

*rarpd*による IPアドレス設定



IPアドレスが初期設定されていない
RTX1000にrarpの仕組みを使って暫定IPア
ドレスを割り当てtelnetで設定可能とする。



RTX1000



RT57i

IPアドレス初期設定	機種例
設定なし	RTX1000,RTX2000 RT100i,RT102i,RT103i,RT105シリーズ(i/e/p), RT140シリーズ(i/e/f/p),RT200i,RT300i
192.168.0.1/24	RT80i,RTA50i,RTA52i,RT60w,RTA54i,RTW65b,RTW65i, RTA55i,RT56v
192.168.100.1/24	RT57i,RTV700

RTX1000初期設定のアクセス方法



[IPアドレスが未設定時の初期設定方法]

- 1) シリアルポート
- 2) RARP
- 3) リモートセットアップ

ISDN回線

ISDN回線越しでヤマハルーターからの
“リモートセットアップ”



設定用シリアル・ポート

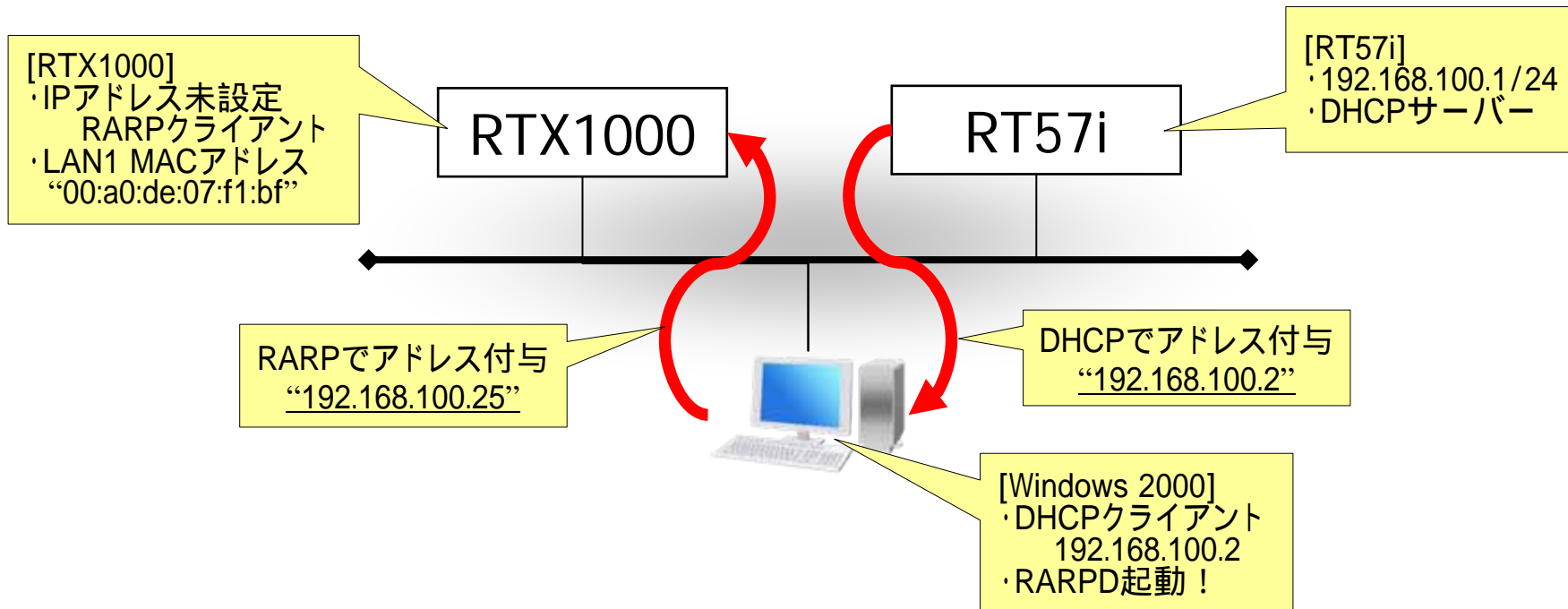
RARPで
初期IPアドレスを付与後、
“telnet”

PCでRARPDを起動

[シリアルのパラメータ]

- ・9600 bit/s
- ・文字長: 8bits
- ・パリティ: なし
- ・ストップビット: 1
- ・漢字: Shift JIS

説明に使用する環境



rarpdをダウンロードする。

rarpdのフリーソフトウェアが公開されている。 [rt100i-users 32899]

