INTRODUÇÃO AO PING E TRACEROUTE

Tatiana Lopes Ferraz
ferraz@cbpf.br

Marcelo Portes Albuquerque
marcelo@cbpf.br

Márcio Portes Albuquerque
mpa@cbpf.br

RESUMO

Esta nota técnica tem como objetivo relatar os principais aspectos relacionados com o PING e o TRACEROUTE – ferramentas importantes para os administradores de uma rede de computadores.

Para uma melhor compreensão destas ferramentas, também serão abordados tópicos tais como: IP, UDP e ICMP.

ÍNDICE

1. INTRODUÇÃO

O aplicativo PING é uma ferramenta de diagnóstico para verificar conectividade entre dois hosts em uma rede, ou seja, é um teste importante para o gerenciamento de redes de computadores. Além disso, o ping mede o tempo de atraso entre o pacote ICMP enviado e o recebido, nos dando uma idéia de como a velocidade da rede está entre o computador local e o remoto.

Este aplicativo costumava ser um ótimo indicador da habilidade de uma máquina enviar e receber pacotes IP em geral. Se você pudesse “pingar” um *host*, você também poderia estabelecer uma conexão ftp ou http com o mesmo. Com advento da filtragem de pacotes para segurança, isso não está sendo mais realidade, muitos *firewalls* desabilitam pacotes ICMP. Um dos motivos para desabilitação de pacotes ICMP são ataques baseados nesse tipo de pacote, por exemplo o “PING OF DEATH”, que usa o aplicativo ping com pacotes de grandes tamanhos para sobrecarregar as camadas IP do alvo.

O aplicativo TRACEROUTE é uma ferramenta que permite descobrir o caminho feito pelos pacotes desde a sua origem até o seu destino. Ele é usado para testes, medidas e gerenciamento da rede. Este aplicativo pode ser utilizado para detectar falhas como, por exemplo, *gateways* intermediários que descartam pacotes ou rotas que excedem a capacidade de um datagrama IP. Com esta ferramenta, o atraso da "viagem" do pacote entre a origem e *gateways* intermediários são reportados, permitindo determinar a contribuição de cada *gateway* para o atraso total da "viagem" do pacote desde a origem até o seu destino.

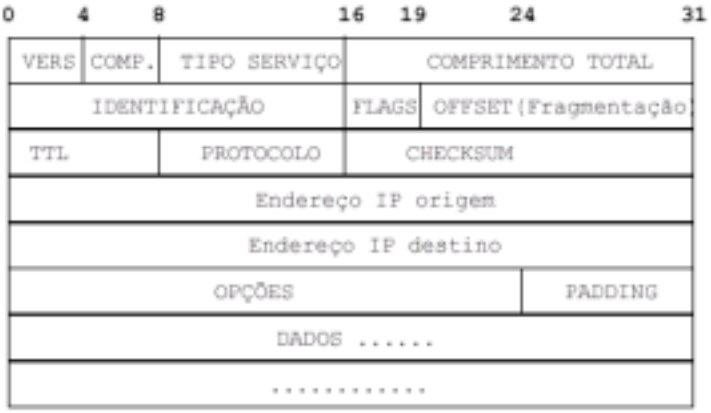
Será apresentado uma pequena noção do IP, UDP e ICMP, protocolos necessários para uma melhor compreensão desses aplicativos.

2. IP (*Internet Protocol*)

O protocolo IP é o protocolo básico da Internet, sendo responsável pela identificação das máquinas e redes e encaminhamento correto das mensagens entre elas. O serviço é definido como um sistema de transmissão sem conexão e não confiável. Não garante se os pacotes de informações serão recebidos, se a seqüência dos pacotes enviados de um computador a outro trafegam no mesmo caminho e que todos pacotes chegarão.

Logo protocolo IP não verifica se um pacote alcançou o seu destino, além de não executar nenhuma ação de correção, caso ele não tenha alcançado

O IP inclui, como parte integrante, o ICMP - protocolo de mensagem de erro. Abaixo é apresentado o datagrama IP.



cabeçalho IP

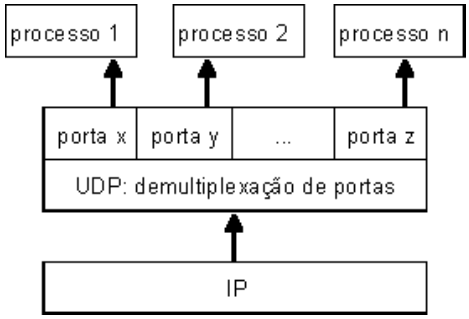
Um dos campos mais importantes para o estudo de ping e traceroute é o *TTL* (Time To Live) que define o número máximo de roteadores pelos quais o pacote pode trafegar. Isso evita, por exemplo, que um pacote fique indefinidamente circulando pela rede antes de achar o seu endereço destino. Normalmente as aplicações definem um TTL de valor 60, que é mais do que suficiente para os pacotes chegarem ao destino. Em cada roteador por onde passam esses pacotes o valor do TTL é decrementado em uma unidade. Se chegar a zero, sem encontrar o destino, uma mensagem ICMP é enviada ao computador de origem e o pacote é completamente descartado, impedindo a criação de loops e assim garantindo a estabilidade ao processo de roteamento.

3. UDP (*User Datagram Protocol*)

O protocolo UDP fornece uma forma simples de acesso ao sistema de comunicação, provendo um serviço sem conexão, sem confiabilidade e sem correção de erros. O usuário pode enviar uma mensagem na rede sem estabelecer uma conexão com o receptor, isto é, o usuário simplesmente coloca mensagem na rede com o endereço destino e espera que essa chegue.

O protocolo UDP é basicamente uma interface de aplicação para o protocolo IP. Este protocolo não adiciona ao IP qualquer confiabilidade, controle de fluxo ou recuperação de erros, mas simplesmente serve como um multiplexador/demultiplexador para o envio e recepção de datagramas IP, utilizando portas para direcioná-los. Enquanto trata a chegada de mensagens o serviço UDP recebe os datagramas do serviço IP e o demultiplexa baseado na porta de destino UDP, conforme figura a seguir.

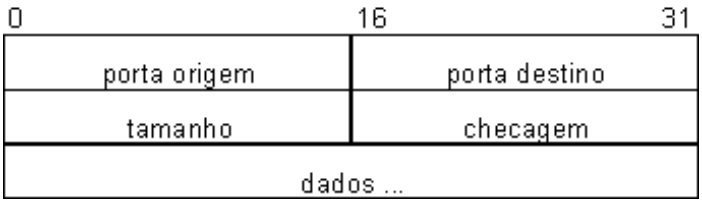
Quando o serviço UDP recebe um datagrama ele verifica se a porta de destino coincide com alguma porta existente. Caso isto não ocorra, uma notificação ICMP de porta não acessível é transmitida e o datagrama é descartado. Se a porta destino coincide com alguma porta em uso o datagrama é deixado no *buffer* da porta. Caso o *buffer* já esteja cheio o datagrama recebido é descartado.



UDP, demultiplexação baseada nas portas

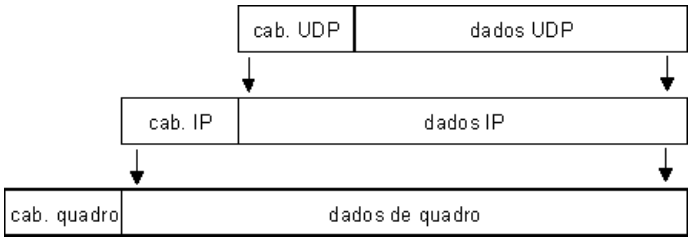
Conceitualmente a multiplexação e demultiplexação entre o serviço UDP e as aplicações ocorre através do mecanismo de portas. Na prática cada aplicação precisa negociar com o sistema operacional para obter uma porta no protocolo e o número de porta associado antes de enviar um datagrama UDP. Uma vez obtida a porta todos os datagramas enviados através da porta terão este número no campo porta de origem.

