenhado pelo CORE (*Circuit Operation and Rule Establishment*), que concentra a base de dados e o portal do serviço. Os RACEs (*Rule Applier and Circuit Endpoint*), por outro lado, agem como equipamentos clientes, instalando regras nas extremidades do circuito e analisando o perfil do tráfego do usuário para desviá-lo para o circuito dinâmico, caso haja necessidade. Toda interação do usuário com o serviço ocorre através de um portal Web, implementado pelo componente CORE\_Web, no CORE. Através do portal, usuários e administradores do serviço podem instalar, monitorar e remover regras nos RACEs. Estas regras encaminharão automaticamente um determinado perfil de tráfego através de um circuito dinâmico. Nas seções a seguir, apresentaremos as principais funcionalidades deste portal.

### 1.1 Portal do Serviço

O portal Web é o ponto através do qual os usuários têm acesso aos serviços providos pelo ATER. O acesso aos serviços, no entanto, depende do perfil do usuário. No ATER, existem três perfis possíveis, como descrito a seguir.

1. Cliente: representa o usuário de uma instituição que tem necessidade de enviar um grande volume de dados, de maneira confiável, por um circuito dinâmico. O cliente pode executar as seguintes atividades no portal:

- Criar uma conta;
  - Solicitar a criação de regras para encaminhamento de tráfego;
  - Remover suas próprias regras;
  - Visualizar estatísticas de circuitos gerados por suas próprias regras.
2. Gerente Local: representa o usuário que tem permissão para gerenciar a criação de regras de encaminhamento em uma instituição ou domínio. Geralmente, o gerente local gerencia os clientes de sua instituição, podendo executar as mesmas atividades desses clientes, além das atividades seguintes:
- Aprovar contas locais;
  - Aprovar regras locais;
  - Visualizar estatísticas de circuitos que tenham a instituição ou domínio local como uma das extremidades.
3. Administrador: representa o usuário que administra o serviço ATER, podendo executar as seguintes atividades:
- Aprovar contas independentemente de instituição ou domínio;
  - Aprovar regras independentemente de instituição ou domínio;
  - Criar ou remover qualquer regra;
  - Visualizar estatísticas de qualquer circuito, independentemente de extremidade;
  - Mudar o nível de um usuário para gerente local;
  - Configurar parâmetros do serviço.

A seguir, apresentaremos a visão do portal sob o ponto de vista de cada perfil de usuário.

## 1.2 Portal na Visão do Cliente

A Figura 2 mostra a página inicial do portal do serviço ATER. Para ter acesso aos serviços oferecidos pelo portal, o cliente precisa possuir uma conta. Para criar uma conta, o cliente deve clicar no botão **Solicitar Cadastro**. Dessa forma, ele será direcionado para a tela mostrada na Figura 3. Caso o cliente já possua uma conta no portal, ele deve acessar a página inicial e digitar seu *login* e senha, clicando, em seguida, no botão **Entrar**.

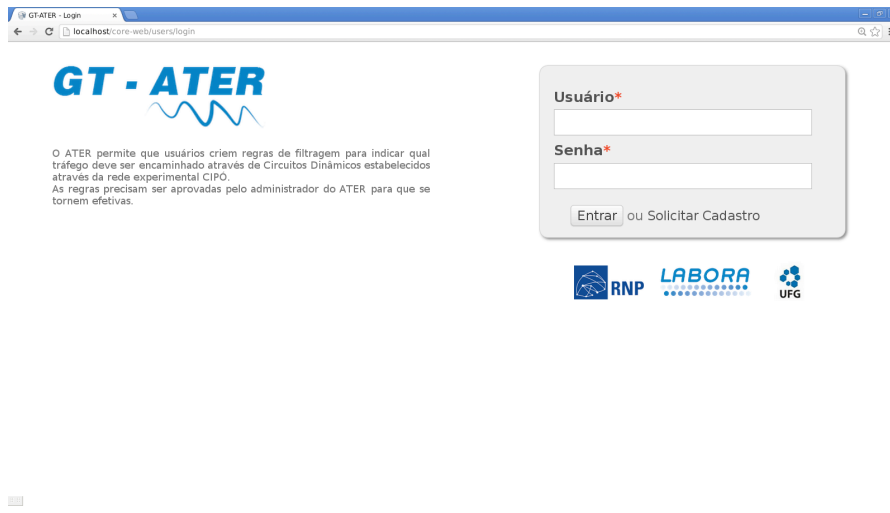


Figura 2: Página inicial do portal Web

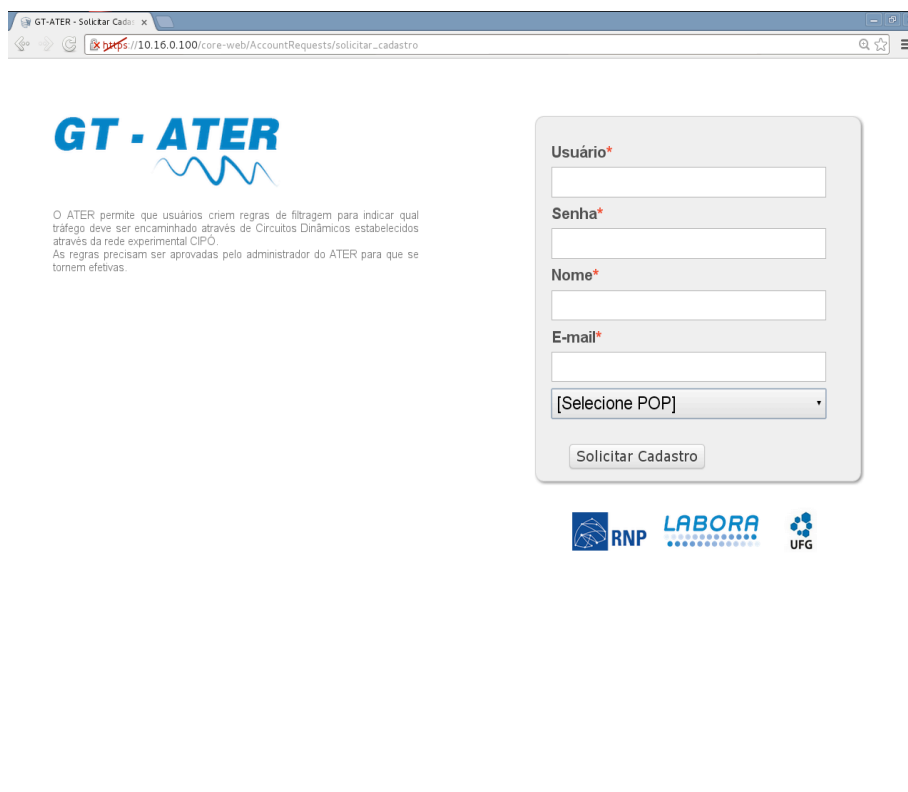


Figura 3: Solicitação de cadastro

Uma vez autenticado, o cliente será direcionado para a página principal do portal (*Dashboard*), mostrada na Figura 4. Nessa página, usuários com o perfil cliente podem solicitar a criação de regras ou visualizar suas regras já criadas ou pendentes de aprovação.

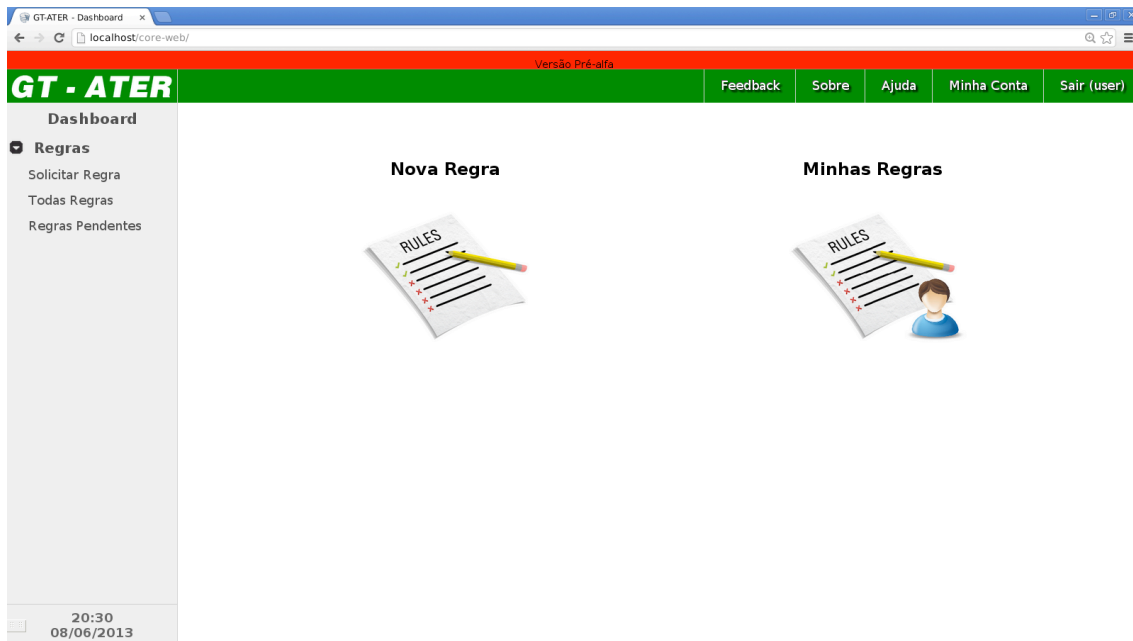


Figura 4: *Dashboard* na visão do cliente

Para solicitar a criação de uma regra, o cliente deve selecionar a opção **Nova Regra**. Neste caso, ele será direcionado para a tela mostrada na Figura 5.