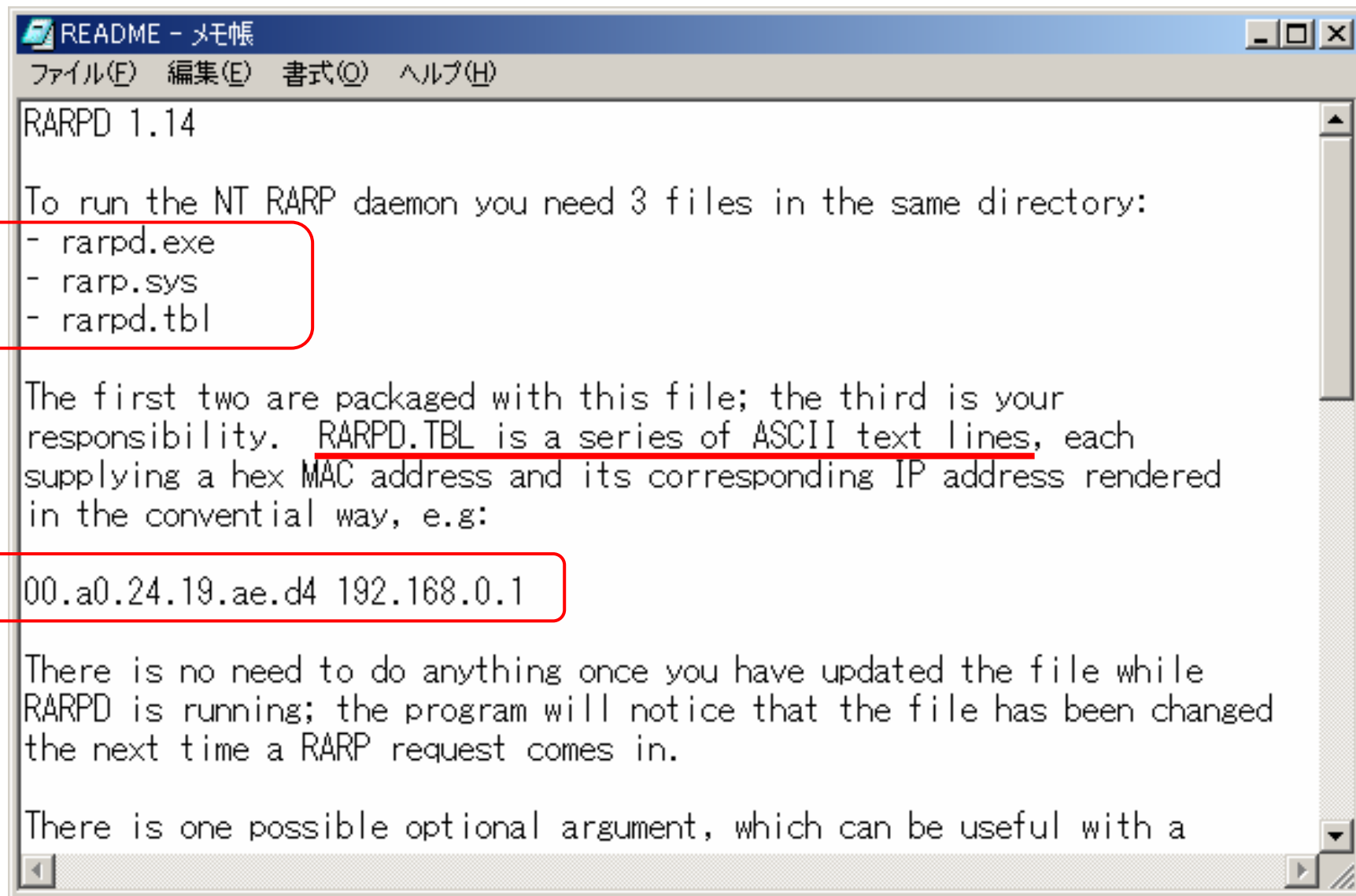
作者: Lew Perin

HomePage: <http://www.panix.com/~perin/>

File: <http://www.panix.com/~perin/rarpd.zip>



zipを解凍すると
ソースファイル
と実行ファイル
がある。



```
README - メモ帳
ファイル(F) 編集(E) 書式(O) ヘルプ(H)

RARPD 1.14

To run the NT RARP daemon you need 3 files in the same directory:
- rarpd.exe
- rarp.sys
- rarpd.tbl

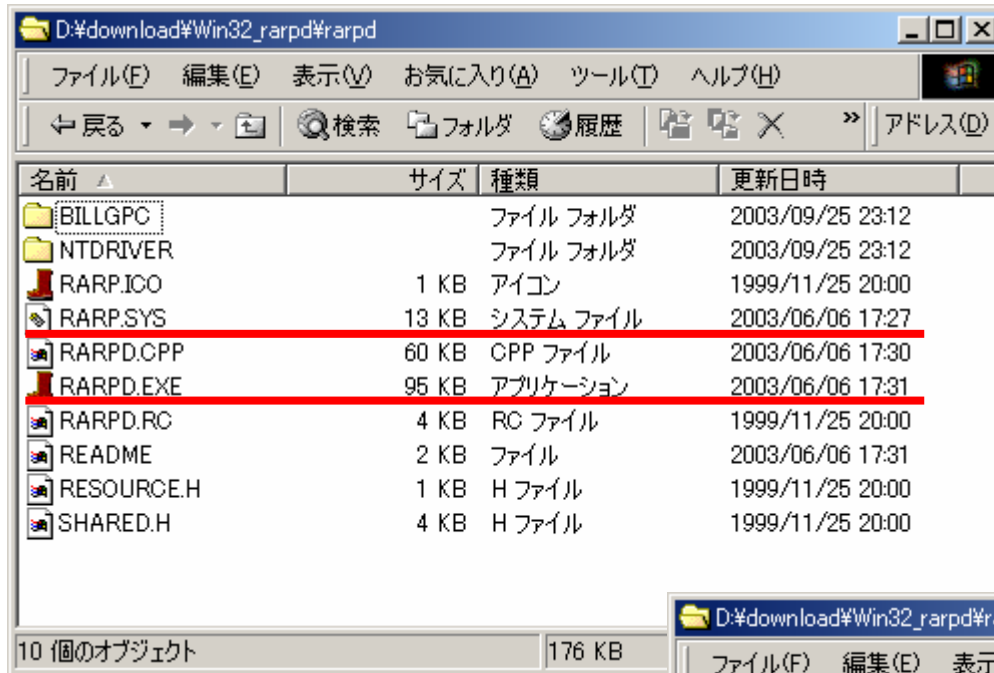
The first two are packaged with this file; the third is your
responsibility. RARP.DTL is a series of ASCII text lines, each
supplying a hex MAC address and its corresponding IP address rendered
in the convential way, e.g:

00.a0.24.19.ae.d4 192.168.0.1

There is no need to do anything once you have updated the file while
RARPD is running; the program will notice that the file has been changed
the next time a RARP request comes in.

There is one possible optional argument, which can be useful with a
```

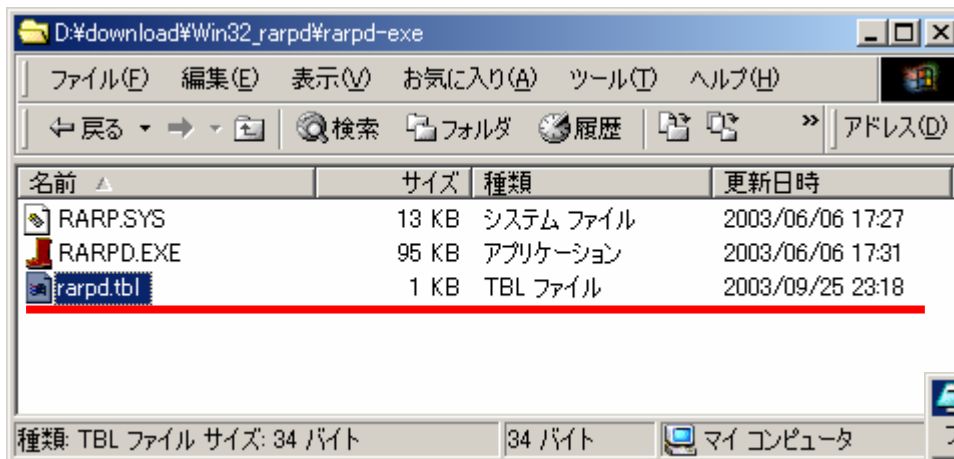
rarpdの実行に必要なファイルをコピーする



2つのファイルを
コピーする。
・RARP.SYS
・RARPD.EXE

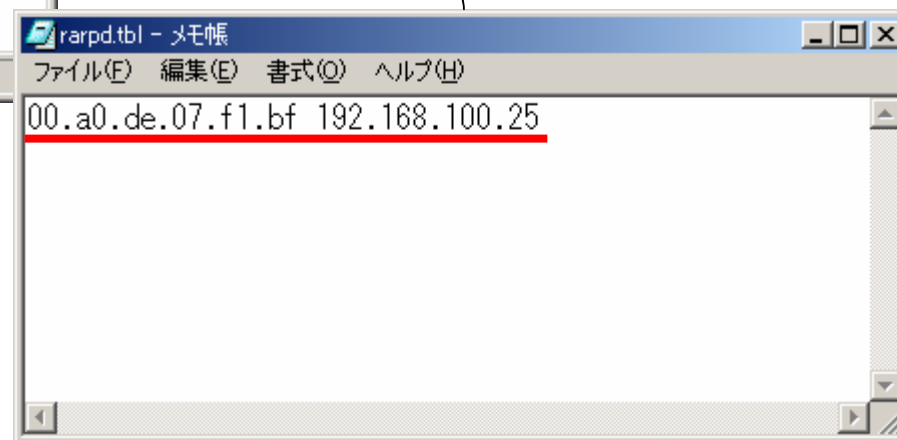
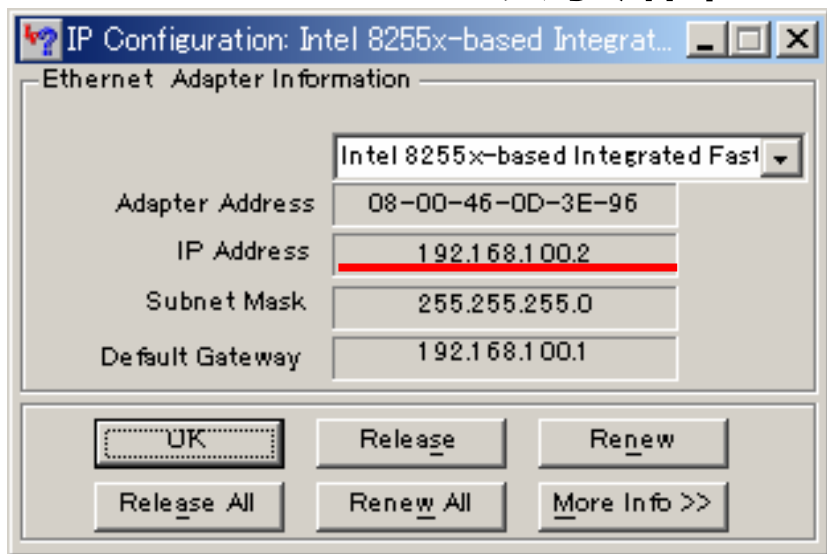