Abaixo é apresentado o formato do datagrama UDP.



Datagrama UDP

O protocolo UDP (camada de transporte) está uma camada acima do protocolo IP (camada de rede). Isto significa que um segmento completo UDP, incluindo o cabeçalho e dados, é encapsulado ao datagrama IP enquanto trafega pela rede de comunicação, conforme a figura a seguir.



Segmento UDP encapsulado em um datagrama IP

Deve-se observar que os protocolos UDP e IP não fornecem qualquer confiabilidade, desta forma é tarefa da aplicação prover controle de fluxo e correção de erros.

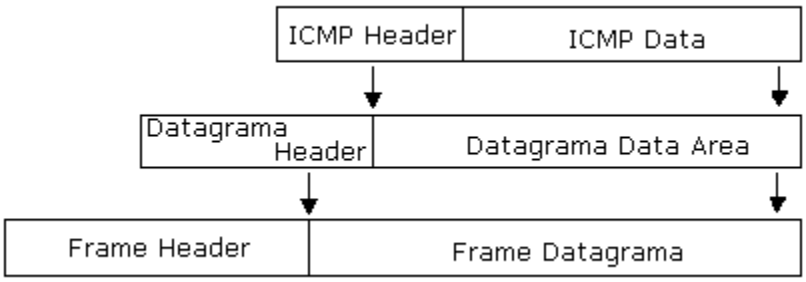
4. ICMP (*Internet Control Message Protocol*)

O ICMP é similar ao UDP pois utiliza mensagens que cabem num só datagrama, sendo no entanto ainda mais simples uma vez que não possui a indicação, no seu cabeçalho, das portas.

O protocolo de mensagem de controle da Internet (ICMP), é obrigatório em implementações da camada IP. Na sua maioria indicam a ocorrência de problemas no transporte de algum datagrama ou servem para operações de controle. Esses problemas podem ser causados, quando a máquina de destino não está conectada na rede, quando o

campo do datagrama TTL expira ou quando os roteadores ficam muito congestionados, não conseguindo processar o tráfego de entrada. Desta forma o ICMP envia mensagens de erro ou controle para os hosts, que fizeram a requisição.

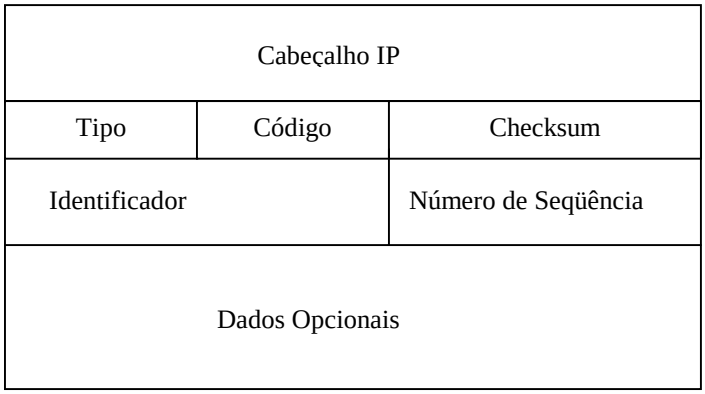
O ICMP utiliza o IP para o transporte de mensagem, não oferecendo, portanto, garantia de entrega. A figura a seguir apresenta como uma mensagem ICMP é encapsulada em um datagrama IP.



Quando ocorre algum problema previsto pelo ICMP, uma mensagem ICMP descrevendo a situação é preparada e entregue à camada IP, que adiciona esta ao seu cabeçalho e envia ao emissor do datagrama com o qual ocorreu o problema.

É importante saber que o ICMP é um mecanismo de aviso de erros e não especifica a ação para correção do erro. O computador de origem é quem deve relatar o erro a um programa de aplicação para correção deste problema.

O formato geral de uma mensagem ICMP é apresentado na figura abaixo. O campo TIPO identifica a mensagem ICMP particular, o campo CÓDIGO é usado na especificação dos parâmetros da mensagem e o campo CHECKSUM corresponde ao código verificador de erro, calculado a partir da mensagem ICMP completa.

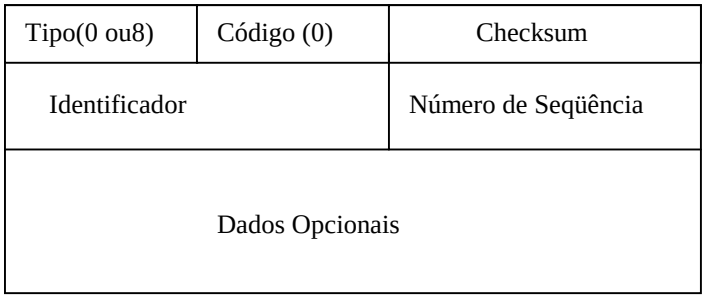


Pacote ICMP

6. PING

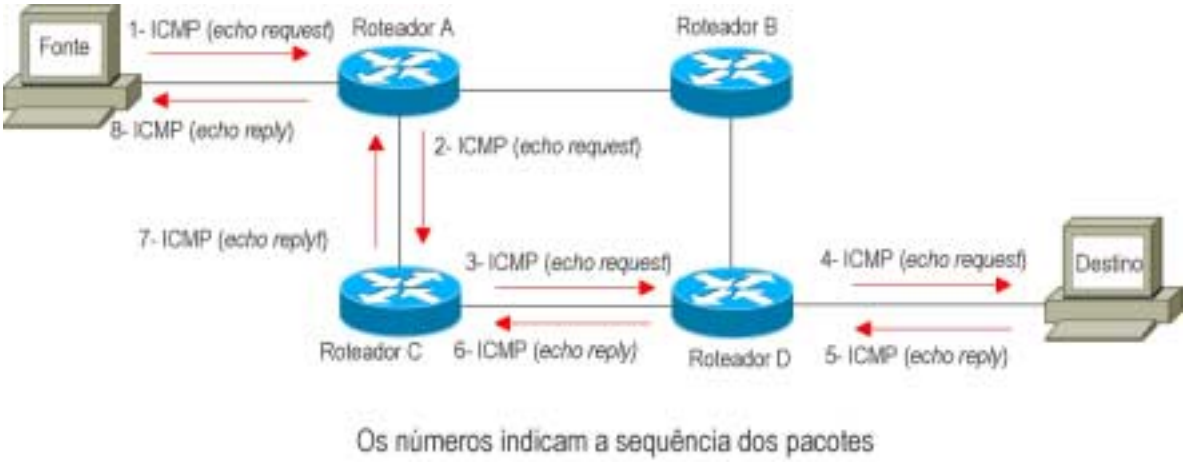
O ping é um aplicativo que utiliza o protocolo ICMP e permite ao usuário verificar a conectividade entre dois hosts.

Este aplicativo envia pacotes ICMP (*echo request*) para uma determinada máquina e aguarda uma mensagem ICMP de resposta (*echo reply*). O formato da mensagem ICMP, utilizada no ping, é apresentado na figura a seguir.