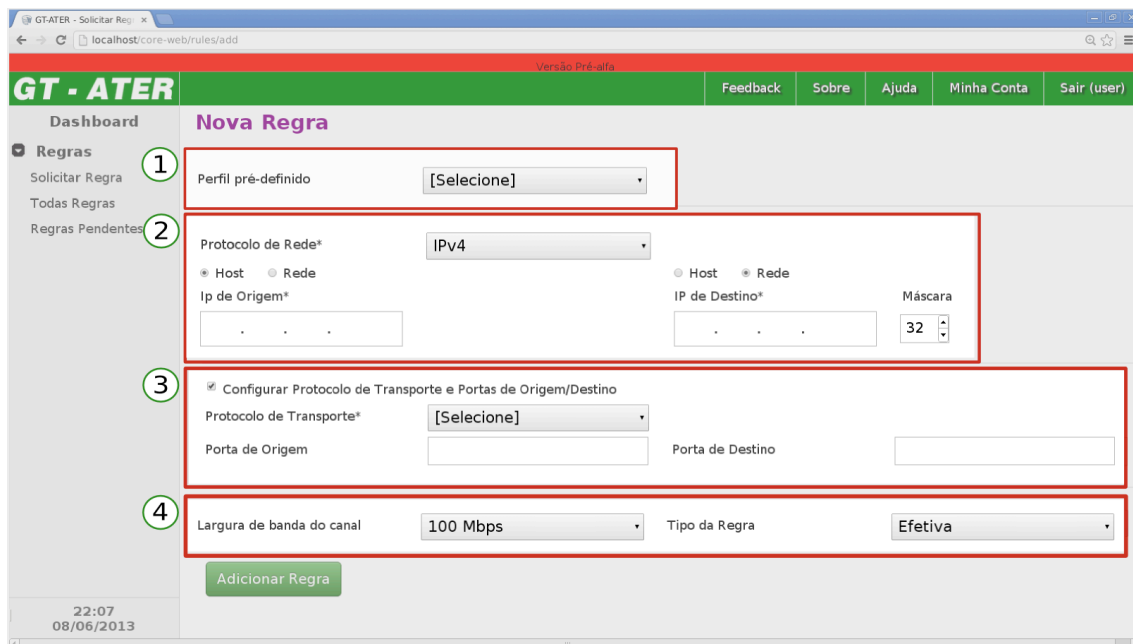


Figura 5: Solicitação de nova regra

Na página de solicitação de nova regra, o cliente deve informar os dados que caracterizam o perfil do tráfego que deve seguir por um circuito dinâmico, informando: protocolo de rede, IP de origem e destino, protocolo de transporte e porta de origem e destino. Também é necessário informar a largura de banda do canal e o tipo da regra que será criada. No ATER, regras são classificadas em dois tipos: efetiva – regra que gera um circuito dinâmico quando o

tráfego casar com a regra – e de monitoramento – regra que apenas contabiliza o tráfego que casa com a regra sem, no entanto, gerar um circuito dinâmico.

Como alternativa ao preenchimento dos campos que caracterizam o protocolo de transporte usado na regra, é possível escolher um perfil de tráfego configurado previamente, como mostrado no campo 1 da Figura 5. Atualmente, existem dois perfis: Web e FTP. O perfil Web define o protocolo de transporte como TCP e a porta destino como 80. O perfil FTP usa o protocolo TCP e a porta 21 como destino.

O mapeamento entre IPs - de origem e destino - e RACEs - de origem e destino - é feita automaticamente através dos blocos de IPs das instituições. É importante mencionar que as regras criadas pelos clientes precisam ser aprovadas pelo gerente local ou administrador do serviço. Apenas as regras aprovadas são instaladas nos RACEs.

Por fim, o cliente pode visualizar suas regras aprovadas ou pendentes de aprovação. Para isto, o cliente deve clicar na opção **Minhas Regras**, no *Dashboard*. Ele, então será direcionado para a tela apresentada na Figura 6.

ID	IP de Origem	Porta de Origem	IP de Destino	Porta de Destino	Protocolo de Transporte	Status da Regra	Tipo da Regra	Ações
141	10.1.0.1	-1	10.2.0.1	80	ICMP	Pendente	Efetiva	 
140	10.1.0.1	-1	10.2.0.1	80	UDP	Pendente	Efetiva	 

Página 1 de 1, 2 Resultados encontrados, mostrando de 1 até 2

< Anterior   Próximo >

22:49  
09/06/2013

Figura 6: Visualização de regras criadas pelo cliente

### 1.3 Portal na Visão do Gerente Local

Uma vez autenticado, o gerente local tem acesso ao *Dashboard* do serviço, mostrado na Figura 7. Como ocorre no *Dashboard* do cliente, o gerente local pode criar novas regras, através da opção **Novas Regras**, e visualizar suas próprias regras, acessando a opção **Minhas Regras**. Entretanto, diferentemente do *Dashboard* do cliente, a página principal do gerente local possui também opções para gerência de usuários e gerência de requisições.

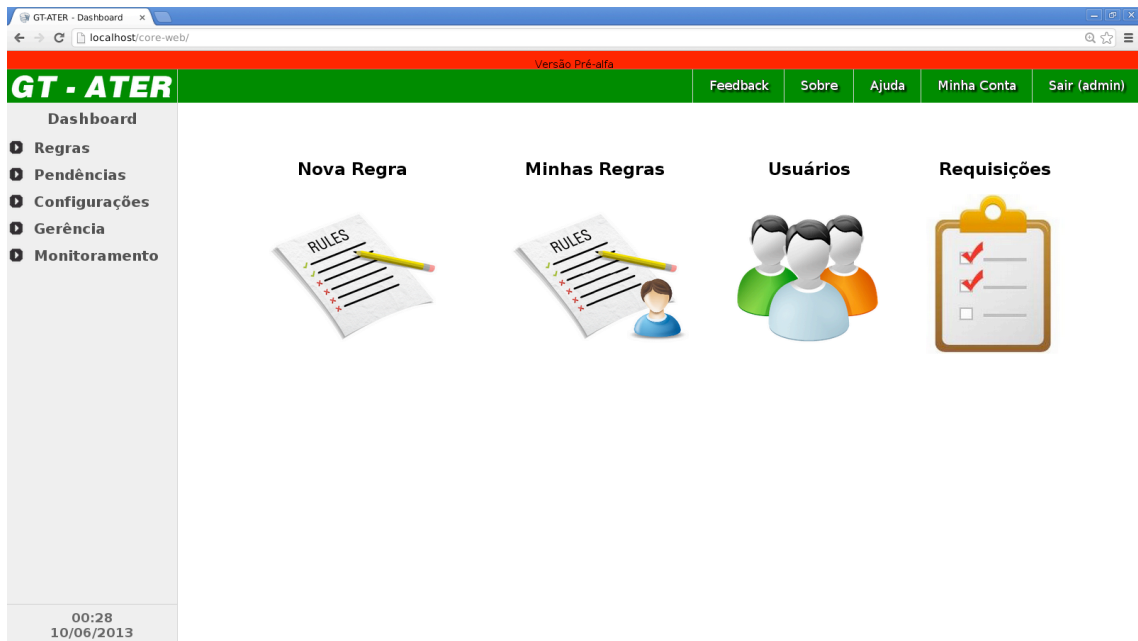


Figura 7: *Dashboard* na visão do gerente local

As opções **Nova Regra** e **Minhas Regras** apresentam funcionalidades semelhantes às opções de mesmo nome no *Dashboard* do cliente. A diferença, no entanto, é que regras criadas pelo gerente local são automaticamente instaladas nos RACEs, sem a necessidade de aprovação.

A opção **Usuários** permite ao gerente local visualizar todos os usuários da instituição sob sua administração, como mostrado na Figura 8. É possível também adicionar novos usuários, como ilustrado na Figura 9.