rarpd用設定ファイルを作成し実行する。



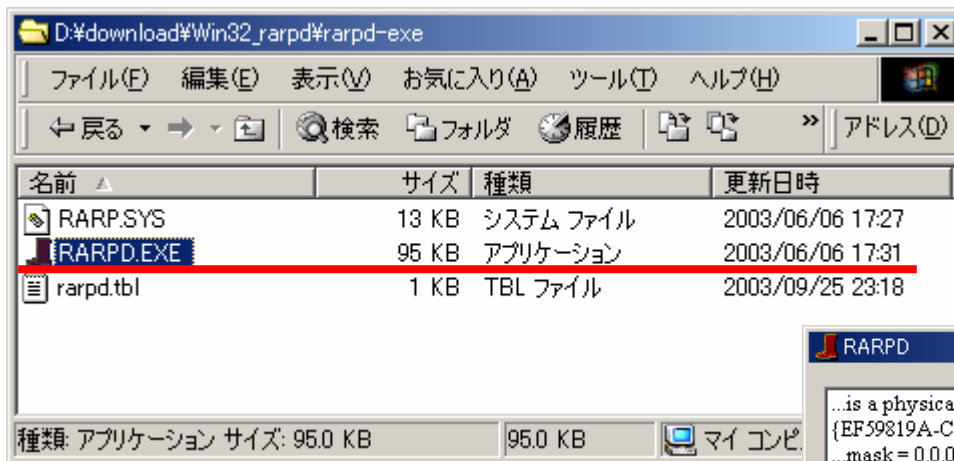
設定ファイルの
“RARP.D.TBL”を
作成する。

rarpdを起動しているWindows 2000のWNTIPCFG表示結果

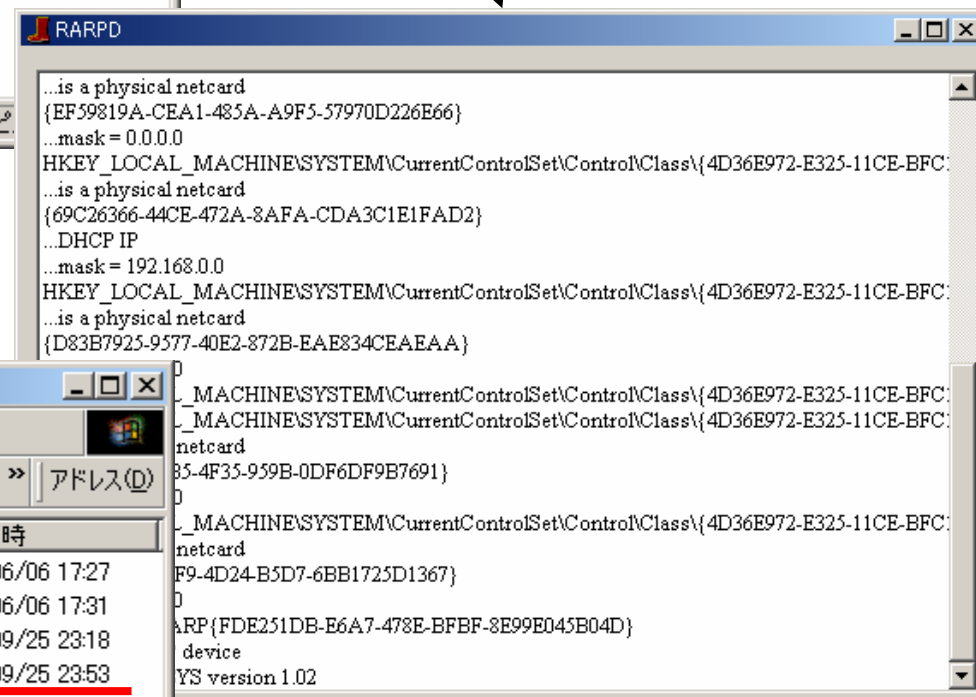


設定ファイルにはRTX1000の
MACアドレスとPCからアクセ
ス可能な暫定IPアドレスを記
述する。

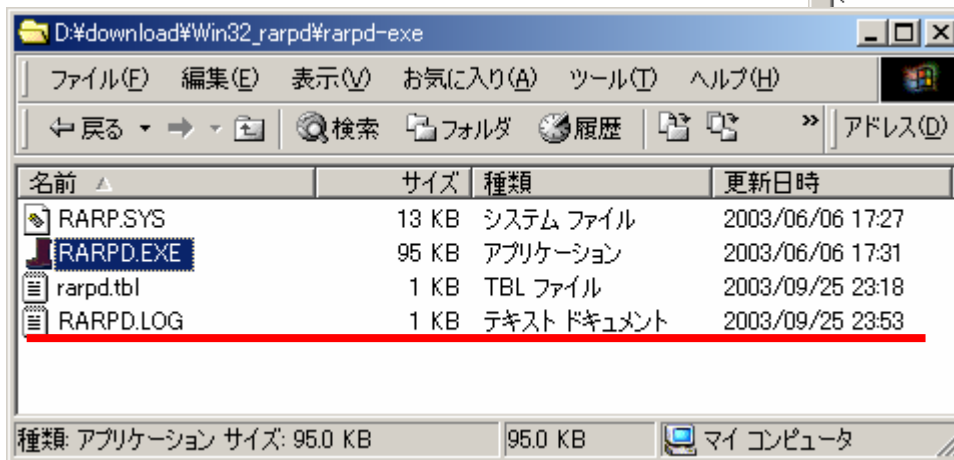
rarpdを起動する。



起動すると起動時メッセージが表示される。動作状態は表示されない。



起動時にログが作成される。



終了時は「×」をクリック。

```
コマンド プロンプト

C:¥>arp -a

Interface: 192.168.100.2 on Interface 0x1000003
Internet Address      Physical Address      Type
192.168.100.1         00-a0-de-07-f2-9e    dynamic

C:¥>ping 192.168.100.25

Pinging 192.168.100.25 with 32 bytes of data:

Reply from 192.168.100.25: bytes=32 time<10ms TTL=64
Reply from 192.168.100.25: bytes=32 time<10ms TTL=64
Reply from 192.168.100.25: bytes=32 time<10ms TTL=64
Reply from 192.168.100.25: bytes=32 time<10ms TTL=64

Ping statistics for 192.168.100.25:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:¥>arp -a

Interface: 192.168.100.2 on Interface 0x1000003
Internet Address      Physical Address      Type
192.168.100.1         00-a0-de-07-f2-9e    dynamic
192.168.100.25        00-a0-de-07-f1-bf    dynamic

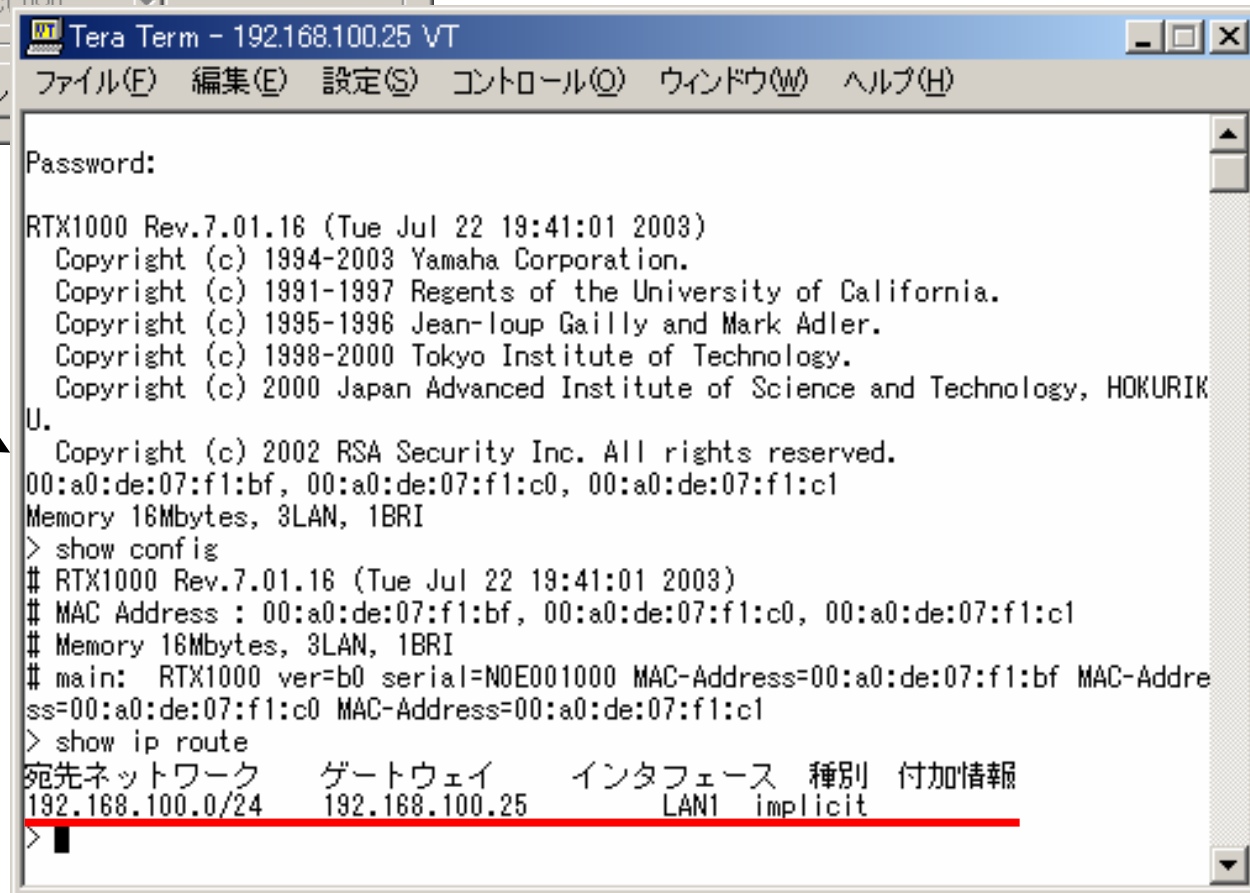
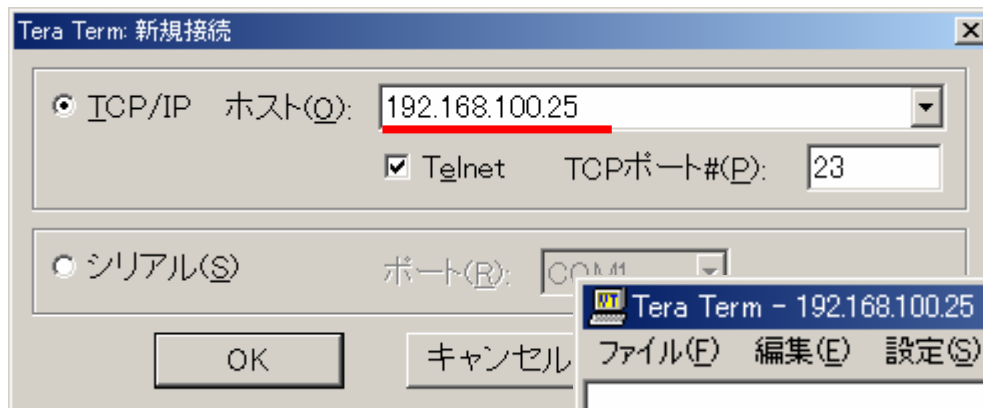
C:¥>
```