Mensagem do ping

Um pequeno exemplo demostrando o caminho dos pacotes ICMP enviados pelo aplicativo ping é apresentado, na figura a seguir.



Quando o destino recebe a mensagem ICMP de *echo request* da fonte, ele retorna com uma mensagem ICMP *echo reply*, não modificando os campos: identificador, número de seqüência e dados opcionais.

O campo de dados opcionais é usado para armazenar o momento que a mensagem ICMP de *echo request* foi enviada. Quando a fonte receber a mensagem de retorno (*echo reply*), esta pode determinar o tempo necessário para o pacote ir e voltar do seu destino. Este tempo é conhecido como RTT (*Round Trip Time*).

O RTT deve ser usado como comparação, pois o comando ping não possui prioridade, ou seja, se tiver outra tarefa a ser realizada, esta será feita antes do ping.

Na resposta do ping são mostrados 3 tempos, que corresponde ao mínimo, médio e máximo do RTT, o tamanho do pacote e TTL. Grandes diferenças nos valores RTT indicam rede congestionada ou um problema nela. O ping utiliza pacotes pequenos, pois o tamanho do pacote influência no valor do RTT.

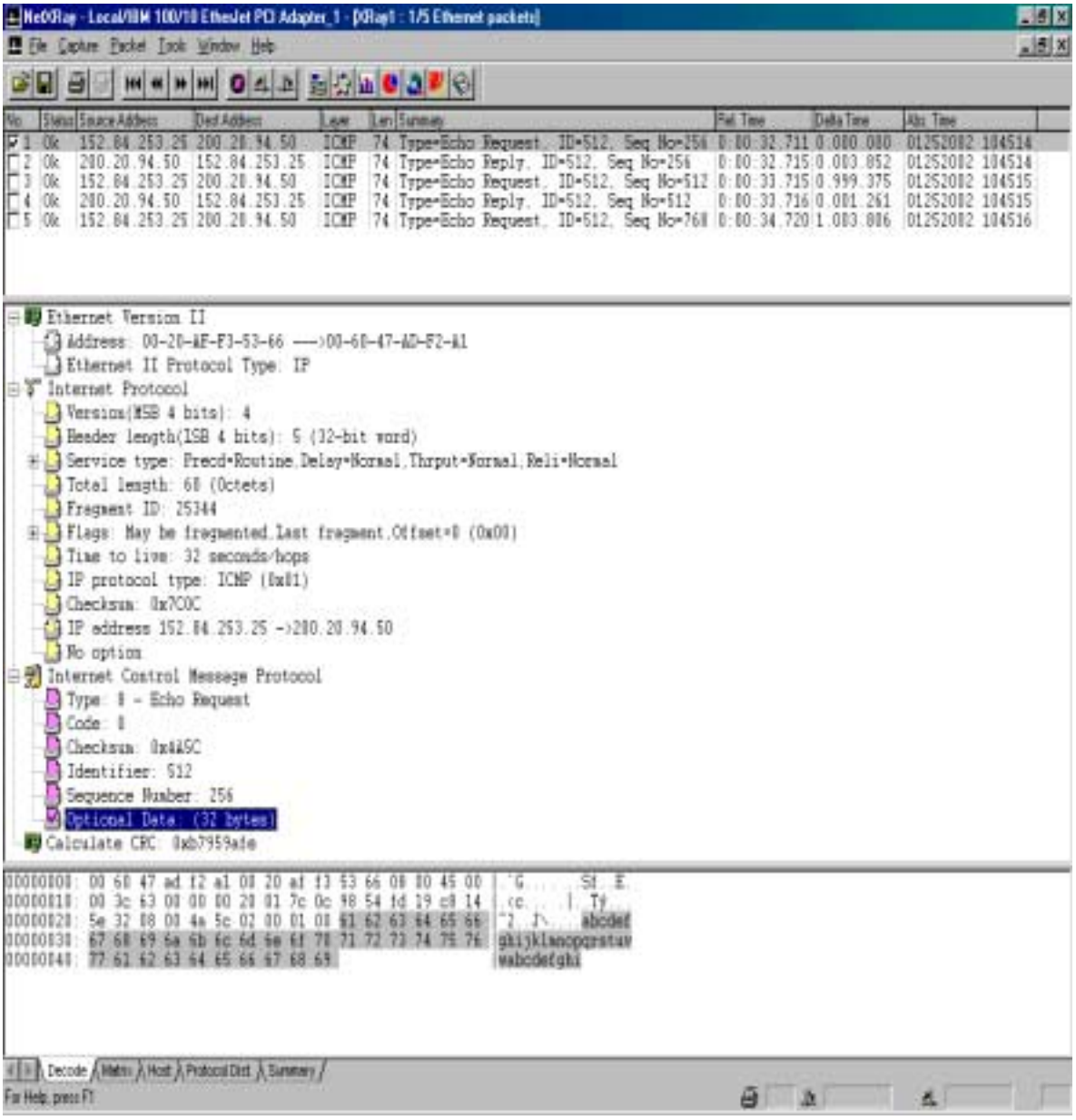
O campo TTL é preenchido com seu valor máximo de 255 e a cada roteador que o pacote passar é diminuído 1 deste valor. Logo o valor mostrado corresponde a 255 nemos número de roteadores que o pacote passou.

A seguir será apresentado um exemplo da execução do aplicativo ping, realizado numa máquina, onde o sistema operacional é o windows.