Figura 8: Lista de usuários

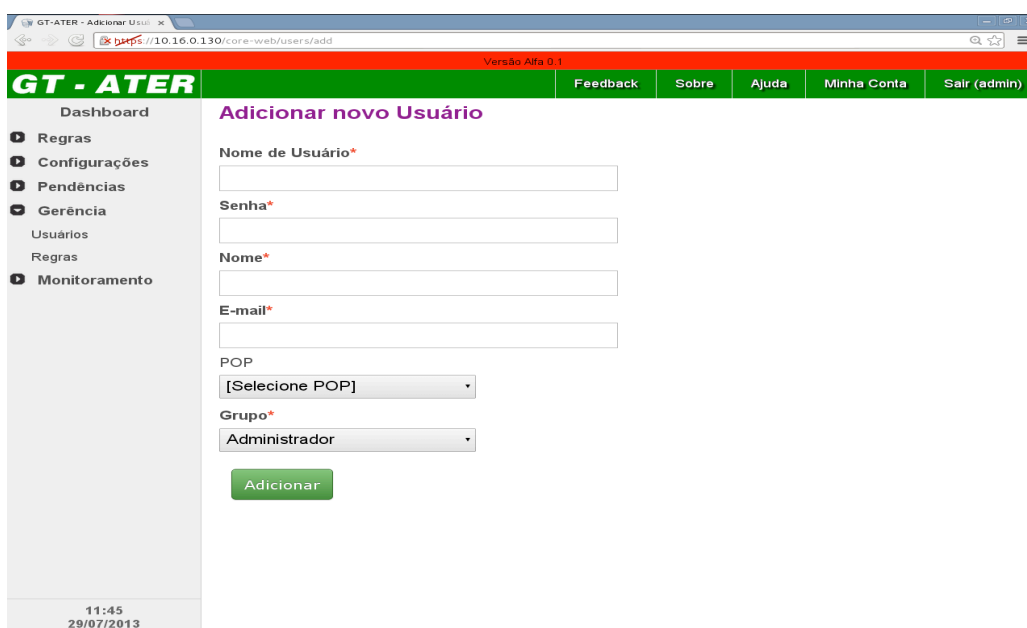


Figura 9: Adição de novo usuário

A opção **Requisições** na Figura 7 é acessada pelo gerente local para aprovar a criação de contas e também regras que foram solicitadas por usuários da instituição sob sua administração. A Figura 10 ilustra a visualização de contas pendentes de aprovação. Nesta tela, o gerente local pode aprovar uma solicitação de criação de conta, recusá-la ou mesmo visualizar detalhes do requerente.



Figura 10: Contas pendentes de aprovação



A Figura 11 ilustra a visualização das regras pendentes de aprovação, criadas por usuários pertencentes à instituição sob a administração do gerente local. Nesta tela, o gerente local poderá aprovar a regra ou removê-la.

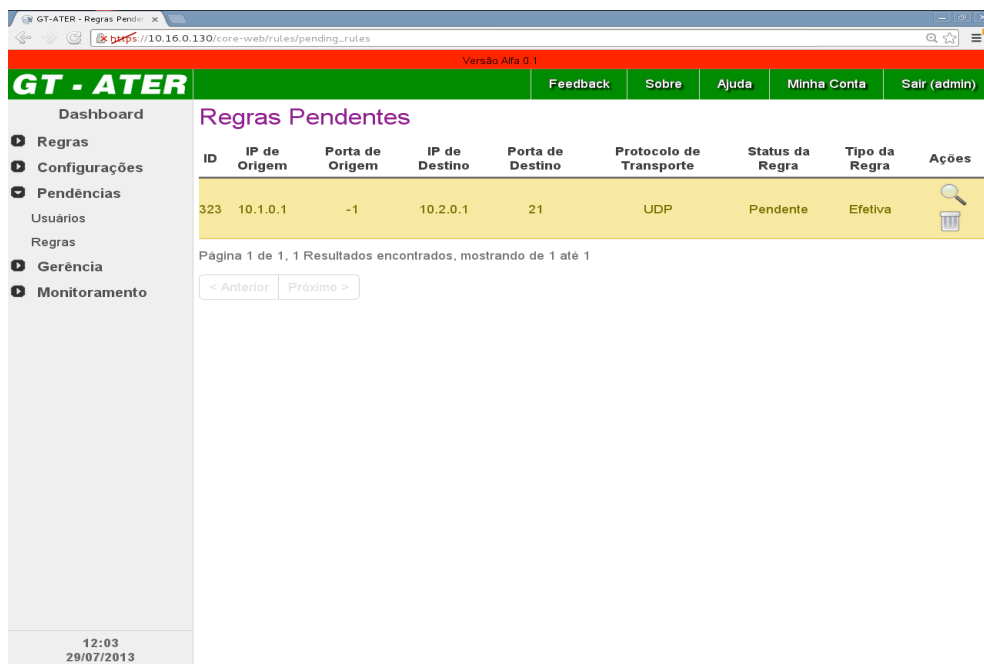


Figura 11: Regras pendentes de aprovação

Finalmente, na canto esquerda do *Dashboard*, o gerente local tem a opção de monitorar as regras e os circuitos criados, através da opção **Monitoramento**. A Figura 12 ilustra a opção de monitoramento de regras. Selecionando uma regra particular e clicando no ícone de visualização (lupa), o gerente local será direcionado para a tela mostrada na Figura 13, onde é possível visualizar gráficos de vazão do tráfego que casou com a regra nos últimos 10 minutos. Também é possível visualizar gráficos de vazão na última hora, dia e semana.

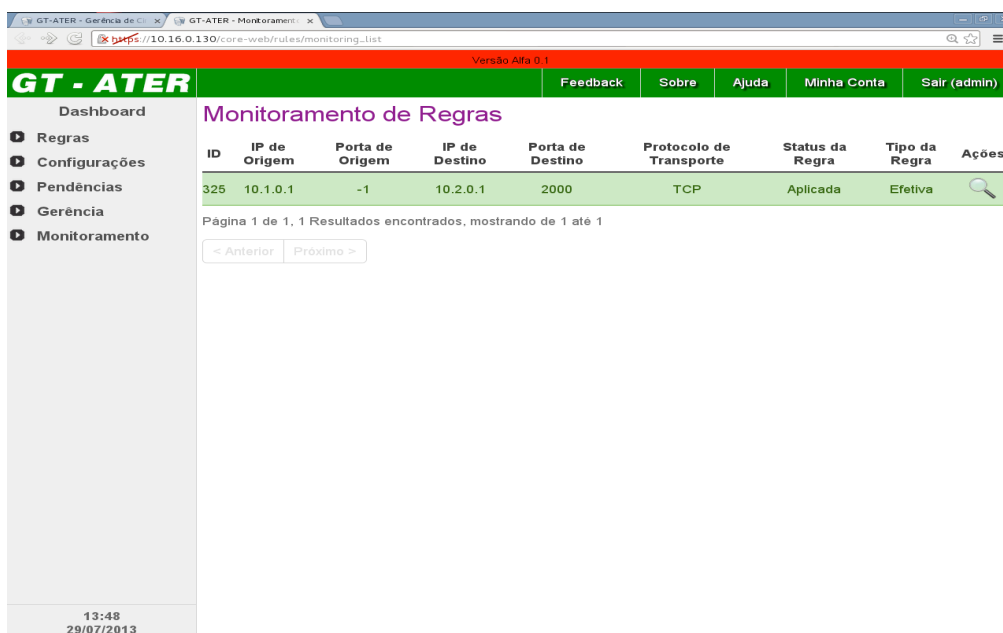


Figura 12: Monitoramento de regras

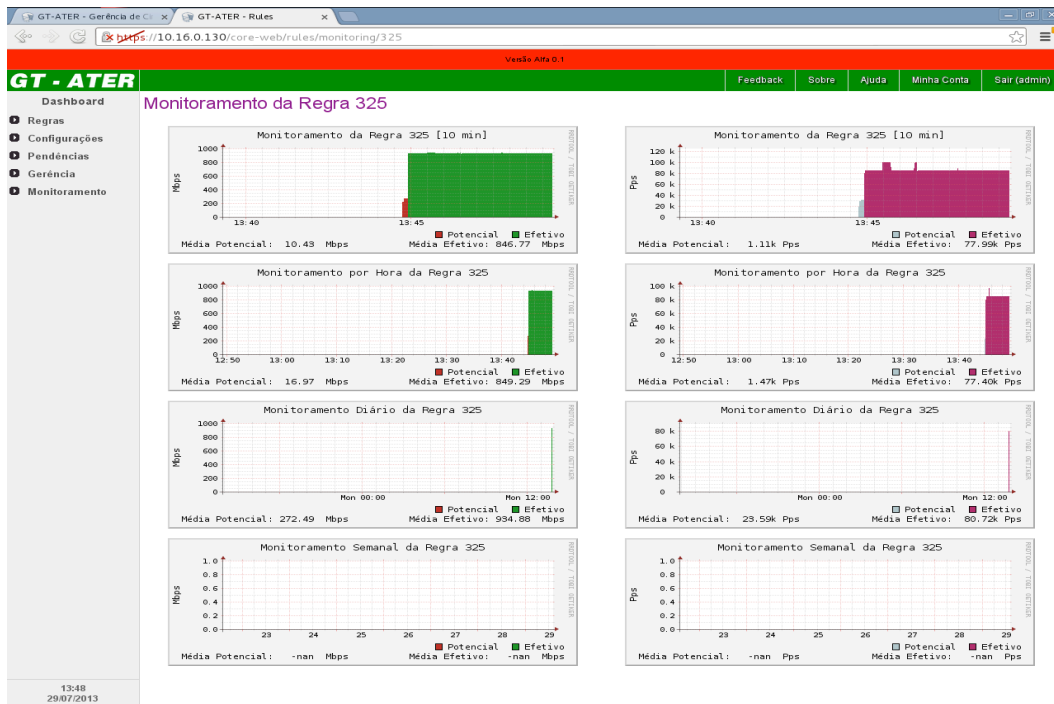


Figura 13: Visualização de gráficos de vazão de uma regra

Além das regras, é possível também monitorar circuitos. A Figura 14 ilustra o gráfico de vazão para um circuito específico. Essa opção é acessada através da guia **Monitoramento**, selecionando-se a opção **Circuitos**.