RTX1000を起動する。

RTX1000にpingすると返事がある。

ARPテーブルを表示するとRTX1000のMACアドレスが登録されていることが確認できる。

RTX1000にtelnetしてみる

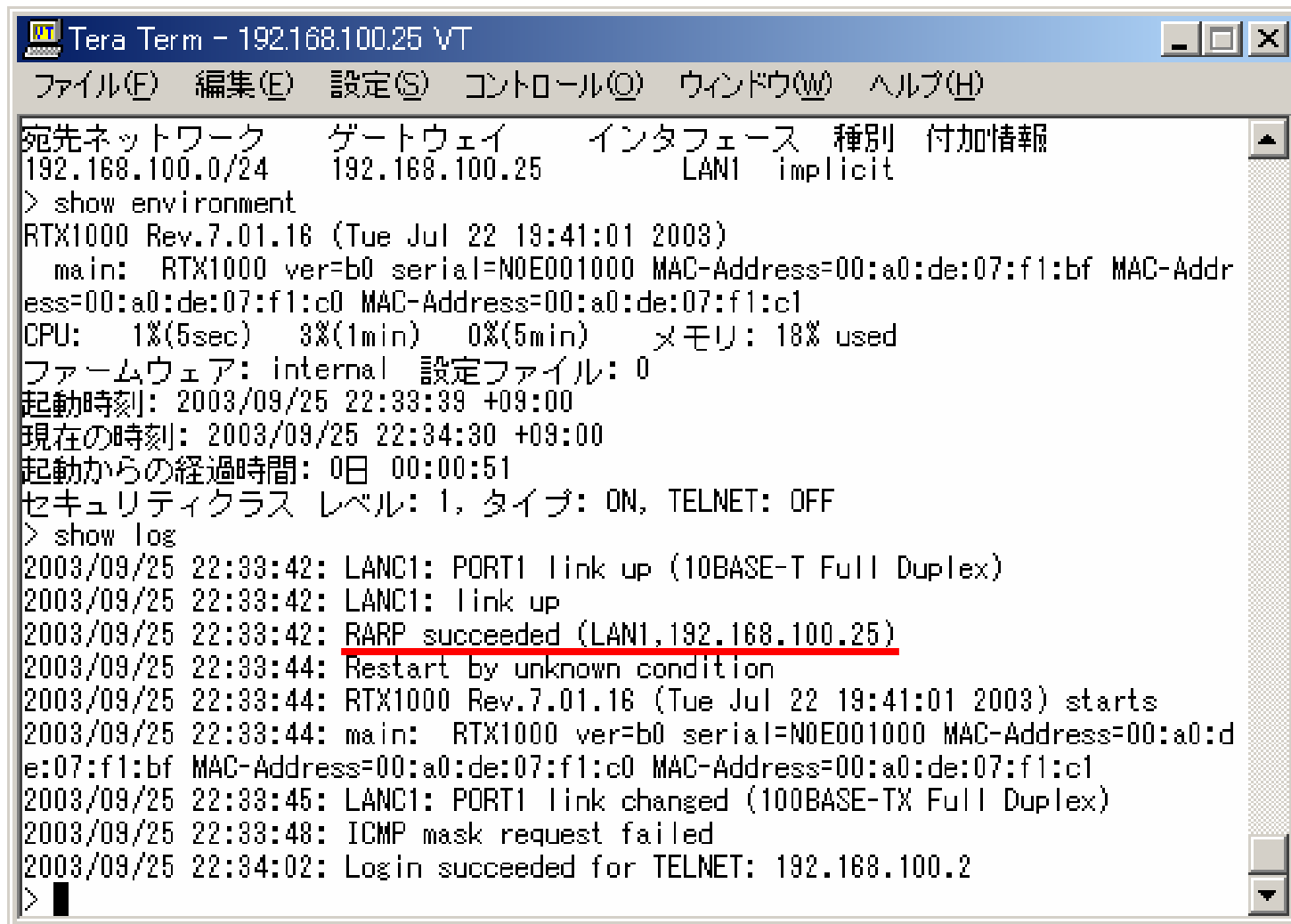


telnetでログイン
してみる。

パスワードも
configも未設定。

経路情報でLAN1
のIPアドレスが確
認可能。

ログに「RARPでIPアドレスを取得した」ことが記録されている。



```
Tera Term - 192.168.100.25 VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
宛先ネットワーク ゲートウェイ インタフェース 種別 付加情報
192.168.100.0/24 192.168.100.25 LAN1 implicit
> show environment
RTX1000 Rev.7.01.16 (Tue Jul 22 19:41:01 2003)
  main: RTX1000 ver=b0 serial=N0E001000 MAC-Address=00:a0:de:07:f1:bf MAC-Addr
ess=00:a0:de:07:f1:c0 MAC-Address=00:a0:de:07:f1:c1
CPU: 1%(5sec) 3%(1min) 0%(5min) メモリ: 18% used
ファームウェア: internal 設定ファイル: 0
起動時刻: 2003/09/25 22:33:39 +09:00
現在の時刻: 2003/09/25 22:34:30 +09:00
起動からの経過時間: 0日 00:00:51
セキュリティクラス レベル: 1, タイプ: ON, TELNET: OFF
> show log
2003/09/25 22:33:42: LANC1: PORT1 link up (10BASE-T Full Duplex)
2003/09/25 22:33:42: LANC1: link up
2003/09/25 22:33:42: RARP succeeded (LAN1,192.168.100.25)
2003/09/25 22:33:44: Restart by unknown condition
2003/09/25 22:33:44: RTX1000 Rev.7.01.16 (Tue Jul 22 19:41:01 2003) starts
2003/09/25 22:33:44: main: RTX1000 ver=b0 serial=N0E001000 MAC-Address=00:a0:d
e:07:f1:bf MAC-Address=00:a0:de:07:f1:c0 MAC-Address=00:a0:de:07:f1:c1
2003/09/25 22:33:45: LANC1: PORT1 link changed (100BASE-TX Full Duplex)
2003/09/25 22:33:48: ICMP mask request failed
2003/09/25 22:34:02: Login succeeded for TELNET: 192.168.100.2
>
```