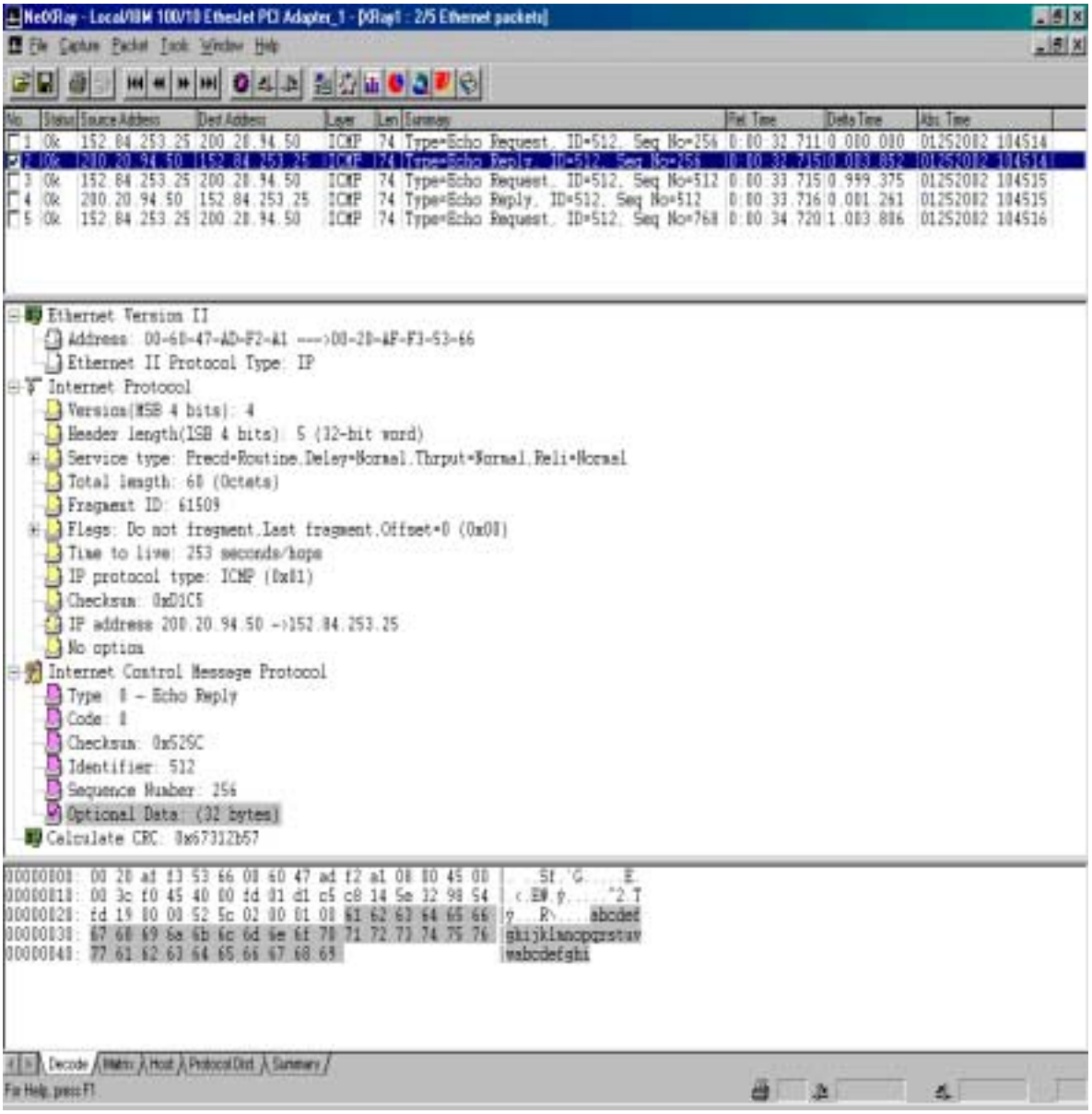
```
Microsoft(R) Windows 98
  (C)Copyright Microsoft Corp 1981-1998.

C:\WINDOWS>ping 200.20.94.50

Pinging 200.20.94.50 with 32 bytes of data:

Reply from 200.20.94.50: bytes=32 time=5ms TTL=253
Reply from 200.20.94.50: bytes=32 time=2ms TTL=253
Reply from 200.20.94.50: bytes=32 time=2ms TTL=253
Reply from 200.20.94.50: bytes=32 time=2ms TTL=253

Ping statistics for 200.20.94.50:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 2ms, Maximum =  5ms, Average =  2ms
```

Observou-se que 4 pacotes ICMP com 32 bytes foram enviados para 200.20.94.50. O primeiro pacote demorou 5ms para ir e voltar ao remetente, o segundo, o terceiro e o quarto 2ms. O número de roteadores que o pacote passou foram 255-253=2 roteadores. Pela estatísticas percebemos que 4 pacotes foram enviados e nenhum foi perdido e que o tempo mínimo foi de 2ms, máximo de 5ms e médio de 2ms.

Utilizando o programa NetXray, o qual captura pacotes, podemos observar as características do pacote enviado e recebido. A seguir são mostrados os pacotes capturados, quando foi realizado o ping acima.



Nesta situação observou-se o primeiro pacote que foi enviado de 152.84.253.25 para 200.20.94.50. Percebe-se que o pacote capturado é ICMP do tipo *echo request* (tipo=8, código=0) e que contém 32 bytes.



Já nesta situação observou-se o primeiro pacote que foi enviado de 200.20.94.50 para 152.84.253.25. Percebe-se que o pacote capturado é ICMP do tipo *echo reply* (tipo=0 código=0), ou seja a resposta do pacote anterior, e que contém 32 bytes.

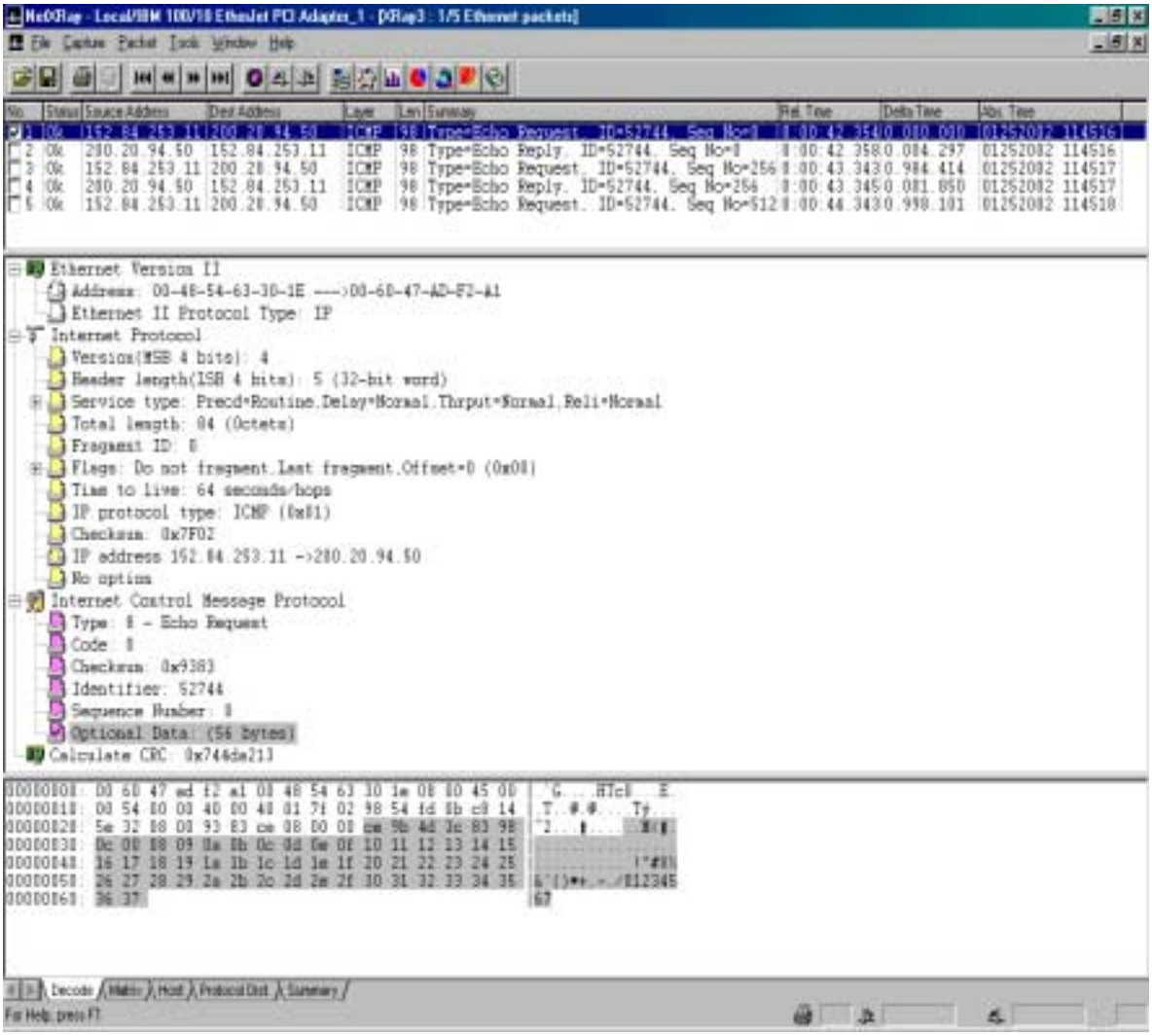
Quando foi capturado segundo pacote de *echo request*, verificou-se que não havia diferença dele para o primeiro pacote de *echo request*. Concluiu-se que o tempo de ida e volta não é mostrado pelos pacotes capturados, já que a área de dados não é modificada entre pacotes de mesmo tipo.