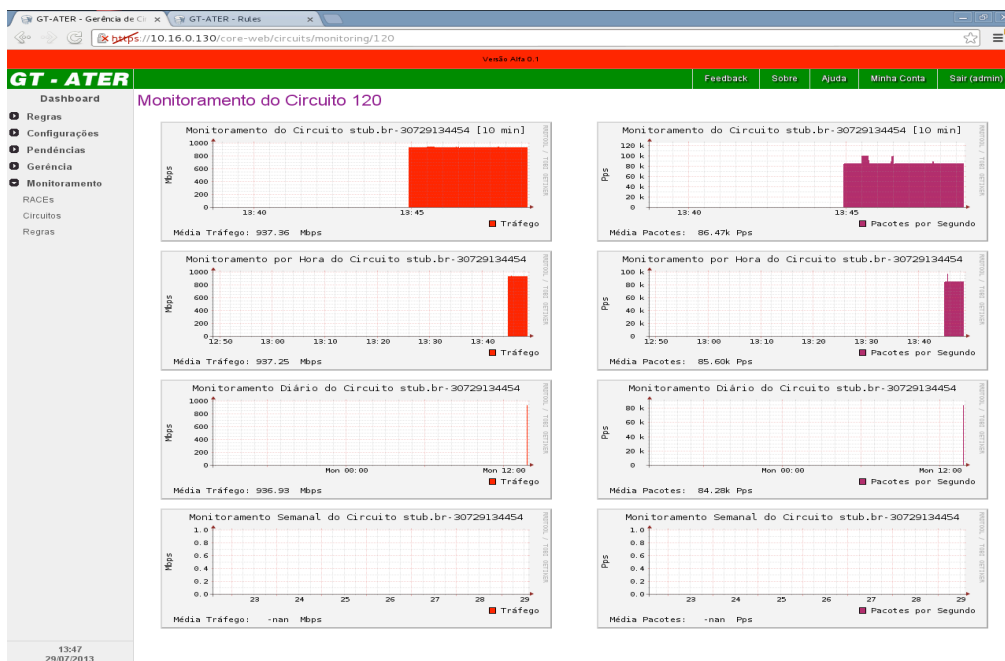


Figura 14: Visualização de gráficos de vazão de uma circuito

#### 1.4 Portal na Visão do Administrador

O perfil administrador pode executar todas as tarefa permitidas a um gerente local, além da permissão para configurar o serviço. Nesta seção, descreveremos apenas a opção de

configuração de serviço, uma vez que as outras opções são semelhantes a aquelas já descritas na visão do gerente local.

Na configuração do serviço, são definidas as informações básicas para o funcionamento do ATER. Para realizar qualquer tipo de configuração, a partir do *Dashboard* do administrador, é necessário clicar na guia **Configurações**, no menu à esquerda do *Dashboard*, como mostra a Figura 14. É possível configura as seguintes informações:

- Domínios: indicam localizações, geralmente POPs, onde o serviço está implantado.
- Equipamentos: contem informações sobre os equipamentos onde o SE-CIPÓ está implantado. Correspondem aos *switches* do POD.
- Instituições: contem informações das instituições usuárias do serviço.
- RACEs: contem informações sobre os RACEs, equipamentos onde são instaladas as regras.
- Portas: contem informações sobre as portas dos equipamentos (POD) nas quais se conectam instituições usuárias do serviço.
- Blocos de Endereço: contem informações dos blocos de endereço IPv4 associados a cada instituição usuária do serviço.
- Perfil de tráfego: contem informações dos perfis de tráfego. O cadastro de perfis de tráfego permite maior facilidade no momento de criação da regra, pois, para cada perfil, têm-se valores padrões associados ao protocolo de transporte e a porta de destino.
- Parâmetros do Serviço: permite configurar parâmetros como status do CORE, duração padrão de um circuito, IP da máquina onde executa o CORE\_OSCARS\_Driver, dentre outros.

The screenshot displays the 'GT-ATER' web application interface. At the top, there is a navigation bar with the logo 'GT-ATER' and the version 'Versão Alfa 0.1'. Below this, there are links for 'Feedback', 'Sobre', 'Ajuda', 'Minha Conta', and 'Sair (admin)'. On the left side, a sidebar menu is visible with categories like 'Dashboard', 'Regras', 'Configurações', 'Pendências', 'Gerência', and 'Monitoramento'. The 'Configurações' section is expanded, showing sub-items: 'Domínios', 'Equipamentos', 'Instituições', 'Portas', 'RACEs', 'Blocos de Endereços', and 'Perfis de Tráfego'. The main content area is titled 'Lista de Domínios' and features a table with the following data:

ID	Estado	Ações
8	GO	[Search] [Edit] [Delete]
9	RJ	[Search] [Edit] [Delete]
11	ATER	[Search] [Edit] [Delete]

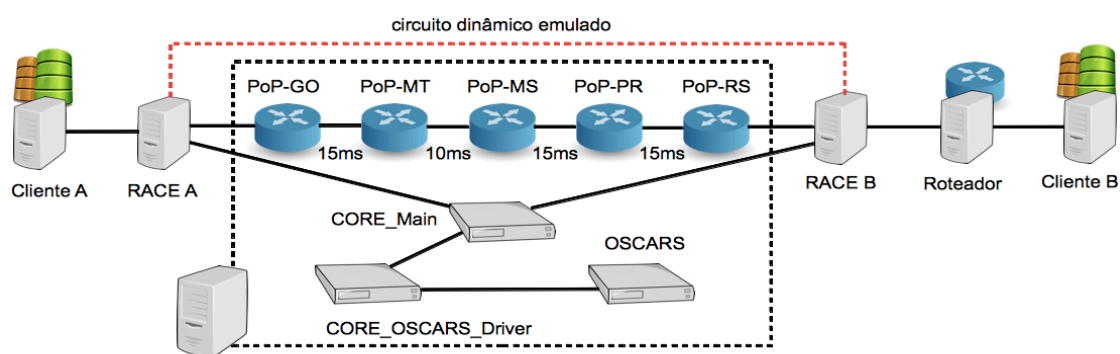
Below the table, there is a pagination control showing 'Página 1 de 1, 3 Resultados encontrados, mostrando de 1 até 3' and buttons for '< Anterior' and 'Próximo >'. At the bottom left of the interface, the time '10:28' and date '29/07/2013' are displayed.

Figura 14: Configuração do Serviço

## 2. Resultados dos testes

Os testes apresentados nesta seção foram realizados de acordo com o plano de testes proposto no relatório RT2 com algumas alterações descritas a seguir. Vale lembrar que o plano de testes na RNP, também chamado de Estudo de caso, está previsto para execução posterior a este relatório. Portanto, seus resultados serão entregues em relatório posterior.

Foi elaborado um ambiente de testes emulado maior e mais completo, em termos de variedade de características da rede, que o proposto originalmente no relatório RT2. Esse ambiente é ilustrado pela Figura 1. Esse ambiente é um superconjunto do ambiente virtualizado e, portanto, todos os testes foram realizados no ambiente emulado.



**Figura 1: Topologia final do ambiente emulado.**

Nessa versão final do ambiente emulado, tentamos representar um cenário comum da RNP. Para tanto, escolhemos alguns PoPs e realizamos algumas medidas de atrasos entre eles. Essas medidas foram usadas para configurar os atrasos artificiais ilustrados na Figura 1. Incluímos também os cenários nos quais há conexão direta do cliente ao RACE e o que há um roteador entre o RACE e o cliente. Embora possam existir mais roteadores entre o cliente e o RACE, sob o ponto de vista do ATER não há diferença. O RACE precisa distinguir apenas dois casos: entrega direta e entrega indireta ao cliente final.

O ambiente emulado também foi preparado para apresentar configurações com e sem VLANs (rotuladas ou *tagged*) conforme ilustrado pela Figura 2. As VLANs sem rótulo mostradas na Figura 2 equivalem à ausência de VLANs, pois os rótulos são usados apenas internamente no *switch*.

