

## Testbed Céu na Terra

Testes realizados:

### Parte I: Teste de filtragem de portas para aplicações IPv6

- Simular conexão de um endereço IP diferente do endereço IP cadastrado na configuração do experimento.
- Alterar o endereço IP da máquina do usuário na configuração de um teste durante a execução de um experimento.
- Tentar acessar mote utilizando porta de acesso não disponível

### Parte II: Teste do mecanismo de depuração “printf” nas aplicações IPv6

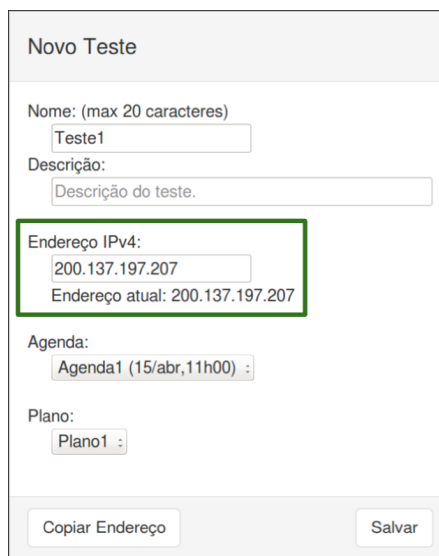
- Executar aplicação Java PrintfClient para um nó específico da RSSF e verificar saídas.

### Parte III: Acompanhar execução da aplicação através da saída no log do experimento

- Verificar se o log do experimento exibe os estados dos nós (ativo, desativado, aguardando) de acordo com um determinado script de execução

## Parte I: Testes de filtragem de portas para aplicações IPv6

A filtragem de portas tem como objetivo garantir que somente o usuário que realizou a reserva para um determinado slot de tempo no testbed tenha acesso à porta de dados durante a execução do experimento. Para isso, quando o usuário cria um novo teste, o portal web grava o endereço IP da máquina do usuário ou da máquina responsável pelo NAT (Figura 1).



Novo Teste

Nome: (max 20 caracteres)  
Teste1

Descrição:  
Descrição do teste.

Endereço IPv4:  
200.137.197.207  
Endereço atual: 200.137.197.207

Agenda:  
Agenda1 (15/abr,11h00) :

Plano:  
Plano1 :

Copiar Endereço      Salvar

*Figura 1: Procedimento de configuração de um experimento no portal web: O usuário deve informar o endereço IP da máquina que irá se comunicar com a RSSF durante a execução do experimento.*

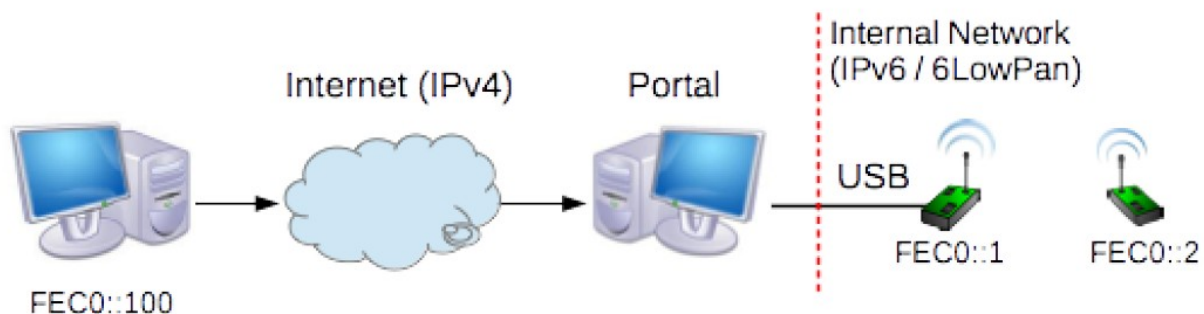
Isso garante que o portal web só permita a conexão da máquina com o endereço IP cadastrado junto com o teste. É possível permitir também que não só uma máquina, mas uma rede tenha acesso às portas de dados durante a execução de um experimento. Isso se dá através da definição da máscara de rede para um conjunto de endereços IP. Por exemplo, o endereço 200.137.197.**255** permite que qualquer máquina que inicie com 200.137.197.**xxx** acesse as portas de dados. Caso o endereço configurado seja 255.255.255.255 as portas ficarão abertas para conexão de qualquer máquina. Por outro lado, caso o campo “Endereço IPv4” seja deixado em branco, será impossível se comunicar com qualquer nó durante a execução do experimento.

### Ambiente de execução dos testes

Os testes realizados se basearam na arquitetura de uma aplicação IPv6: De um lado é executada a aplicação IPv6 na RSSF (testbed, neste caso), e do outro um usuário com uma máquina conectada à Internet (IPv4) executa uma aplicação que processa os dados advindos do experimento no testbed (Figura 2).