Agora será mostrado um exemplo de um ping executado por uma máquina em que o sistema operacional é o linux.

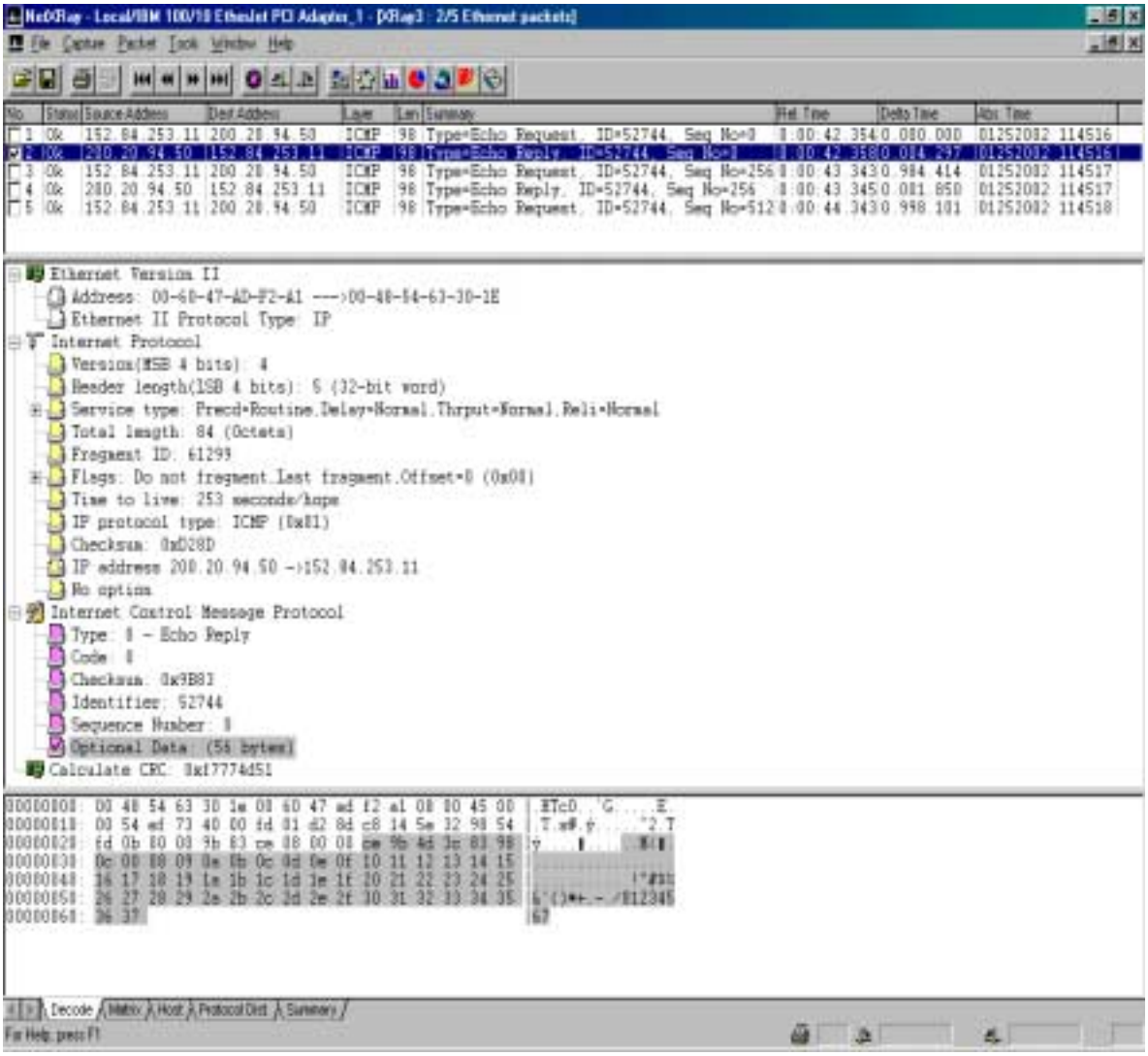
```
PING 200.20.94.50 (200.20.94.50) from 152.84.253.11 : 56(84) bytes of
data.
64 bytes from 200.20.94.50: icmp_seq=0 ttl=253 time=5.787 msec
64 bytes from 200.20.94.50: icmp_seq=1 ttl=253 time=7.145 msec
64 bytes from 200.20.94.50: icmp_seq=2 ttl=253 time=1.791 msec
64 bytes from 200.20.94.50: icmp_seq=3 ttl=253 time=2.679 msec
64 bytes from 200.20.94.50: icmp_seq=4 ttl=253 time=2.514 msec
64 bytes from 200.20.94.50: icmp_seq=5 ttl=253 time=2.939 msec
64 bytes from 200.20.94.50: icmp_seq=6 ttl=253 time=2.834 msec
64 bytes from 200.20.94.50: icmp_seq=7 ttl=253 time=2.825 msec
64 bytes from 200.20.94.50: icmp_seq=8 ttl=253 time=3.112 msec
64 bytes from 200.20.94.50: icmp_seq=9 ttl=253 time=3.464 msec
64 bytes from 200.20.94.50: icmp_seq=10 ttl=253 time=6.550 msec
64 bytes from 200.20.94.50: icmp_seq=11 ttl=253 time=2.325 msec
64 bytes from 200.20.94.50: icmp_seq=12 ttl=253 time=3.859 msec
64 bytes from 200.20.94.50: icmp_seq=13 ttl=253 time=3.463 msec
64 bytes from 200.20.94.50: icmp_seq=14 ttl=253 time=3.819 msec
64 bytes from 200.20.94.50: icmp_seq=15 ttl=253 time=3.972 msec
64 bytes from 200.20.94.50: icmp_seq=16 ttl=253 time=3.729 msec
64 bytes from 200.20.94.50: icmp_seq=17 ttl=253 time=3.462 msec
64 bytes from 200.20.94.50: icmp_seq=18 ttl=253 time=1.733 msec
64 bytes from 200.20.94.50: icmp_seq=19 ttl=253 time=3.840 msec
64 bytes from 200.20.94.50: icmp_seq=20 ttl=253 time=1.983 msec
64 bytes from 200.20.94.50: icmp_seq=21 ttl=253 time=2.260 msec
64 bytes from 200.20.94.50: icmp_seq=22 ttl=253 time=3.367 msec
64 bytes from 200.20.94.50: icmp_seq=23 ttl=253 time=1.797 msec

--- 200.20.94.50 ping statistics ---
24 packets transmitted, 24 packets received, 0% packet loss
round-trip min/avg/max/mdev = 1.733/3.385/7.145/1.375 ms
```

Observou-se que 24 pacotes ICMP com 64 bytes foram enviados para 200.20.94.50. O número de roteadores que o pacote passou foram 255-253=2 roteadores. Pela estatísticas percebe-se que 24 pacotes foram enviados e nenhum foi perdido e que o tempo mínimo foi de 1.733ms, máximo de 7.145ms e médio de 3.385ms.

A seguir são mostrados os pacotes capturados, quando foi realizado o ping acima.



Nesta situação observou-se o primeiro pacote que foi enviado de 152.84.253.11 para 200.20.94.50. Percebe-se que o pacote capturado é ICMP do tipo *echo request* (tipo=8, código=0) e que contém 64 bytes.