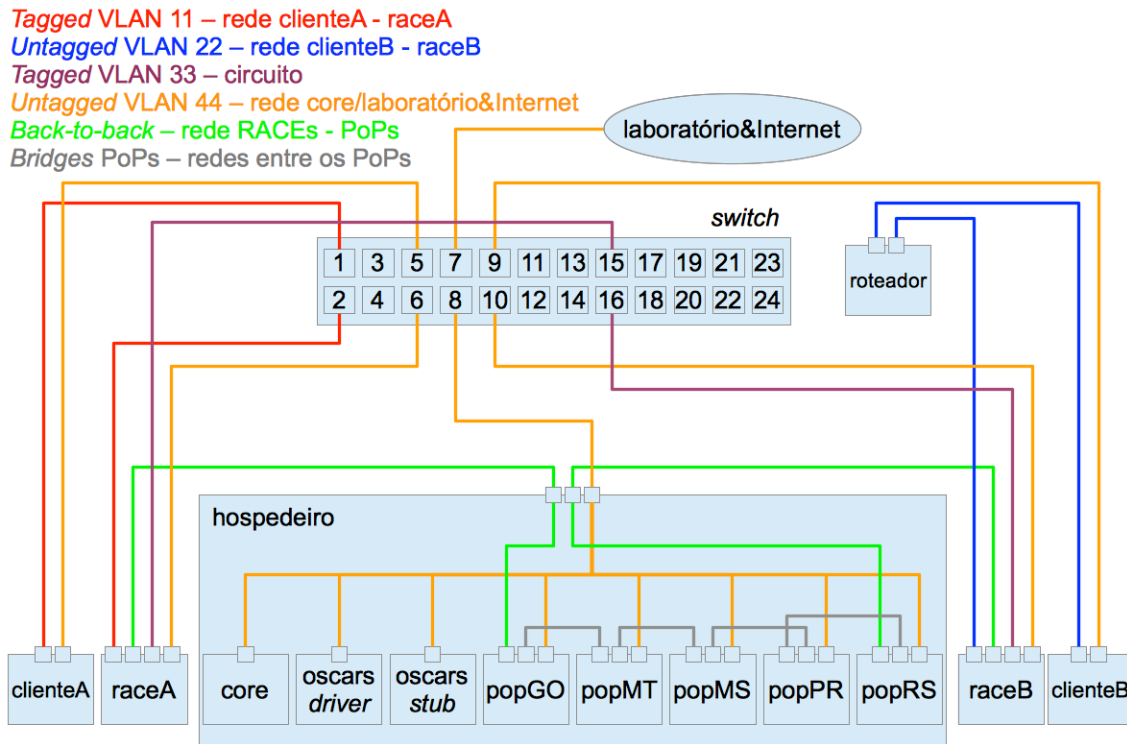


Figura 2: Configuração física do ambiente emulado.

#### Testes de funcionalidades e desempenho básico

<b>Nome</b>	Criação de regra
<b>Componentes envolvidos</b>	Módulo Web, Módulo de Encaminhamento e Filtragem, Canal Seguro
<b>Descrição</b>	Esse teste consiste em um usuário criar uma regra através da interface Web do serviço ATER. Após a criação, a regra é enviada para os RACEs, os quais deverão aguardar a chegada do primeiro pacote para solicitar a criação de um circuito dinâmico.
<b>Resultados esperados</b>	O usuário deve poder listar a regra criada, visualizando seu estado. Nos RACEs, deve ser possível visualizar uma nova regra na tabela do ebttables. Tanto CORE quanto RACEs também registram em seus logs de operação a criação da regra.

Este teste alcançou os resultados esperados, conforme mostrado na sequência de passos a seguir. Os passos são ilustrados através de telas capturadas ao longo da execução do teste.

Passo 1: criar uma nova regra efetiva, com o perfil de cliente.

Versão Alfa 0.1

**GT - ATER** Feedback Sobre Ajuda

Dashboard

**Nova Regra**

Perfil pré-definido [Selecione] ▾

---

Protocolo de Rede\* IPv4 ▾

Host  Rede

IP de Origem\*

Host  Rede

IP de Destino\*

---

Configurar Protocolo de Transporte e Portas de Origem/Destino

Protocolo de Transporte\* TCP ▾

Porta de Origem

Porta de Destino

---

Largura de banda do canal  ▾

Tipo da Regra  ▾

Passo 2: verificar se a regra efetiva foi criada e está aguardando aprovação.

Versão Alfa 0.1

**GT - ATER** Feedback Sobre Ajuda Minha Conta Sair (ater1)

Dashboard

**Regra Solicitada com sucesso!**

**Minhas Regras Pendentes**

ID	IP de Origem	Porta de Origem	IP de Destino	Porta de Destino	Protocolo de Transporte	Status da Regra	Tipo da Regra	Ações
303	10.1.0.1	-1	10.2.0.1	2000	TCP	Pendente	Efetiva	 

Página 1 de 1, 1 Resultados encontrados, mostrando de 1 até 1

< Anterior Próximo >

Passo 3: como administrador, verificar as regras pendentes de aprovação.

Versão Alfa 0.1

**GT - ATER** Feedback Sobre Ajuda Minha Conta Sair (admin)

Dashboard

**Regras Pendentes**

ID	IP de Origem	Porta de Origem	IP de Destino	Porta de Destino	Protocolo de Transporte	Status da Regra	Tipo da Regra	Ações
303	10.1.0.1	-1	10.2.0.1	2000	TCP	Pendente	Efetiva	 

Página 1 de 1, 1 Resultados encontrados, mostrando de 1 até 1

< Anterior Próximo >

Passo 4: como administrador, verificar detalhes sobre uma regra efetiva pendente de aprovação.

Versão Alfa 0.1

**GT - ATER** Feedback Sobre

Dashboard

**Regra #303**

ID	303
Data de Criação	2013-07-19
IP de Origem	10.1.0.1
Porta de Origem	-1
IP de Destino	10.2.0.1
Porta de Destino	2000
Protocolo de Transporte	TCP
Protocolo de Rede	IPv4
Largura de banda do canal	1000
Usuário	ater1

**Opções Administrativas da Regra**

Circuit Duration

Passo 5: como administrador, aprovar a regra efetiva pendente.

Versão Alfa 0.1

**GT - ATER** Feedback Sobre

Dashboard

**Regra criada com sucesso!**

**Regra #303**

ID	303
Data de Criação	2013-07-19
IP de Origem	10.1.0.1
Porta de Origem	-1
IP de Destino	10.2.0.1
Porta de Destino	-1
Protocolo de Transporte	-1
Protocolo de Rede	IPv4
Largura de banda do canal	1000
Usuário	ater1

Passo 6: como cliente, confirmar a aprovação da regra efetiva que estava pendente.

Versão Alfa 0.1

**GT - ATER** Feedback Sobre Ajuda Minha Conta Sair (ater1)

Dashboard

**Minhas Regras**

ID	IP de Origem	Porta de Origem	IP de Destino	Porta de Destino	Protocolo de Transporte	Status da Regra	Tipo da Regra	Ações
303	10.1.0.1	-1	10.2.0.1	-1	-1	Aplicada	Efetiva	

Página 1 de 1, 1 Resultados encontrados, mostrando de 1 até 1

Passo 7: verificar regra efetiva no RACE mais próximo ao IP de origem.

```

root@clientA:~
root@raceA:~# ebttables -L
Bridge table: filter

Bridge chain: INPUT, entries: 0, policy: ACCEPT

Bridge chain: FORWARD, entries: 1, policy: ACCEPT
-p IPv4 --ip-src 10.1.0.1 --ip-dst 10.2.0.1 --ip-proto tcp --ip-dport 2000 --u-log-prefix "" --u-log-nlgroup 5 --u-log-cprange default_cprange --u-log-qthreshold 1 -j ACCEPT

Bridge chain: OUTPUT, entries: 0, policy: ACCEPT
root@raceA:~#

```

Passo 8: verificar regra efetiva complementar no RACE mais próximo ao IP de destino.

```
root@clientA:~# ebttables -L
Bridge table: filter

Bridge chain: INPUT, entries: 0, policy: ACCEPT

Bridge chain: FORWARD, entries: 1, policy: ACCEPT
-p IPv4 --ip-src 10.2.0.1 --ip-dst 10.1.0.1 --ip-PROTO tcp --ip-sport 2000 --uLog-prefix "" --uLog-nlgroup 5 --uLog-cprange default_cprange --uLog-qthreshold 1 -j ACCEPT

Bridge chain: OUTPUT, entries: 0, policy: ACCEPT
root@raceB:~#
```