ARPとRARPの書式



0	7	8	15	16	23	24	31
ハードウェアタイプ (=Ethernet:0x0001)				プロトコルタイプ (=IP:0x0800)			
HLEN (=6 octets)		PLEN (=4 octets)		オペレーション			
送信元MACアドレス							
送信元MACアドレス(続き)				送信元IPアドレス			
送信元IPアドレス(続き)				探索するMACアドレス			
探索するMACアドレス(続き)							
探索するIPアドレス							

ARP(Address Resolution Protocol)とRARP(Reverse Address Resolution Protocol)の書式差分点

	ARP request	ARP reply	RARP request	RARP reply
Ethernet type	0x0806	0x0806	0x8035	0x8035
ハードウェアタイプ(Ethernet:0x0001)	0x0001	0x0001	0x0001	0x0001
プロトコルタイプ (IP:0x0800)	0x0800	0x0800	0x0800	0x0800
オペレーション	0x0001	0x0002	0x0003	0x0004
HLEN: MACアドレスの長さ(オクテット)	6	6	6	6
PLEN: IPアドレスの長さ(オクテット)	4	4	4	4

RARP(request: RTX1000 PC/RARPD)



The screenshot shows the Wireshark interface with a packet capture of a RARP reverse request. The packet list pane shows two packets:

No.	Time	Source	Destination	Protocol	Info
1	0.000000	YAMAHA_07:f1:bf	ff:ff:ff:ff:ff:ff	RARP	who is 00:a0:de:07:f1:bf?
2	0.000509	sony_0d:3e:96	YAMAHA_07:f1:bf	RARP	08:00:46:0d:3e:96 is at 0.0.0.0

The packet details pane for the selected packet (Frame 1) shows the following structure:

- Ethernet II
 - Destination: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)
 - Source: 00:a0:de:07:f1:bf (YAMAHA_07:f1:bf)
 - Type: RARP (0x8035)
 - Trailer: 00000000000000000000000000000000...
- Address Resolution Protocol (reverse request)
 - Hardware type: Ethernet (0x0001)
 - Protocol type: IP (0x0800)
 - Hardware size: 6
 - Protocol size: 4
 - Opcode: reverse request (0x0003)
 - Sender hardware address: 00:a0:de:07:f1:bf
 - Sender protocol address: 0.0.0.0
 - Target hardware address: 00:a0:de:07:f1:bf
 - Target protocol address: 0.0.0.0

The packet bytes pane shows the raw data in hexadecimal and ASCII:

```
0000  ff ff ff ff ff ff 00 a0 de 07 f1 bf 80 35 00 01  .....5..
0010  08 00 06 04 00 03 00 a0 de 07 f1 bf 00 00 00 00  .....
0020  00 a0 de 07 f1 bf 00 00 00 00 00 00 00 00 00 00  .....
0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
```

The filter bar at the bottom shows the filter: `Address Resolution Protocol (arp)`.

RARP(reply: PC/RARPD RTX1000)



The screenshot shows the Wireshark interface with a packet capture of a RARP reverse reply. The packet list pane shows two packets:

No.	Time	Source	Destination	Protocol	Info
1	0.000000	YAMAHA_07:f1:bf	ff:ff:ff:ff:ff:ff	RARP	who is 00:a0:de:07:f1:bf?
2	0.000509	Sony_0d:3e:96	YAMAHA_07:f1:bf	RARP	08:00:46:0d:3e:96 is at 0.0

The packet details pane for Frame 2 (42 on wire, 42 captured) shows the following structure:

- Ethernet II
 - Destination: 00:a0:de:07:f1:bf (YAMAHA_07:f1:bf)
 - Source: 08:00:46:0d:3e:96 (MKT-hirano-vaio)
 - Type: RARP (0x8035)
- Address Resolution Protocol (reverse reply)
 - Hardware type: Ethernet (0x0001)
 - Protocol type: IP (0x0800)
 - Hardware size: 6
 - Protocol size: 4
 - Opcode: reverse reply (0x0004)
 - Sender hardware address: 08:00:46:0d:3e:96
 - Sender protocol address: 0.0.0.0
 - Target hardware address: 00:a0:de:07:f1:bf
 - Target protocol address: 192.168.100.25

The packet bytes pane shows the raw data in hexadecimal and ASCII:

```
0000  00 a0 de 07 f1 bf 08 00 46 0d 3e 96 80 35 00 01  ..... F.>..5..
0010  08 00 06 04 00 04 08 00 46 0d 3e 96 00 00 00 00  ..... F.>.....
0020  00 a0 de 07 f1 bf c0 a8 64 19  ..... d.
```

The filter bar at the bottom shows: Filter: [] / Reset Address Resolution Protocol (arp)

ARP(request: PC RTX1000)



The screenshot shows the Wireshark interface with a packet capture of an ARP request. The packet list pane shows two packets: packet 4 is an ARP request from MKT-hirano-vaio to ff:ff:ff:ff:ff:ff, and packet 5 is an ARP response from YAMAHA_07:f1:bf to MKT-hirano-vaio. Packet 4 is selected, and the packet details pane shows the following information:

- Frame 4 (42 on wire, 42 captured)
- Ethernet II
 - Destination: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)
 - Source: 08:00:46:0d:3e:96 (MKT-hirano-vaio)
 - Type: ARP (0x0806)
- Address Resolution Protocol (request)
 - Hardware type: Ethernet (0x0001)
 - Protocol type: IP (0x0800)
 - Hardware size: 6
 - Protocol size: 4
 - Opcode: request (0x0001)
 - Sender hardware address: 08:00:46:0d:3e:96
 - Sender protocol address: 192.168.100.2
 - Target hardware address: 00:00:00:00:00:00
 - Target protocol address: 192.168.100.25

The packet bytes pane shows the raw data for the selected packet:

```
0000  ff ff ff ff ff ff 08 00 46 0d 3e 96 08 06 00 01  ..... F.>....  
0010  08 00 06 04 00 01 08 00 46 0d 3e 96 c0 a8 64 02  ..... F.>...d.  
0020  00 00 00 00 00 00 c0 a8 64 19  ..... d.
```

The filter bar at the bottom shows the filter: `Address Resolution Protocol (arp)`.

ARP(reply: RTX1000 PC)



The screenshot shows the Wireshark interface with a packet capture of an ARP reply. The packet list pane shows two packets: packet 4 is an ARP request from MKT-hirano-vaio to ff:ff:ff:ff:ff:ff asking for 192.168.100.25, and packet 5 is an ARP reply from YAMAHA_07:f1:bf to MKT-hirano-vaio for 192.168.100.25. Packet 5 is selected.