Já nesta situação observou-se o primeiro pacote que foi enviado de 200.20.94.50 para 152.84.253.11. Percebe-se que o pacote capturado é ICMP do tipo *echo reply* (tipo=0 código=0), ou seja a resposta do pacote anterior, e que contém 64 bytes.

Quando foi capturado o segundo pacote de *echo request*, verificou-se que a única diferença dele para o primeiro pacote de *echo request* , era a área de dados, que apresentava parâmetros diferentes. No linux, o ping usa os primeiros 8 bytes da área de dados para incluir o horário que será usado no cálculo do tempo de ida e volta do pacote. Na área de dados é enviada a hora em que o pacote saiu da sua máquina (no *echo request*). Como a área de dados deve retornar sem ser alterada no echo reply, o sistema subtrai da hora atual o

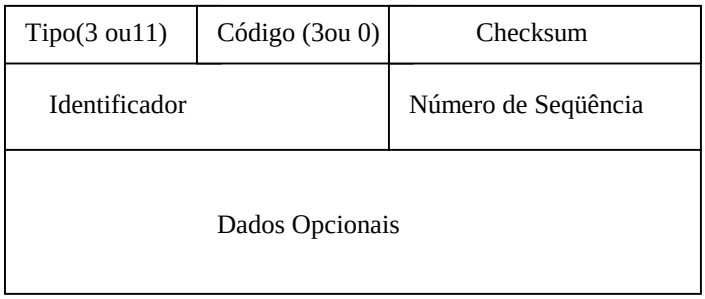
valor registrado na área dos dados e calcula o RTT. Isto foi verificado pelos pacotes capturados, já que a área de dados modificava a cada *echo request*.

7. TRACEROUTE

Com o aplicativo traceroute o usuário pode descobrir o caminho percorrido pelo pacote até seu destino.

O aplicativo traceroute envia 3 pacotes UDP com a porta de destino não usada por nenhum aplicativo, inicialmente com TTL igual a um (1). Quando passar pelo primeiro roteador, tornar-se-á zero e uma mensagem ICMP tempo excedido retornará. Com isso teremos informações sobre o primeiro roteador no meio do caminho e o RTT da fonte até este roteador. Em seguida, o TTL é aumentado para dois (2) e novamente são enviados 3 pacotes UDP, porém a mensagem ICMP ocorrerá somente no segundo roteador. O processo se repete até que tenhamos conhecimento de cada roteador no meio do caminho, entre a nossa máquina e o destino. Quando alcançamos o destino não retornará mais a mensagem ICMP tempo excedido e sim uma mensagem ICMP porta inacessível. O comando traceroute usa DNS reverso, para descobrir o endereço lógico através do número IP.

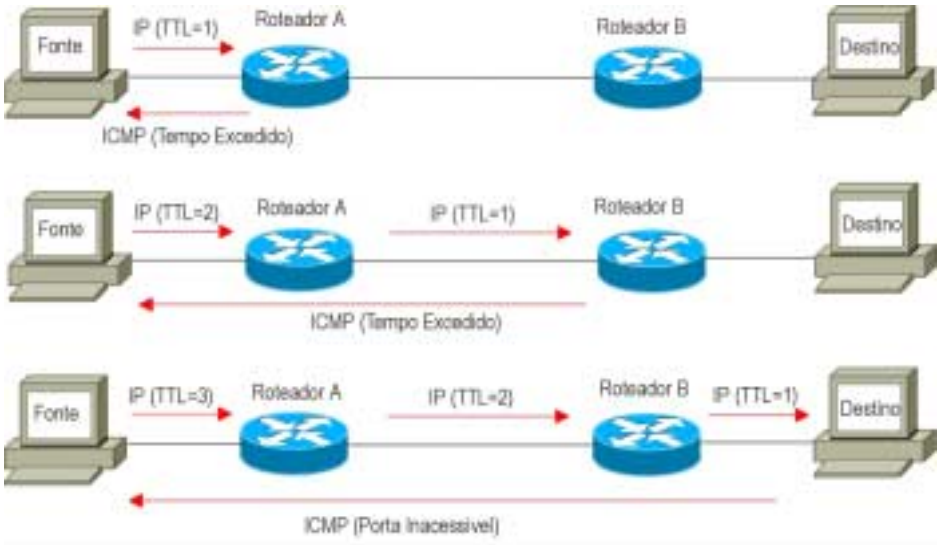
O formato da mensagem ICMP, utilizada no traceroute, é apresentado na figura a seguir.



Mensagem do traceroute

Tipo=11, código=0: ICMP tempo excedido
Tipo=3, código=3: ICMP porta inacessível

Um pequeno exemplo demostrando cada fase na operação do traceroute é apresentado, na figura a seguir.



No exemplo acima é verificado que o aplicativo traceroute retorna primeiramente com o endereço do roteador A, com o RTT entre a fonte e este roteador e com uma mensagem ICMP de tempo excedido. Em seguida retorna com o endereço do roteador B, com o RTT entre a fonte e B e com a mesma mensagem anterior. Finalizando retorna com o endereço do destino, com o RTT entre a fonte e o destino e com a mensagem ICMP de porta inacessível.

Se as respostas ao pacote vêm de gateways diferentes, o traceroute imprime o endereço IP de cada um deles. Se não houver resposta dentro de um intervalo de *time-out* de três segundos, um * (asterisco) é impresso. Um ponto de exclamação é impresso depois do RTT do pacote se o valor Max_ttl é um *hop* (roteador) ou menos.

Se a maioria dos *probes* resultam em um erro, o comando traceroute encerra a execução.

A seguir será apresentado um exemplo da execução do aplicativo traceroute, realizado numa máquina, onde o sistema operacional é:

- Windows

```
C:\WINDOWS>tracert 200.20.94.50
Tracing route to guanabara.rederio.br [200.20.94.50]
over a maximum of 30 hops:

  1    2 ms    1 ms    2 ms  cisco7513-CBPF.cat.cbpf.br [152.84.253.1]
  2    3 ms    2 ms    3 ms  rederio-atm-cbpf.rederio.br [200.20.94.41]
  3    3 ms    2 ms    3 ms  guanabara.rederio.br [200.20.94.50]

Trace complete.
```

Descrição:

30 hops: Máximo TTL

1: indica o número de quantos roteadores o pacote já passou

2ms 1ms 2ms: RTT de cada pacote enviado

cisco7513-cbpf.cat.cbpf.br: Nome do roteador

152.84.253.1: Endereço IP

- Linux

```
[root@sodium root]# traceroute 200.20.94.50
traceroute 200.20.94.50, 30hops max, 38 bytes packets
1 cisco7513-bpf (152.84.253.1)      9.232ms      2.891ms      8.132ms
2 rederio-atm-cbpf.rederio.br(200.20.94.41)      9.321ms    3.679ms      1.677ms
3 guanabara.rederio.br (200.20.94.50)    2.421ms    *      1.622ms
```

Descrição:

30 hops: Máximo TTL

38 bytes packets: Tamanho do pacote

1: indica o número de quantos roteadores o pacote já passou

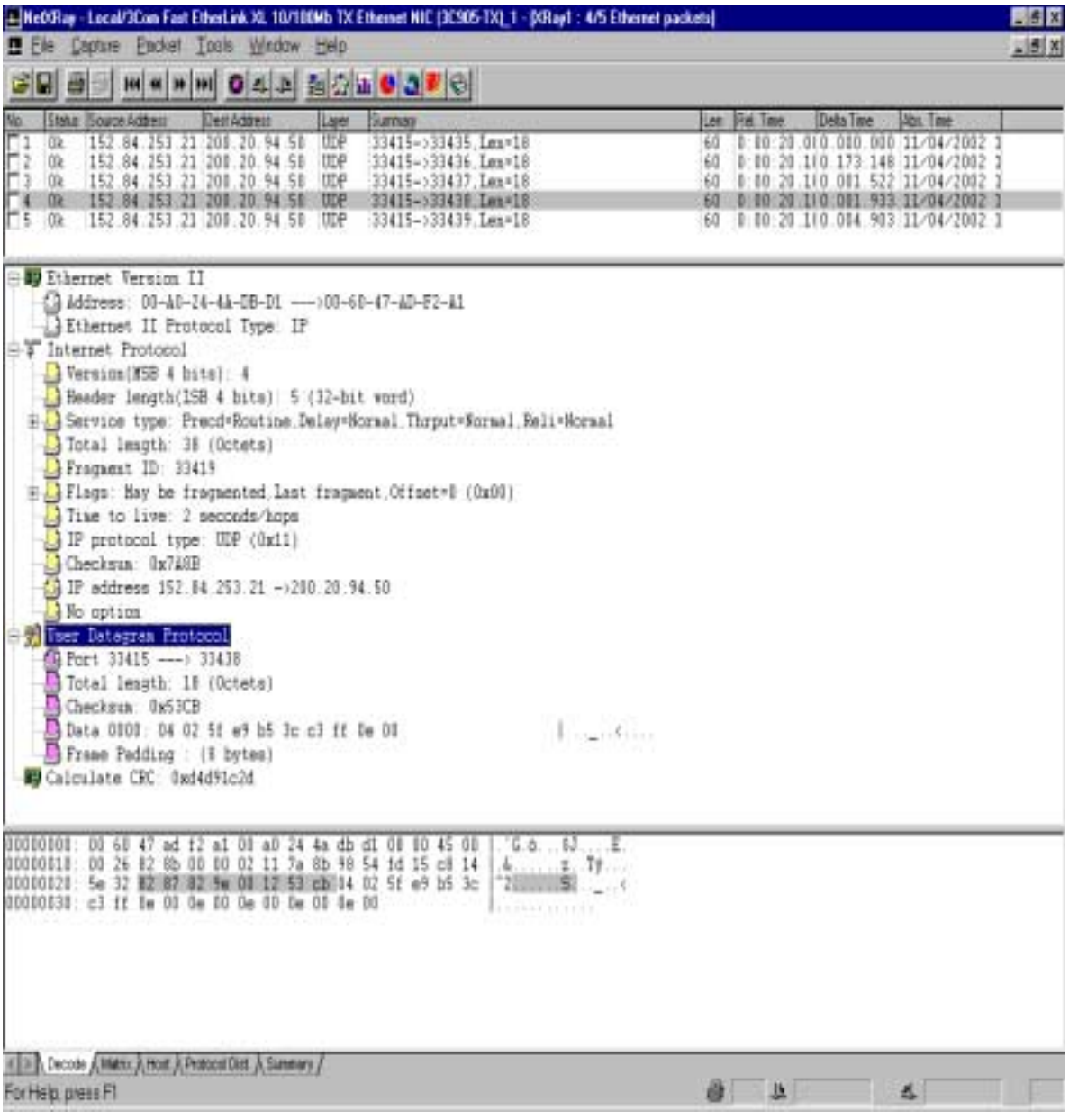
cisco7513-cbpf.cat.cbpf.br: Nome do roteador

152.84.253.1: Endereço IP

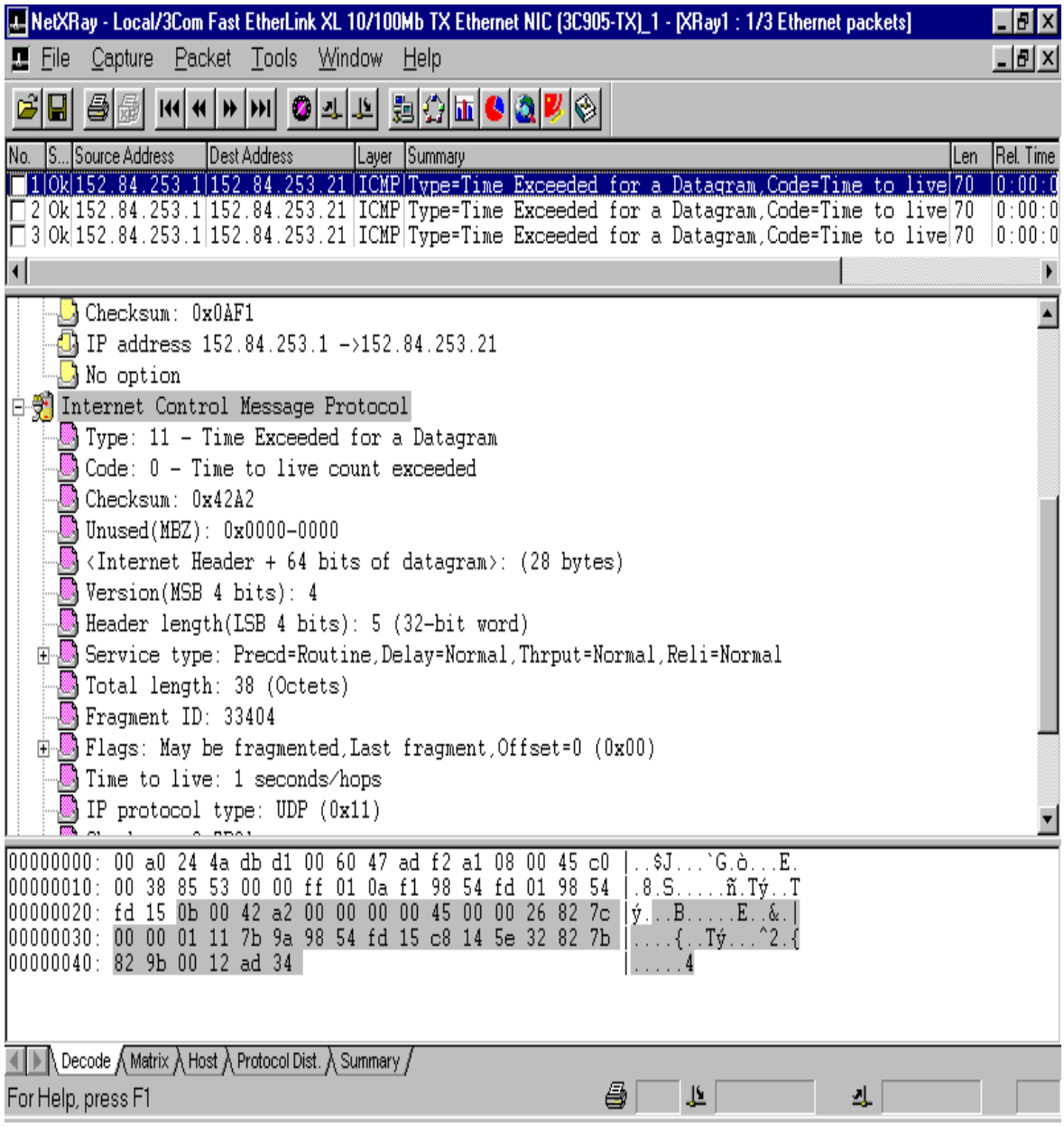
9.232ms 2.891ms 8.132ms: RTT de cada pacote enviado

*: Indica que o tempo de “timeout” expirou antes que a mensagem ICMP fosse recebida pelo datagrama.