Passo 9: verificar, no log do CORE, o envio da regra efetiva complementar para o RACE mais próximo ao IP de destino:

```
ameters.
[DEBUG - 19/07/2013 19:39:57] NiceSSH.exec - Executing a command in the host.
[DEBUG - 19/07/2013 19:39:57] NiceSSH.exec - Command to be executed:
/root/race/race_client/src/bin/race_client --name setRule --id 303 --action NEW --extremity DESTINATION --source-ip 10.2.0.1 --destination-ip 10.1.0.1 --source-netmask 32 --destination-netmask 32 --network-protocol IPv4 --transport-protocol TCP --source-port 2000 --type EFFECTIVE; echo '$_result('$?)'.
[DEBUG - 19/07/2013 19:39:57] NiceSSH.getConnectionStatus - Getting the connection status.
[DEBUG - 19/07/2013 19:39:57] NiceSSH.getConnectionStatus - The connection is active.
[DEBUG - 19/07/2013 19:39:57] NiceSSH.exec - SSH command executed.
[DEBUG - 19/07/2013 19:39:57] NiceSSH.exec - Command execution result = '[DEBUG - 19/07/2013 19:39:57] Trying connection with
```

Passo 10: verificar, no log do CORE, o envio da regra efetiva para o RACE mais próximo ao IP de origem.

```
ameters.
[DEBUG - 19/07/2013 19:39:57] NiceSSH.exec - Executing a command in the host.
[DEBUG - 19/07/2013 19:39:57] NiceSSH.exec - Command to be executed:
/root/race/race_client/src/bin/race_client --name setRule --id 303 --action NEW --extremity SOURCE --source-ip 10.1.0.1 --destination-ip 10.2.0.1 --source-netmask 32 --destination-netmask 32 --network-protocol IPv4 --transport-protocol TCP --destination-port 2000 --type EFFECTIVE; echo '$_result('$?)'.
[DEBUG - 19/07/2013 19:39:57] NiceSSH.getConnectionStatus - Getting the connection status.
[DEBUG - 19/07/2013 19:39:57] NiceSSH.getConnectionStatus - The connection is active.
[DEBUG - 19/07/2013 19:39:57] NiceSSH.exec - SSH command executed.
[DEBUG - 19/07/2013 19:39:57] NiceSSH.exec - Command execution result = '[DEBUG - 19/07/2013 19:39:57] Trying connection with
```

Passo 11: verificar, no log do RACE mais próximo ao IP de origem, o recebimento da regra efetiva.

```
[DEBUG - 19/07/2013 19:39:57] SET RULE RECEIVED!
[DEBUG - 19/07/2013 19:39:57] Rule values:
[DEBUG - 19/07/2013 19:39:57] rule-id = 303
[DEBUG - 19/07/2013 19:39:57] rule-action = 70
[DEBUG - 19/07/2013 19:39:57] rule-extremity = 80
[DEBUG - 19/07/2013 19:39:57] rule-state = 34
[DEBUG - 19/07/2013 19:39:57] rule-type = 90
[DEBUG - 19/07/2013 19:39:57] rule-network_protocol = 40
[DEBUG - 19/07/2013 19:39:57] rule-transport_protocol = 50
[DEBUG - 19/07/2013 19:39:57] rule-source_ip = 10.1.0.1
[DEBUG - 19/07/2013 19:39:57] rule-destination_ip = 10.2.0.1
[DEBUG - 19/07/2013 19:39:57] rule-source_netmask = 32
[DEBUG - 19/07/2013 19:39:57] rule-destination_netmask = 32
[DEBUG - 19/07/2013 19:39:57] rule-source_port = -1:-1
[DEBUG - 19/07/2013 19:39:57] rule-destination_port = 2000:2000
[DEBUG - 19/07/2013 19:39:57] rule-time_stamp = 1374273597
[DEBUG - 19/07/2013 19:39:57] RULE with ID = 303 REGISTERED!!!
[DEBUG - 19/07/2013 19:39:57] rule[0] = 303 50 40 10.1.0.1 10.2.0.1 -1:-1 2000:2000
[DEBUG - 19/07/2013 19:39:57] ebttables -A FORWARD --protocol IPv4 --ip-protocol TCP --ip-source 10.1.0.1/32 --ip-destination 10.2.0.1/32 --ip-destination-port 2000 --uLog-nlgroup 5 -j ACCEPT
[DEBUG - 19/07/2013 19:39:57] ebttables - system message returned 0
[DEBUG - 19/07/2013 19:39:57] APPLY RULE -A SUCCESS for RULE with ID = 303
[DEBUG - 19/07/2013 19:39:57] RULE APPLIED and REGISTERED!!!
[DEBUG - 19/07/2013 19:39:57] RULE with ID = 303 REGISTERED!!!
[DEBUG - 19/07/2013 19:39:57] rule[0] = 303 50 40 10.2.0.1 10.1.0.1 2000:2000 -1:-1
[DEBUG - 19/07/2013 19:39:57] RULE REGISTERED!!!
```

Passo 12: verificar, no log do RACE mais próximo ao IP de destino, o recebimento da regra efetiva complementar.



```

root@clientA:~
root@raceA:~/race/race...
root@raceB:~/race/race...
root@clientB:~
root@core:~
root@raceA:~
root@raceB:~

[DEBUG - 19/07/2013 19:39:57] SET RULE RECEIVED!
[DEBUG - 19/07/2013 19:39:57] Rule values:
[DEBUG - 19/07/2013 19:39:57] rule-id = 303
[DEBUG - 19/07/2013 19:39:57] rule-action = 70
[DEBUG - 19/07/2013 19:39:57] rule-extremity = 81
[DEBUG - 19/07/2013 19:39:57] rule-state = 34
[DEBUG - 19/07/2013 19:39:57] rule-type = 90
[DEBUG - 19/07/2013 19:39:57] rule-network_protocol = 40
[DEBUG - 19/07/2013 19:39:57] rule-transport_protocol = 50
[DEBUG - 19/07/2013 19:39:57] rule-source_ip = 10.2.0.1
[DEBUG - 19/07/2013 19:39:57] rule-destination_ip = 10.1.0.1
[DEBUG - 19/07/2013 19:39:57] rule-source_netmask = 32
[DEBUG - 19/07/2013 19:39:57] rule-destination_netmask = 32
[DEBUG - 19/07/2013 19:39:57] rule-source_port = 2000:2000
[DEBUG - 19/07/2013 19:39:57] rule-destination_port = -1:-1
[DEBUG - 19/07/2013 19:39:57] rule-time_stamp = 1374273597
[DEBUG - 19/07/2013 19:39:57] RULE with ID = 303 REGISTERED!!!
[DEBUG - 19/07/2013 19:39:57] rule[0] = 303 50 40 10.2.0.1 10.1.0.1 2000:2000 -1:-1
[DEBUG - 19/07/2013 19:39:57] ebttables -A FORWARD --protocol IPv4 --ip-protocol TCP --ip-source 10.2.0.1/32 --ip-destination 10.1.0.1/32 --ip-source-port 2000 --u-log-nlgroup 5 -j ACCEPT
[DEBUG - 19/07/2013 19:39:57] ebttables - system message returned 0
[DEBUG - 19/07/2013 19:39:57] APPLY RULE -A SUCCESS for RULE with ID = 303
[DEBUG - 19/07/2013 19:39:57] RULE APPLIED and REGISTERED!!!
[DEBUG - 19/07/2013 19:39:57] RULE with ID = 303 REGISTERED!!!
[DEBUG - 19/07/2013 19:39:57] rule[0] = 303 50 40 10.1.0.1 10.2.0.1 -1:-1 2000:2000
[DEBUG - 19/07/2013 19:39:57] RULE REGISTERED!!!

```

<b>Nome</b>	Operação de regra efetiva no ambiente virtual
<b>Componentes envolvidos</b>	Módulo de Criação de Circuitos, Módulo de Encaminhamento e Filtragem, Canal Seguro
<b>Descrição</b>	Este teste consiste em enviar tráfego que combine com uma regra efetiva. Quando um RACE detecta o primeiro pacote que combina com uma regra efetiva, ele entra em contato com o CORE para solicitar a criação de um circuito dinâmico.
<b>Resultados esperados</b>	Após a criação do circuito, será possível observar um caminho fim-a-fim com menos roteadores. O atraso fim-a-fim também deve ser reduzido. A criação do circuito dinâmico é registrada em <i>log</i> pelo CORE. Vale lembrar que o OSCARS utilizado nesse teste emprega uma topologia simulada sem falhas.

Este teste alcançou os resultados esperados, conforme mostrado na sequência de passos a seguir. Os passos são ilustrados através de telas capturadas ao longo da execução do teste.

Passo 1: verificar o caminho fim-a-fim antes da criação do circuito.

```

root@clientA:~
root@raceA:~/ra...
root@raceB:~/ra...
root@clientB:~
root@core:~
root@raceA:~

Every 1.0s: traceroute 10.1.0.1 -s 10.2.0.1 -w 0.5 -I -q 1

traceroute to 10.1.0.1 (10.1.0.1), 30 hops max, 60 byte packets
 1  routerB (10.2.0.11)  0.126 ms
 2  rs-3g-oi.bkb.rnp.br (10.50.0.11)  0.536 ms
 3  pr-3g-oi.bkb.rnp.br (10.40.0.1)  15.780 ms
 4  ms-3g-oi.bkb.rnp.br (10.30.0.11)  31.005 ms
 5  mt-3g-oi.bkb.rnp.br (10.20.0.1)  41.220 ms
 6  go-3g-oi.bkb.rnp.br (10.10.0.11)  56.446 ms
 7  clientA (10.1.0.1)  56.988 ms

```

Passo 2: como administrador, criar uma regra efetiva que já é pré-aprovada.