The packet details pane for Frame 5 (60 on wire, 60 captured) shows:

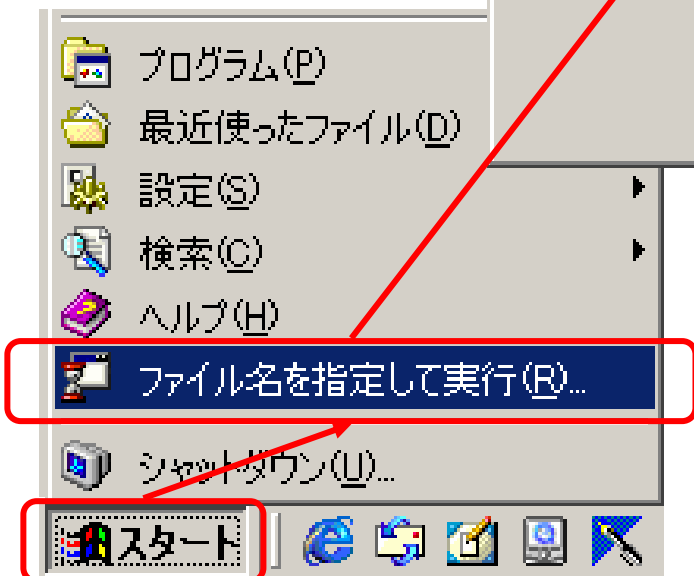
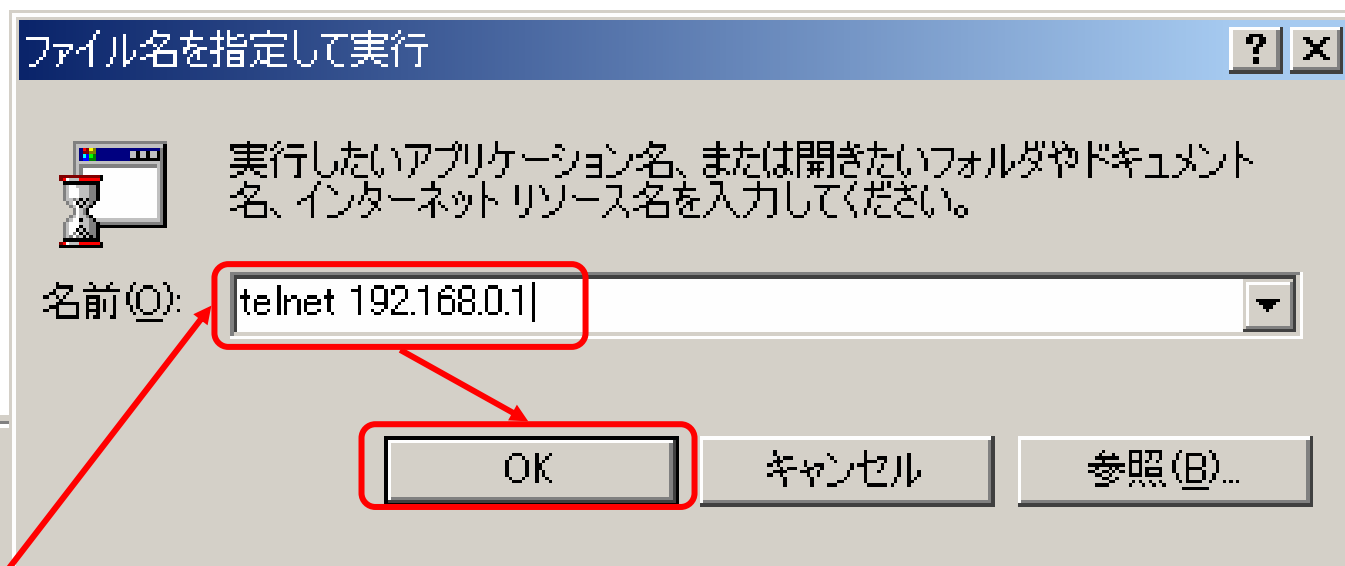
- Ethernet II
 - Destination: 08:00:46:0d:3e:96 (MKT-hirano-vaio)
 - Source: 00:a0:de:07:f1:bf (YAMAHA_07:f1:bf)
 - Type: ARP (0x0806)
 - Trailer: 00000000000000000000000000000000...
- Address Resolution Protocol (reply)
 - Hardware type: Ethernet (0x0001)
 - Protocol type: IP (0x0800)
 - Hardware size: 6
 - Protocol size: 4
 - opcode: reply (0x0002)
 - Sender hardware address: 00:a0:de:07:f1:bf
 - Sender protocol address: 192.168.100.25
 - Target hardware address: 08:00:46:0d:3e:96
 - Target protocol address: 192.168.100.2

The packet bytes pane shows the raw data in hexadecimal and ASCII:

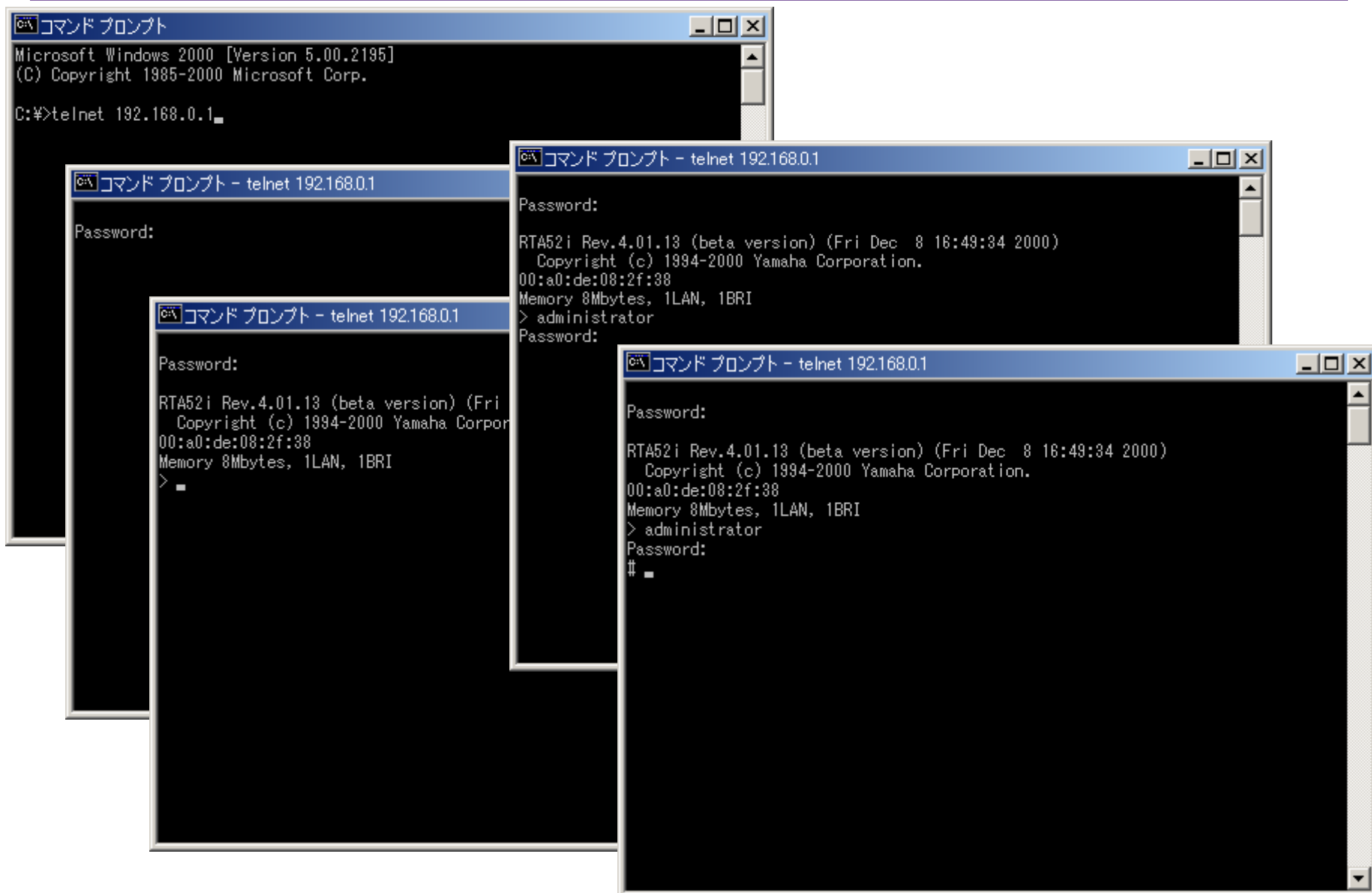
```
0000  08 00 46 0d 3e 96 00 a0 de 07 f1 bf 08 06 00 01  ..F.>... ..  
0010  08 00 06 04 00 02 00 a0 de 07 f1 bf c0 a8 64 19  ..F.>... ..d.  
0020  08 00 46 0d 3e 96 c0 a8 64 02 00 00 00 00 00 00  ..F.>... d.....  
0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
```