Os testes realizados utilizaram aplicações para a plataforma TinyOS em motes do tipo TelosB.

A comunicação entre a aplicação que executa na máquina do usuário sob uma rede IPv4 e a aplicação executando no testbed sob uma rede IPv6 é feita através do protocolo PPP (point-to-point protocol).



*Figura 2: No lado esquerdo, a máquina do usuário conectada ao testbed através da Internet. No lado direito, o testbed, utiliza um nó sink (FEC0::1) que é conectado ao gateway (Portal) através da porta USB.*

Assim que o experimento é iniciado, o usuário em um sistema Linux utiliza a ferramenta de linha de comando *ppp*, que cria uma interface de rede virtual permitindo que qualquer pacote enviado para esta interface seja repassado ao nó sink e que qualquer pacote enviado ao nó *sink* seja recebido na interface. A criação da interface entre a máquina do usuário e o nó *sink* é feita através do endereço IP do portal e a porta de dados na qual o nó sink com a aplicação *Ppp Router* (distribuído com o TinyOS) está conectado:

```
sudo pppd debug passive noauth nodetach 115200 socket 146.164.247.234:10000 \
nocrtscts nocdtrcts lcp-echo-interval 0 noccp noip ipv6 ::23,::24
```

O teste de filtragem de portas será realizado executando um experimento no testbed e executando no lado do usuário a ferramenta de linha de comando *ppp* (mostrada acima). O interfaceamento entre o testbed e a máquina do usuário só é bem sucedido caso o endereço IP da máquina do usuário seja equivalente ao endereço IP ou à máscara de rede informado no momento de cadastrar o experimento no portal.

## **1. Testes**

### **1.1. Simular conexão de um endereço IP diferente do endereço IP cadastrado na configuração do experimento.**

#### **Descrição:**

Como mostrado anteriormente na Figura 1, quando um experimento é configurado no portal web, é necessário informar o endereço IP da máquina que irá se comunicar com a RSSF durante o experimento. Este teste tem como objetivo simular a tentativa de acesso de uma máquina com um endereço IP diferente do endereço IP informado no momento de configuração do experimento. O teste segue o seguinte roteiro:

- a. O usuário cria um experimento no portal web informando um endereço IP diferente do endereço IP de sua máquina e inicia o teste.
- b. Após isso, o usuário executa em sua máquina a ferramenta de linha de comando *ppp*, para que seja possível receber os dados provenientes da RSSF.

#### **Resultado esperado:**

- A conexão é recusada, já que o endereço IP do usuário é diferente do endereço IP informado na configuração do teste.
- Endereço IP da máquina que tentou realizar conexão é exibida no log do experimento no portal web.

#### **Resultado obtido:**

1. *[Host/ Lado do Usuário]*: Quando o comando *ppp* é executado, é retornada a mensagem de erro de conexão recusada (*Connection refused*).
2. *[Portal Web/Testbed]*: O log do experimento exibe o IP da máquina do usuário que tentou realizar conexão.

### **1.2. Alterar o endereço IP da máquina do usuário na configuração de um teste durante a execução de um experimento.**

#### **Descrição:**

Este teste tem como objetivo simular o que acontece quando, durante a execução de um experimento, o usuário altera o endereço IP informado na configuração do teste para um endereço IP de uma rede diferente. Abaixo, o roteiro de teste:

- a. O usuário cria um experimento no portal web informando o endereço IP de sua máquina e inicia o teste.
- b. Após isso, o usuário executa em sua máquina a ferramenta de linha de comando *ppp*, para que seja possível receber os dados provenientes da RSSF. O usuário aguarda até que a comunicação seja estabelecida com sucesso.
- c. Durante a execução do experimento, o usuário altera o endereço IP na configuração do experimento no portal web para um endereço IP de uma rede diferente.

#### **Resultado esperado:**

Assim que ocorrer qualquer alteração no endereço IP configurado para um experimento, o experimento deverá ser reiniciado automaticamente para validar a nova configuração, ou seja, todos os motes envolvidos no teste devem ser desativados e ativados novamente para o novo endereço IP informado.

#### **Resultado obtido:**

1. *[Portal Web/Testbed]*: Os registros de log mostram o experimento sendo reiniciado (todos os motes são reiniciados) para a nova configuração de endereço IP feita.
2. *[Máquina do Usuário]*: A execução da ferramenta de linha comando *ppp* é terminada, já que o endereço IP da máquina do usuário não é mais compatível com o novo endereço IP definido para o experimento.
3. *[Portal Web/Testbed]*: Logo após a execução do comando *ppp* na máquina do usuário, o log do portal exibe o IP da máquina do usuário que tentou realizar conexão.