Versão Alfa 0.1

**GT - ATER** Feedback Sobre Ajuda

Dashboard

**Regras**

- Nova Regra
- Todas Regras
- Regras Pendentes

**Configurações**

**Pendências**

**Gerência**

**Monitoramento**

Regra criada com sucesso!

### Nova Regra

Perfil pré-definido: [Selecione]

---

Protocolo de Rede\*: IPv4

Host  Rede

IP de Origem\*: 10 . 2 . 0 . 1

Host  Rede

IP de Destino\*: 10 . 1 . 0 . 1

---

Configurar Protocolo de Transporte e Portas de Origem/Destino

Protocolo de Transporte\*: ICMP

---

Largura de banda do canal: 100 Mbps

Duração do Circuito: 1800

Tipo da Regra: Efetiva

Adicionar Regra

**Passo 3: verificar a criação do circuito no log do CORE.**

```

root@clientA: ~ * root@raceA: ~/ra... * root@raceB: ~/ra... * root@clientB: ~ * root@core: ~ * root@raceA: ~ * root@raceB: ~ * root@core: ~ * micaielmello@not...
[DEBUG - 19/07/2013 20:43:19] NiceSSH.exec - Executing a command in the host.
[DEBUG - 19/07/2013 20:43:19] NiceSSH.exec - Command to be executed:
/usr/local/core_oscars_driver/bin/oscars_driver -c setCircuit -a 'NEW' -g 'null' -s 'urn:ogf:network:domain=stub.br:node=SDR:po
rt=ge-2/3/4:link=10.0.9.1' -t '-1' -d 'urn:ogf:network:domain=stub.br:node=BSA:port=x-3/1/1:link=10.0.0.105' -e '-1' -b '100'
-x '1374277399' -y '1374279199' -l '' | grep -v php.
[DEBUG - 19/07/2013 20:43:19] NiceSSH.getConnectionStatus - Getting the connection status.
[DEBUG - 19/07/2013 20:43:19] NiceSSH.getConnectionStatus - The connection is active.
[DEBUG - 19/07/2013 20:43:19] NiceSSH.exec - SSH command executed.
[DEBUG - 19/07/2013 20:43:55] NiceSSH.exec - Command execution result = 'status: 0
globalReservationId: stub.br-1483
reservationStatus: ACTIVE
srcUrn: urn:ogf:network:domain=stub.br:node=SDR:port=ge-2/3/4:link=10.0.9.1
isSrcTagged: true
srcTag: 1209
destUrn: urn:ogf:network:domain=stub.br:node=BSA:port=x-3/1/1:link=10.0.0.105
isDestTagged: true
destTag: 1209
path: urn:ogf:network:domain=stub.br:node=SDR:port=ge-2/3/4:link=10.0.9.1;urn:ogf:network:domain=stub.br:node=SDR:port=x-3/2/0
:link=10.0.0.117;urn:ogf:network:domain=stub.br:node=BHE:port=x-2/1/0:link=10.0.0.118;urn:ogf:network:domain=stub.br:node=BHE:
port=x-2/1/1:link=10.0.0.102;urn:ogf:network:domain=stub.br:node=BSA:port=x-2/1/1:link=10.0.0.101;urn:ogf:network:domain=stub
.br:node=BSA:port=x-3/1/1:link=10.0.0.105;
bandwidth: 100
createTime: 1374277406 (Sex Jul 19 20:43:26 BRT 2013)
startTime: 1374277399 (Sex Jul 19 20:43:19 BRT 2013)
endTime: 1374279199 (Sex Jul 19 21:13:19 BRT 2013)
startExec: 1374277399 (Fri, 19 Jul 2013 20:43:19 -0300)
endExec: 1374277435 (Fri, 19 Jul 2013 20:43:55 -0300)
execTime: 36 s.
  
```

**Passo 4: verificar o caminho fim-a-fim após a criação do circuito.**

```

root@clientA: ~ * root@raceA: ~/ra... * root@raceB: ~/ra... * root@clientB: ~ * root@core: ~ * root@raceA: ~
Every 1.0s: traceroute 10.1.0.1 -s 10.2.0.1 -w 0.5 -I -q 1
traceroute to 10.1.0.1 (10.1.0.1), 30 hops max, 60 byte packets
 1  routerB (10.2.0.11)  0.208 ms
 2  clientA (10.1.0.1)  1.051 ms
  
```

<b>Nome</b>	Operação de regra de monitoramento no ambiente virtual
<b>Componentes envolvidos</b>	Módulo de Encaminhamento e Filtragem, Coleta de Estatística, Canal Seguro
<b>Descrição</b>	Este teste consiste em enviar tráfego que combine com uma regra de monitoramento. Os pacotes que combinam devem ser contabilizados e as estatísticas referentes

	devem ser enviadas para o CORE.
<b>Resultados esperados</b>	As estatísticas referentes ao fluxo transmitido devem estar disponíveis no banco de dados do CORE.

Este teste alcançou os resultados esperados, conforme mostrado na sequência de passos a seguir. Os passos são ilustrados através de telas capturadas ao longo da execução do teste.

Passo 1: como administrador, criar uma regra de monitoramento.

The screenshot shows the 'Nova Regra' configuration page in the GT-ATER interface. The page includes a sidebar with navigation options like 'Regras', 'Configurações', and 'Monitoramento'. The main content area shows a success message and the following configuration fields:

- Perfil pré-definido: [Selecione]
- Protocolo de Rede\*: IPv4
- Host (selected) / Rede (radio buttons)
- Ip de Origem\*: 10 . 1 . 0 . 1
- IP de Destino\*: 10 . 2 . 0 . 1
- Configurar Protocolo de Transporte e Portas de Origem/Destino (checked)
- Protocolo de Transporte\*: TCP
- Porta de Origem: [ ]
- Porta de Destino: 2000
- Largura de banda do canal: 1 Gbps
- Tipo da Regra: Monitoramento
- Duração do Circuito: 1800
- Adicionar Regra button

Passo 2: confirmar a criação da regra de monitoramento.

The screenshot shows the 'Minhas Regras' page in the GT-ATER interface. The page displays a table with the following data:

ID	IP de Origem	Porta de Origem	IP de Destino	Porta de Destino	Protocolo de Transporte	Status da Regra	Tipo da Regra	Ações
215	10.1.0.1	-1	10.2.0.1	2000	TCP	Aplicada	Monitoramento	[Search] [Delete]

Below the table, it shows 'Página 1 de 1, 1 Resultados encontrados, mostrando de 1 até 1' and navigation buttons for '< Anterior' and 'Próximo >'.

Passo 3: verificar as estatísticas de tráfego antes de iniciar o envio de dados. É importante lembrar que a regra de monitoramento não gera circuito e, portanto, o casamento do tráfego com a regra é o alvo do monitoramento.



Passo 4: enviar tráfego que combine com a regra de monitoramento definida.