telnet コマンド

書式: telnet IPアドレスまたはホスト名 ポート番号



telnet コマンド



```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C: #>telnet 192.168.0.1

C:\ コマンド プロンプト - telnet 192.168.0.1
Password:

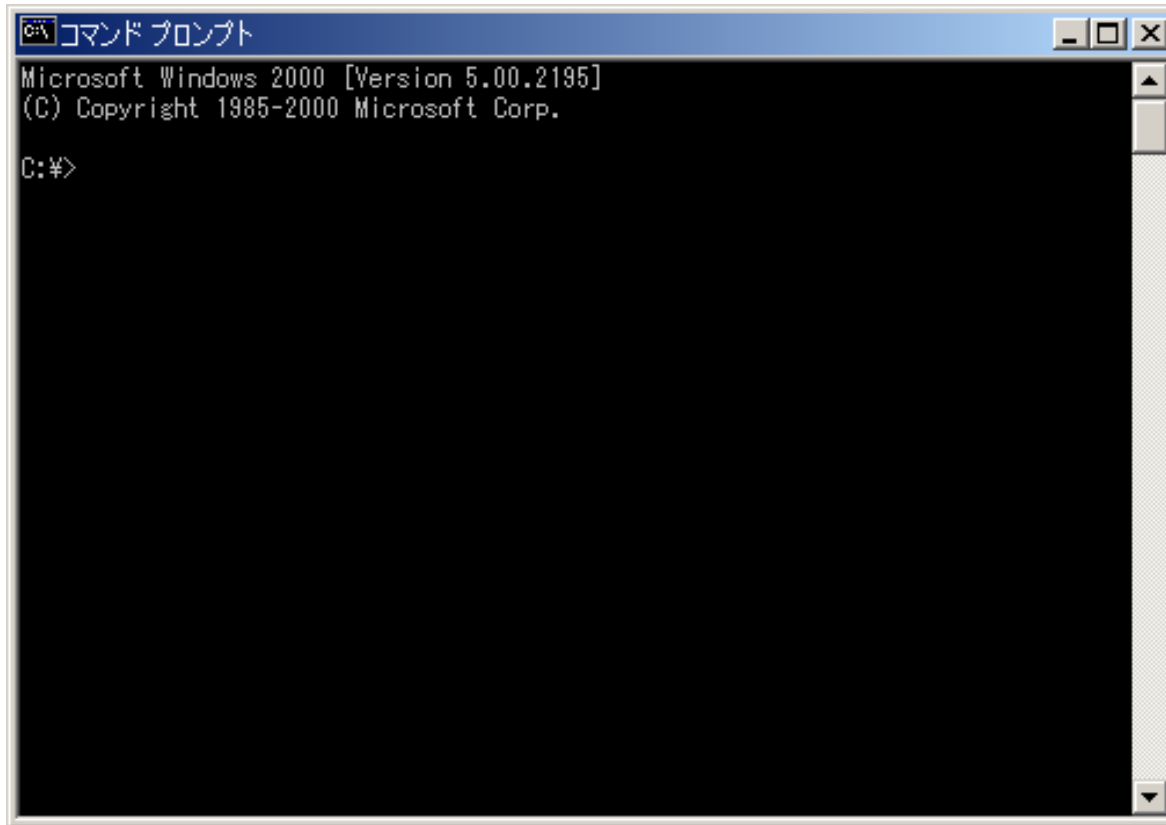
C:\ コマンド プロンプト - telnet 192.168.0.1
Password:

RTA52i Rev.4.01.13 (beta version) (Fri Dec 8 16:49:34 2000)
Copyright (c) 1994-2000 Yamaha Corporation.
00:a0:de:08:2f:38
Memory 8Mbytes, 1LAN, 1BRI
> administrator
Password:

C:\ コマンド プロンプト - telnet 192.168.0.1
Password:

RTA52i Rev.4.01.13 (beta version) (Fri Dec 8 16:49:34 2000)
Copyright (c) 1994-2000 Yamaha Corporation.
00:a0:de:08:2f:38
Memory 8Mbytes, 1LAN, 1BRI
> administrator
Password:
#
```

MS-DOS プロンプト



[スタート]
[プログラム(P)]
[アクセサリ]
[コマンド プロンプト]

ステータスバーで、
マウスの右ボタンを
クリックすると出る
メニューのプロパティ
でカスタマイズできる。
・色、フォント、...
・ヒストリ
・スクロールバー

[ウィンドウのキャプチャー] Alt + PrtSc

[スクリーンのキャプチャー] Shift + PrtSc



Tera Term Pro

Windows用のフリーソフトウェアのターミナルエミュレータ (通信ソフト)です。VT100エミュレーション、telnet接続、シリアル接続などが可能です。

また、第三者によるTeraTermを拡張するモジュールもいくつか公開されています。

<http://hp.vector.co.jp/authors/VA002416/>

<http://www.sakurachan.org/soft/teraterm-j/files/tterm23.zip>

<ftp://www.sakurachan.org/pub/windows/net/term/teraterm/tterm23.zip>

<http://www.vector.co.jp/authors/VA002416/tterm23.zip>

<ftp://riksun.riken.go.jp/pub/pc/misc/terminal/teraterm/tterm23.zip>

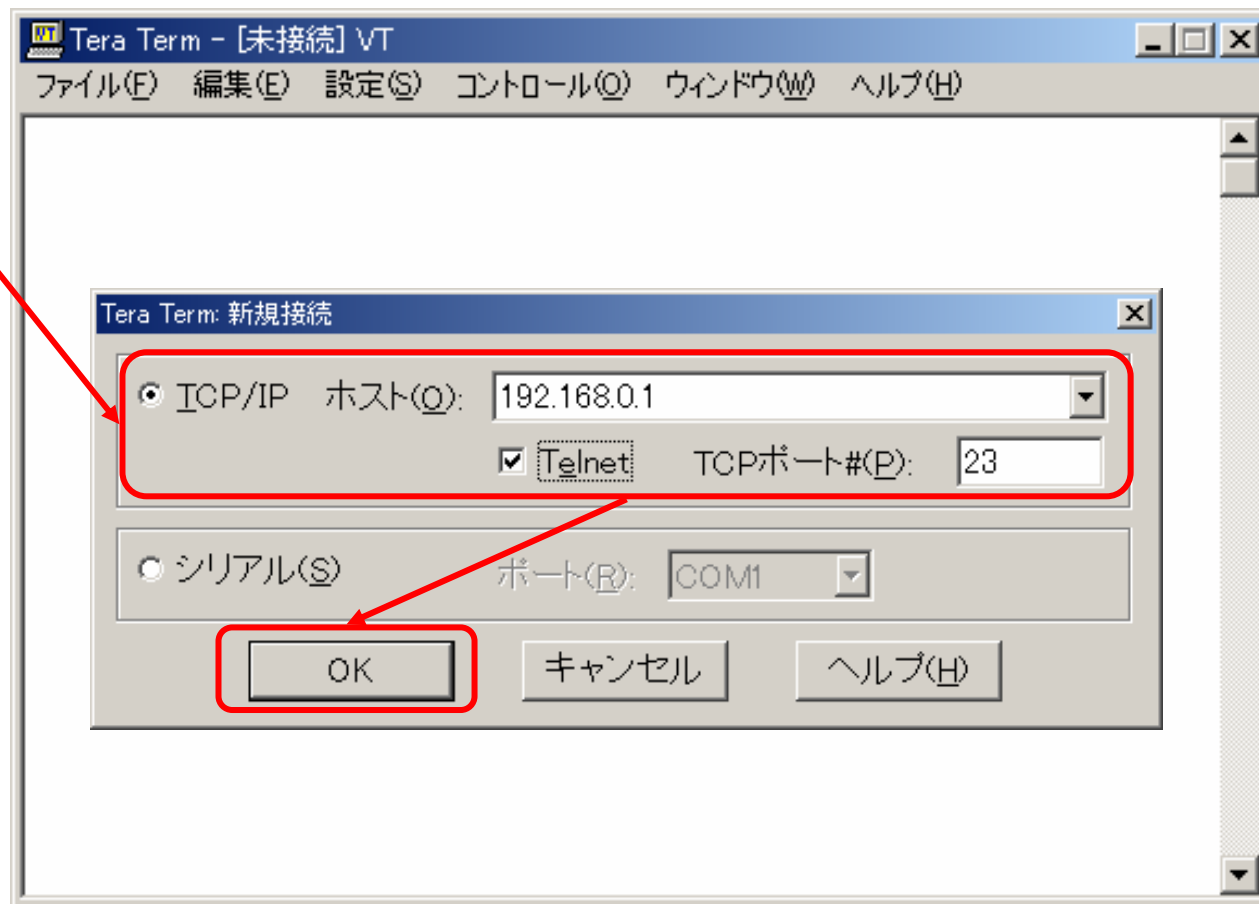
<ftp://ftp.forest.impress.co.jp/pub/win/winsoc/apps/teraterm/tterm23.zip>