### **1.3. Tentar acessar mote utilizando porta de acesso não disponível**

#### **Descrição:**

Como mostrado anteriormente, a ferramenta de linha comando *ppp* permite a comunicação entre a RSSF (IPv6) com a máquina do usuário conectada à Internet (IPv4). Para utilizar essa ferramenta é necessário informar o endereço IP do portal com o número da porta de dados do mote no qual se deseja receber os dados:

```
sudo pppd debug passive noauth nodetach 115200 socket 146.164.247.234:10006 \
nocrtscts nocdtrcts lcp-echo-interval 0 noccp noip ipv6 ::23,::24
```

Usualmente, é utilizada a porta de dados do mote que possui a aplicação Ppp Router instalada, pois este mote funciona como o roteador de borda da aplicação. Quando o usuário executa a ferramenta de linha de comando *ppp*, a máquina do usuário recebe os dados de toda a RSSF que passa pelo roteador de borda (nó *sink*).

No comando acima utiliza-se o endereço IP do portal (*146.164.247.234*) com o número da porta de dados de nó com a aplicação Ppp Router instalada (*10006*).

ID Físico	ID Lógico	Tipo do mote	Porta TCP	Usar mote?	Arquivo default?	Arquivo Binário
				<input type="checkbox"/>	<input type="checkbox"/>	← marcar todos:
1	1	micaz300	10001	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
2	2	micaz300	10002	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
3	3	micaz300	10003	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
4	4	micaz300	10004	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
5	5	micaz300	10005	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
6	6	telosb	10006	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ppp_router
7	7	telosb	10007	<input checked="" type="checkbox"/>	<input type="checkbox"/>	water_monitor
8	8	telosb	10008	<input checked="" type="checkbox"/>	<input type="checkbox"/>	water_monitor

Figura 3: Configuração da topologia da RSSF a ser utilizada no experimento

A cada mote é associado um número para porta de dados. Na Figura 3, observa-se que somente os últimos três motes estão selecionados para o experimento. Assim, somente as últimas três portas TCP estarão disponíveis para acesso durante a execução do experimento.

Este teste tem como objetivo simular a tentativa de conexão entre a máquina do usuário e a RSSF através da ferramenta de linha comando *ppp* utilizando uma porta de dados não disponível para o experimento em execução. Tomando como exemplo a figura 3, temos que as portas no intervalo de 1001 até 1005 estarão indisponíveis durante o experimento.

Abaixo, o roteiro de teste:

- O usuário cria um experimento no portal web e o inicia.
- Após isso, o usuário executa em sua máquina a ferramenta de linha de comando *ppp* utilizando uma porta não disponível para o experimento.

### Resultado Esperado

Não é possível realizar conexão.

### Resultado Obtido:

Comando *ppp* não recebe resposta ao tentar se conectar. Não é possível realizar conexão.

## Parte II: Teste do mecanismo de depuração “printf” nas aplicações IPv6

Eventualmente, quando se desenvolve uma aplicação para executar em uma RSSF, é necessário utilizar mecanismos de depuração da aplicação. Como se trata de aplicações para a plataforma TinyOS, a função *printf* da linguagem *nesC* pode ser utilizada para este objetivo. Para visualizar o retorno desta função durante a execução de uma aplicação no testbed pode-se utilizar a ferramenta de linha de comando em Java PrintfClient distribuído com a plataforma TinyOS. Assim que o experimento é iniciado, o usuário deve executar o comando abaixo em um terminal, informando o endereço IP do testbed juntamente com a porta TCP do mote desejado:

```
java net.tinyos.PrintfClient -comm network@<ip_do_testbed:porta>
```

Essa sequência de testes tem como objetivo verificar o funcionamento deste mecanismo de depuração para uma aplicação executando no testbed.

## **1. Testes**

### **1.1 Executar aplicação Java PrintfClient para um nó específico da RSSF e verificar saídas.**

#### **Descrição:**

Tomando como base a topologia de rede definida na Figura 3, foi selecionado para este teste o nó de ID físico 7 ( com a aplicação IPv6 instalada). Assim que o experimento for iniciado, será executada a aplicação PrintfClient, que receberá como parâmetro o endereço IP do testbed e a porta TCP do nó em questão (porta 10007).