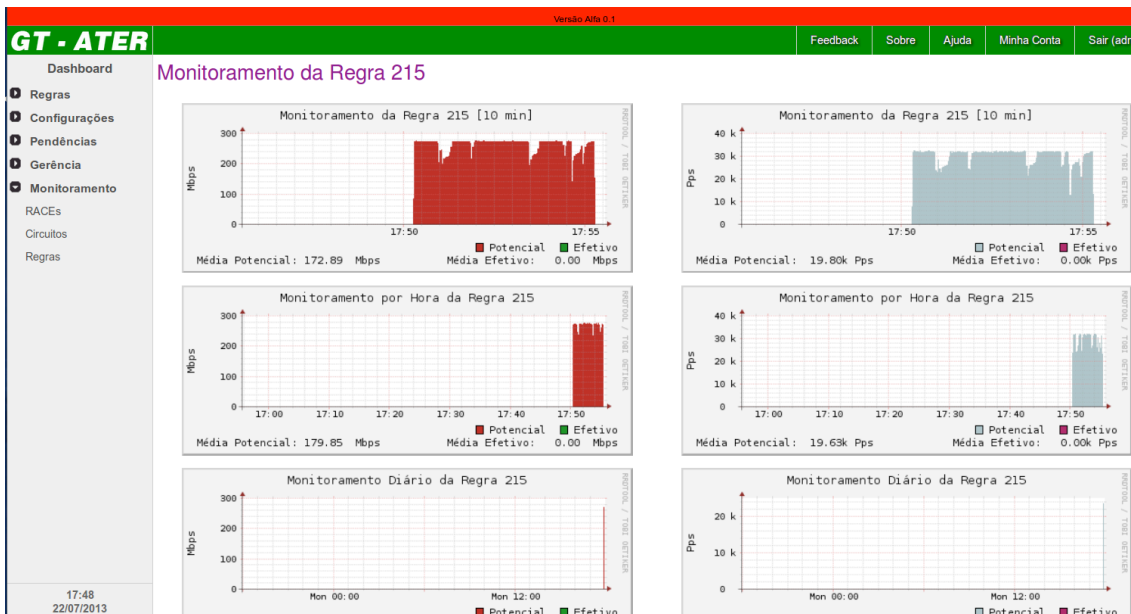
```

root@clientA: ~
root@raceA: ~/race/...
root@raceB: ~/race/...
root@clientB: ~
root@core: ~
root

root@clientA:~# iperf -c 10.2.0.1 -t 300 -i 17 -p 2000
-----
Client connecting to 10.2.0.1, TCP port 2000
TCP window size: 23.5 KByte (default)
-----
[ 3] local 10.1.0.1 port 55290 connected with 10.2.0.1 port 2000
[ ID] Interval            Transfer          Bandwidth
[ 3]  0.0-17.0 sec        511 MBytes       252 Mb/s
[ 3] 17.0-34.0 sec        546 MBytes       270 Mb/s
[ 3] 34.0-51.0 sec        447 MBytes       221 Mb/s
[ 3] 51.0-68.0 sec        479 MBytes       236 Mb/s
[ 3] 68.0-85.0 sec        547 MBytes       270 Mb/s
[ 3] 85.0-102.0 sec       498 MBytes       246 Mb/s
[ 3] 102.0-119.0 sec      547 MBytes       270 Mb/s
[ 3] 119.0-136.0 sec      548 MBytes       270 Mb/s
[ 3] 136.0-153.0 sec      546 MBytes       270 Mb/s
[ 3] 153.0-170.0 sec      546 MBytes       270 Mb/s
[ 3] 170.0-187.0 sec      548 MBytes       270 Mb/s
[ 3] 187.0-204.0 sec      446 MBytes       220 Mb/s
[ 3] 204.0-221.0 sec      532 MBytes       263 Mb/s
[ 3] 221.0-238.0 sec      548 MBytes       271 Mb/s
[ 3] 238.0-255.0 sec      499 MBytes       246 Mb/s
[ 3] 255.0-272.0 sec      444 MBytes       219 Mb/s
[ 3] 272.0-289.0 sec      460 MBytes       227 Mb/s
[ 3]  0.0-300.0 sec      8.83 GBytes      253 Mb/s
root@clientA:~#

```

Passo 5: verificar as estatísticas de tráfego após iniciar o envio de dados.



### Testes de desempenho avançado

<b>Nome</b>	Vazão através de circuito dinâmico emulado
<b>Componentes envolvidos</b>	Módulo Web, Módulo de Criação de Circuitos, Módulo de Encaminhamento e Filtragem, Coleta de Estatística, Canal Seguro
<b>Descrição</b>	Este teste consiste em transmitir um arquivo grande utilizando o TCP. Inicialmente, a transferência será realizada através do caminho que representa a rede convencional, ou seja, tem maior atraso e mais perdas. Posteriormente, é incluída uma regra e a transferência é repetida através do circuito dinâmico emulado.
<b>Resultados esperados</b>	O caminho com menor atraso e menos perdas deve permitir que a transferência ocorra em menor tempo.

Este teste alcançou os resultados esperados, conforme mostrado na sequência de passos a seguir. Os passos são ilustrados através de telas capturadas ao longo da execução do teste.

Passo 1: realizar uma transferência antes de definir uma regra que trate o tráfego e verificar o tempo necessário para completá-la (31 segundos).

```

root@clientA: ~
root@raceA: ~/race/...
root@raceB: ~/race/...
root@clientB: ~
root@core: ~
root@raceA: ~/race/...
root@raceB: ~/race/...
michaelmello@

root@clientA:~# wget 10.2.0.1/data_file_test.dat -O /dev/null
--2013-07-22 18:29:42-- http://10.2.0.1/data_file_test.dat
Connecting to 10.2.0.1:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 943718400 (900M) [application/x-ns-proxy-autoconfig]
Saving to: `/dev/null'

100%[=====] 943,718,400 32.3M/s in 31s
2013-07-22 18:30:13 (29.4 MB/s) - `/dev/null' saved [943718400/943718400]

root@clientA:~#

```

Passo 2: como administrador, criar uma regra efetiva que já é pré-aprovada.

Versão Alfa 0.1

**GT - ATER** Feedback Sobre Ajuda

Dashboard

**Regras**

- Nova Regra
- Todas Regras
- Regras Pendentes

**Configurações**

**Pendências**

**Gerência**

**Monitoramento**

**Regra criada com sucesso!**

### Nova Regra

Perfil pré-definido: WEB

Protocolo de Rede\*: IPv4

Host  Rede

Ip de Origem\*: 10 . 1 . 0 . 1

Host  Rede

IP de Destino\*: 10 . 2 . 0 . 1

Configurar Protocolo de Transporte e Portas de Origem/Destino

Protocolo de Transporte\*: TCP

Porta de Origem:

Porta de Destino: 80

Largura de banda do canal: 1 Gbps

Duração do Circuito: 1800

Tipo da Regra: Efetiva

**Adicionar Regra**

18:30  
22/07/2013

Passo 3: realizar uma transferência após definir uma regra que trate o tráfego. Essa transferência dispara a criação do circuito, porém parte do tráfego ainda é encaminhado pela rede convencional antes da criação do circuito. Assim, é possível perceber uma redução no tempo total de transferência (16 segundos), mas esse tempo ainda pode ser menor quando todo o tráfego é enviado através do circuito.

```

root@clientA: ~
* root@raceA: ~/race/...
* root@raceB: ~/race/...
* root@clientB: ~
* root@core: ~
* root@raceA: ~/race/...
* root@raceB: ~/race/...
* micaelmello@note

root@clientA:~# wget 10.2.0.1/data_file_test.dat -O /dev/null
--2013-07-22 18:30:45-- http://10.2.0.1/data_file_test.dat
Connecting to 10.2.0.1:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 943718400 (900M) [application/x-ns-proxy-autoconfig]
Saving to: '/dev/null'

100%[=====>] 943,718,400 111M/s in 16s
2013-07-22 18:31:02 (55.9 MB/s) - '/dev/null' saved [943718400/943718400]
root@clientA:~#

```

Passo 4: realizar uma transferência após definir a criação do circuito. É possível observar um tempo total de transferência (8,1 segundos) inferior ao do passo anterior.

```

root@clientA: ~
* root@raceA: ~/race/...
* root@raceB: ~/race/...
* root@clientB: ~
* root@core: ~
* root@raceA: ~/race/...
* root@raceB: ~/race/...
* micaelmello@note

root@clientA:~# wget 10.2.0.1/data_file_test.dat -O /dev/null
--2013-07-22 18:31:10-- http://10.2.0.1/data_file_test.dat
Connecting to 10.2.0.1:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 943718400 (900M) [application/x-ns-proxy-autoconfig]
Saving to: '/dev/null'

100%[=====>] 943,718,400 111M/s in 8.1s
2013-07-22 18:31:18 (111 MB/s) - '/dev/null' saved [943718400/943718400]
root@clientA:~#

```

### **3. Avaliação**

A proposta do ATER é basicamente facilitar o acesso e o uso do SE-CIPÓ. Para facilitar o acesso, o ATER permite que a rede do cliente seja conectada a um equipamento, o RACE, o qual fornece conectividade tanto à rede Ipê quanto à rede experimental CIPÓ. Essa abordagem exige que a configuração seja realizada apenas uma vez pela equipe de operação da RNP. A partir de então, a decisão de por qual rede deseja escoar seu tráfego pode ficar a cargo do cliente, sob aprovação de um gestor de rede ou da própria operação da RNP.

A facilidade de uso do ATER pode ser observada na maneira como um cliente final define qual tráfego deseja enviar por circuitos dinâmicos. Basicamente, o cliente acessa uma interface Web e informa qual o perfil do tráfego, através de dados como IP de origem e de destino. Basta que um gerente local da rede ou administrador da RNP aprove sua solicitação para que a regra definida seja aplicada. Desse ponto em diante, o cliente simplesmente usa suas aplicações distribuídas de maneira convencional sem se preocupar com nenhuma ação adicional.

O ATER melhora também a percepção de robustez do SE-CIPÓ por parte dos clientes, pois sempre que a rede experimental estiver inoperante, o tráfego será automaticamente encaminhado pela rede de produção Ipê. Ou seja, o cliente final não sofre interrupções perceptíveis na conectividade caso o SE-CIPÓ se torne indisponível.

Todas essas características foram testadas em laboratório e estão prontas para testes preliminares na rede experimental CIPÓ. Um ponto crítico no ATER é a comunicação com o OSCARS, a qual é realizada através do módulo CORE OSCARS Driver. Esse módulo foi testado durante aproximadamente um mês na rede experimental CIPÓ e, após correções e melhorias, tem se mostrado plenamente operacional.

Nesse contexto, a equipe do GT avalia que o ATER demonstrou alto potencial para se tornar um serviço a ser oferecido pela RNP a seus usuários.