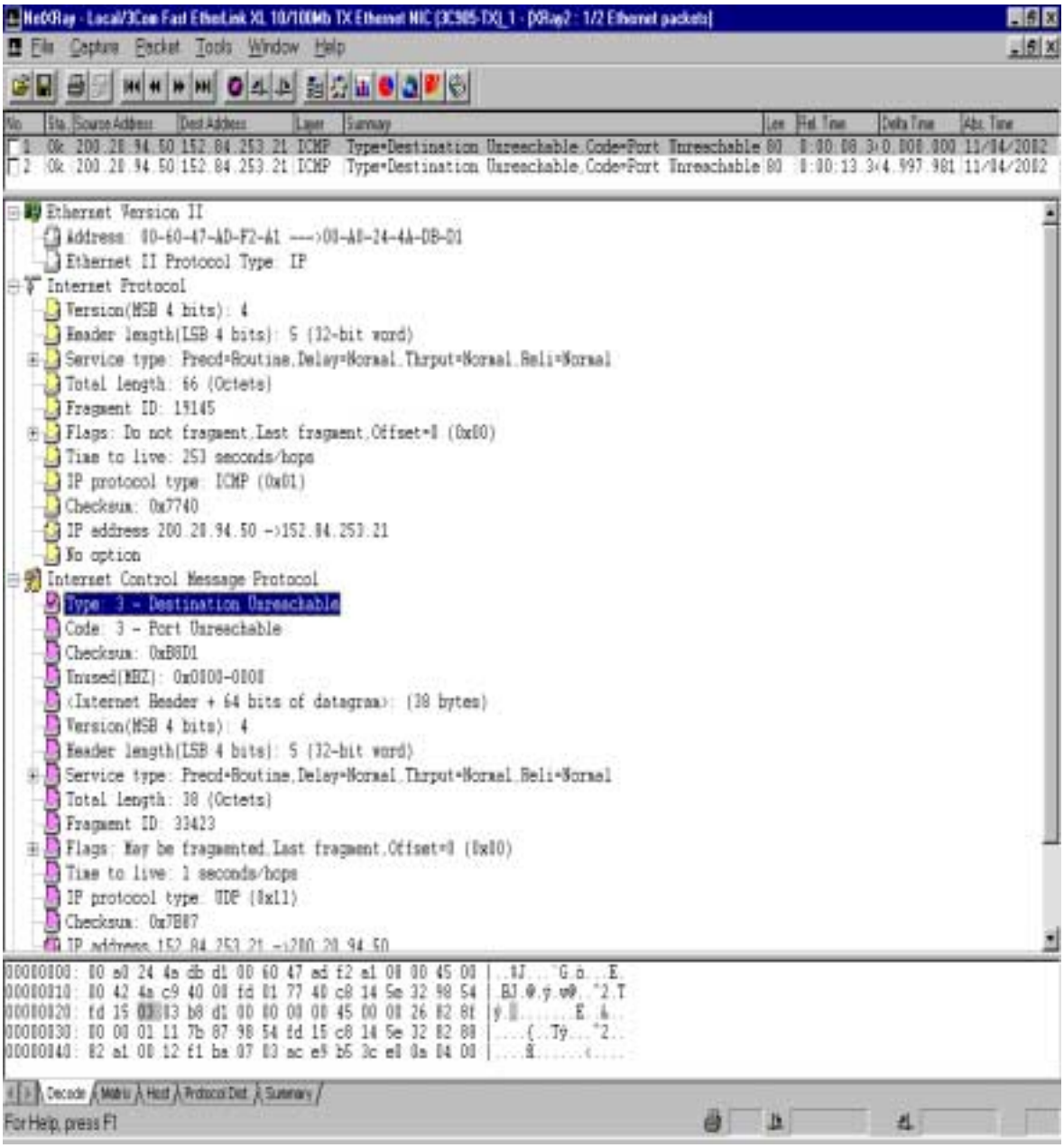
Através do NetXray capturou-se os pacotes enviados pelo traceroute acima. Primeiramente foi capturado somente os pacotes que foram enviados pela fonte. Como pode ser observado, na figura abaixo, todos eram pacotes UDP com um número da porta inexistente.



Posteriormente foram capturados os pacotes oriundos do primeiro roteador no meio do caminho (152.84.253.1) e percebeu-se que os pacotes capturados eram ICMP do tipo tempo excedido (tipo=11 , código=0), como o apresentado na figura abaixo. O mesmo tipo de pacote foi capturado, quando os pacotes eram oriundos do segundo roteador no meio do caminho (200.20.94.41).



Finalmente, quando capturou-se os pacotes oriundos do destino (200.20.94.50), verificou-se que os pacotes eram ICMP do tipo porta inacessível (tipo=3, código=3), como apresentado na figura a seguir. Percebe-se que o programa só capturou 2 pacotes, isto porque quando foi realizado esse traceroute, um pacote foi perdido.



8. CONCLUSÃO

Neste trabalho verificou-se a importância dos aplicativos PING e TRACEROUTE como uma ferramenta de diagnóstico de redes de computadores.

Concluímos que o ping nos diz se determinada máquina pode ou não ser alcançada, mas em caso de erro, não permite determinar onde a comunicação falhou. Esta informação pode ser determinada pelo traceroute.

O traceroute pode nos dizer em que ponto da rede o problema ocorreu (um roteador parado, uma subrede desligada, etc) e qual é o caminho dos pacotes. Percebemos também que quando a conexão, para determinado endereço está mais lenta que o normal, uma das possibilidades é que os pacotes estejam utilizando uma rota diferente.

Outra diferença entre o ping e o traceroute é que o primeiro garante que as mensagens ICMP estão chegando ao destino. O segundo garante que as mensagens UDP estão sendo bem sucedidas.

9. REFERÊNCIAS

- http://penta.ufrgs.br/uel/graziela/graznw13.htm
- www.imasters.com.br/web/canais/linux/artigos/ping.asp
- http://www.marcusfmugf.hpg.ig.com.Br/ferramentas.htm
- www.del.ufrj.br/~edulima/magma
- http://proenca.uel.br/curso-redes-graduacao/1998/trab-08/equipe-01/icmp.html
- http://asc.di.fct.unl.pt/rc/info/aulas/teoricas/cap4.2.pdf
- http://www.i2.com.br/~rora/aulas/redes00a/aula2/
- http://www.geocities.com/SiliconVallery/Lab/3580/avan-2.html
- http://www.dicas-l.unicamp.br/Treinamentos/tcpip/55.html

- www.dpi.ufv.br/~goulart/redesI/aula-lab-1.html
- www.icmi.ufsc.br/redes/redes98/fabiob/top04/#_toc437249157
- http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_5_5/cmd_refr/sl_w_cmd.htm#xtoicd159367
- http://www.cisco.com/univercd/cc/td/doc/product/lan/c3550/1214ea1/3550scg/swtrbl.htm#xtoicd1896113
- http://www.cisco.com/warp/public/105/traceroute.shtml
- http://www.cisco.com/univercd/cc/td/doc/product/atm/c8540/12_0/13_19/trouble/tools.htm
- http://candle.ctit.utwente.nl/wp5/tel-sys/exercises/ping/ping.html
- http://candle.ctit.utwente.nl/wp5/tel-sys/exercises/traceroute/traceroute.html
- *UNIX Network Programming, Volume 1, Second Edition: Networking APIs: Sockets and XTI*, Prentice Hall, 1998, ISBN 0-13-490012-X.
- CBPF-NT-008/98