Abaixo, o roteiro de teste:

- a. Configurar e iniciar a execução de um experimento no testbed de acordo com a topologia de rede mostrada na Figura 3.
- b. Na máquina do usuário, executar aplicação Java PrintfClient passando como parâmetro o endereço IP do portal com a porta TCP do nó de ID 7. Para este teste, o comando executado é o seguinte:

```
java net.tinyos.PrintfClient -comm network@146.164.247.210:10007
```

#### **Resultado esperado:**

Assim que a aplicação Java PrintfClient for iniciada na máquina do usuário, será exibido o retorno da função *printf* na aplicação IPv6 no mote durante a execução do experimento.

#### **Resultados obtidos:**

[Host/ Lado do Usuário]: Aplicação PrintfClient imprime a saída do comando printf da aplicação que executa no mote.

### **Parte III: Acompanhar execução da aplicação através da saída no log do experimento**

Atualmente, o portal web do testbed Céu na Terra tem disponível o recurso de monitoramento dos eventos que acontecem quando uma aplicação é executada no testbed. Estes eventos podem ser ativação/desativação de nós, comandos de espera de execução durante um período de tempo (*wait*), conclusão de upload de código em um mote específico ou mesmo o início de um experimento. Parte destes eventos/comandos são definidos ao se configurar um script de execução para um experimento e é necessário que o usuário tenha noção de como o teste está progredindo conforme o tempo passa.

## **1. Testes**

### **1.1. Verificar se o log do experimento exibe os estados dos nós (ativo, desativado, aguardando) de acordo com um determinado script de execução**

#### **Descrição:**

Será configurado e iniciado um experimento no testbed utilizando a topologia de rede da Figura 3 mostrada anteriormente e o seguinte script de execução:

A 6, 7, 8	// Ativar os nós de IDs físicos 6, 7 e 8
W 10	// Espere 10 segundos
D 7	// Desative o nó de ID físico 7
W 15	// Espere 15 segundos
D 6, 8	// Desative os nós de IDs físicos 6 e 8
W 15	// Espere 15 segundos
A 6, 8	// Ative os nós de IDs físicos 6 e 8

#### **Resultado Esperado:**

Conforme o experimento executa o log exibe os eventos e comandos sendo executados no testbed no momento.

#### **Resultado Obtido:**

A Figura 4 abaixo exibe o registro de log para o experimento com o script acima: As linhas com retângulos vermelhos destacam os comandos definidos no script de execução acima durante a execução do experimento. Além disso, o registro de log também mostra os eventos em execução (ativação ou desativação de nós).

Parar		Reiniciar	Teste atual: teste1		Relógio do Servidor: 22/04/2015 11:38:14	
Data/Hora	Nº	Tipo	Mensagem			
2015/04/22 11:37:34	8	TEST	Activating node			
2015/04/22 11:37:34	6	TEST	Activating node			
2015/04/22 11:37:34	0	TEST	step command: A 6, 8			
2015/04/22 11:37:34	0	TEST	Received Event [Step] at state [Running]			
2015/04/22 11:37:22	8	TEST	Program un-loaded in 2676 ms.			
2015/04/22 11:37:19	0	TEST	step command: W 15			
2015/04/22 11:37:19	0	TEST	Received Event [Step] at state [Running]			
2015/04/22 11:37:19	8	TEST	Deactivating node			
2015/04/22 11:37:19	6	TEST	Deactivating node			
2015/04/22 11:37:19	0	TEST	step command: D 6, 8			
2015/04/22 11:37:19	0	TEST	Received Event [Step] at state [Running]			
2015/04/22 11:37:12	8	TEST	Program loaded in 18055 ms.			
2015/04/22 11:37:04	0	TEST	step command: W 15			
2015/04/22 11:37:04	0	TEST	Received Event [Step] at state [Running]			
2015/04/22 11:37:04	7	TEST	Deactivating node			
2015/04/22 11:37:04	0	TEST	step command: D 7			
2015/04/22 11:37:04	0	TEST	Received Event [Step] at state [Running]			
2015/04/22 11:36:54	0	TEST	step command: W 10			
2015/04/22 11:36:54	0	TEST	Received Event [Step] at state [Running]			
2015/04/22 11:36:54	8	TEST	Activating node			
2015/04/22 11:36:54	7	TEST	Activating node			
2015/04/22 11:36:54	6	TEST	Activating node			
2015/04/22 11:36:54	0	TEST	step command: A 6, 7, 8			
2015/04/22 11:36:54	0	TEST	Received Event [Step] at state [Running]			
2015/04/22 11:36:53	0	TEST	Received Event [Start] at state [Idle]			
2015/04/22 11:36:53	0	TEST	Received Event [NewConfig] at state [Idle]			

**Figura 4: Registros de log durante a execução de um experimento**  
Linhas com retângulos vermelhos indicam qual parte do script  
está sendo executada no momento.