<ftp://ftp.s.u-tokyo.ac.jp/PC/terminal/teraterm/tterm23.zip>

TeraTerm Pro (telnetで接続)



[スタート]
[プログラム(P)]
[Tera Term Pro]
[Tera Term Pro]



TeraTerm Pro (漢字コード選択)



The image shows three overlapping windows of TeraTerm Pro. The top-left window displays the terminal's startup screen with copyright information for Yamaha Corporation and the Tokyo Institute of Technology. The top-right window shows the 'Terminal Settings' dialog box, with the '端末(T)...' menu option highlighted in red. The bottom window shows the terminal output of the 'show environment' command, with a red box highlighting the 'console character' section. The 'console character' section shows the current input mode as 'console character 文字コード' and lists the available character codes: 'ascii', 'sjis', and 'euc'. The '漢字(受信(K))' and '漢字(送信(X))' dropdown menus in the dialog box are also highlighted in red, with 'SJIS' selected in both. The '漢字(送信(X))' dropdown menu also shows 'かな' and '7bit カタカナ' options.

```
RTA54i Rev.4.04.08 (Tue Jan 15 14:08:13 2002)
Copyright (c) 1994-2002 Yamaha Corporation.
Copyright (c) 1998-2000 Tokyo Institute of Technology.
Copyright (c) 2000 Japan Advanced Institute of Science and Technology, HOKURIKI
U.
00:a0:de:00:4d:50, 00:a0:de:00:4d:51
Memory 8Mbytes, 2LAN, 1BRI
>

> show environment
RTA54i Rev.4.04.08 (Tue Jan 15 14:08:13 2002)
MACアドレス: 00:a0:de:00:4d:50, 00:a0:de:00:4d:51   メモリ: 16% used
起動時刻: 2002/05/04 22:22:41 +09:00
現在の時刻: 2002/05/04 22:42:35 +09:00
起動からの経過時間: 0日 00:19:54
セキュリティクラス レベル: 1, タイプ: ON, TELNET: OFF
リモートセットアップ許可: ANY
ログインタイム: 300秒
コンソール: 115200bit/s, SJIS, 80 x 24
システムメッセージ: OFF
課金閾値: OFF
擬似LAN接続: ON
> console character ?
  入力形式: console character 文字コード
            文字コード = 'ascii', 'sjis', or 'euc'
            説明: コンソールポートの出力文字コードを選択します
デフォルト値: sjis
> console character
```

漢字(受信(K)) EUC SJIS EUC JIS

漢字(送信(X)) EUC かな 7bit カタカナ

漢字イン(N): ^[\$B

漢字アウト(O): ^[J

端末サイズ(T) 80 x 24

改行コード 受信(R): CR 送信(M): CR

端末ID: VT100

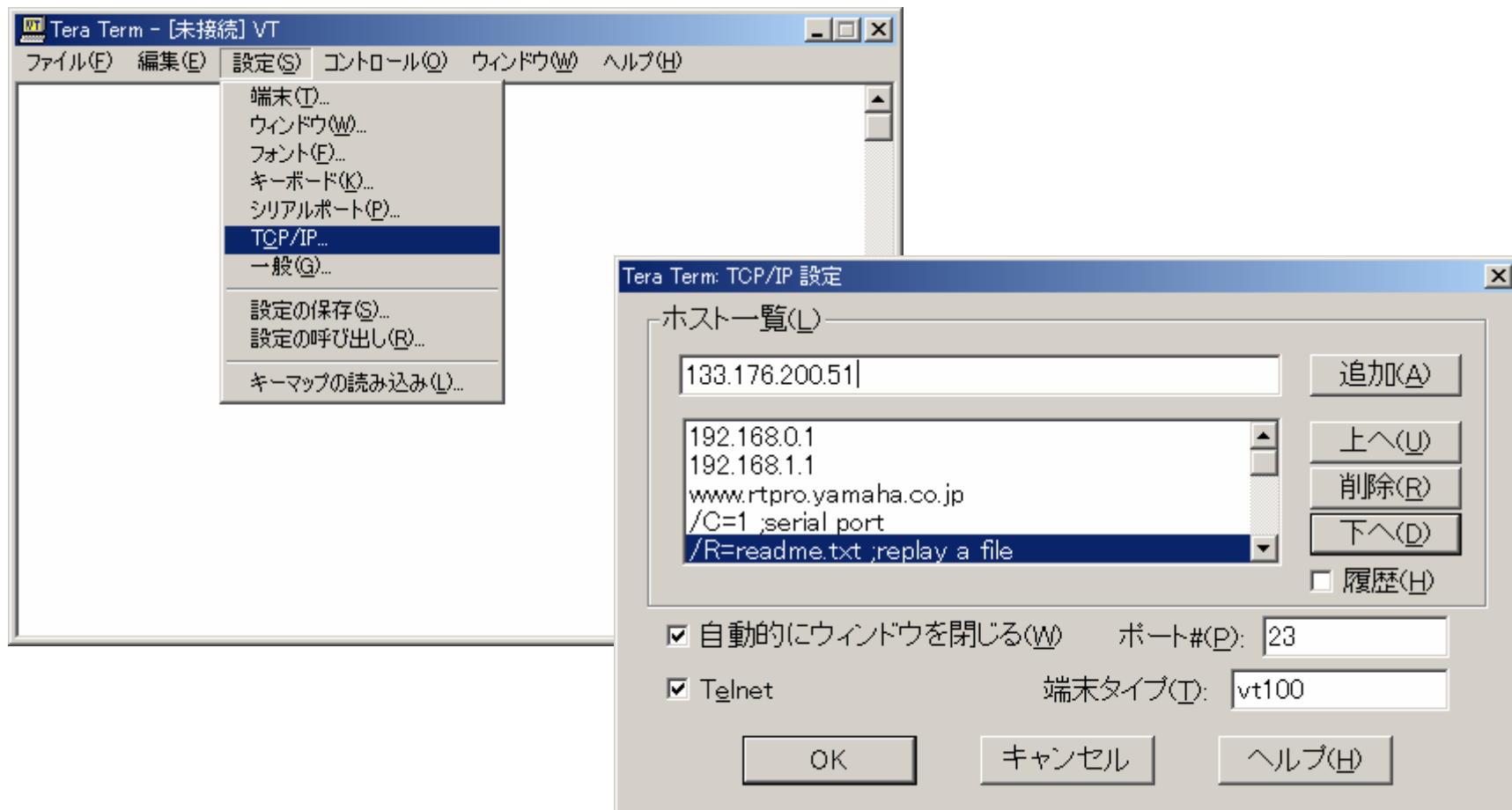
ローカルエコー(L)

自動切り替え(VT<->TEK(U))

OK キャンセル ヘルプ(H)

vs (32bits)

TeraTerm Pro (接続先の登録)





ethereal

フリーのLANアナライザー(プロトコルアナライザー)
いろいろなOSで利用できる

Windows2000/XPなど: WinPcapを併用する。

FreeBSD...ports

フィルタリングの書式は、UNIXのtcpdump準拠

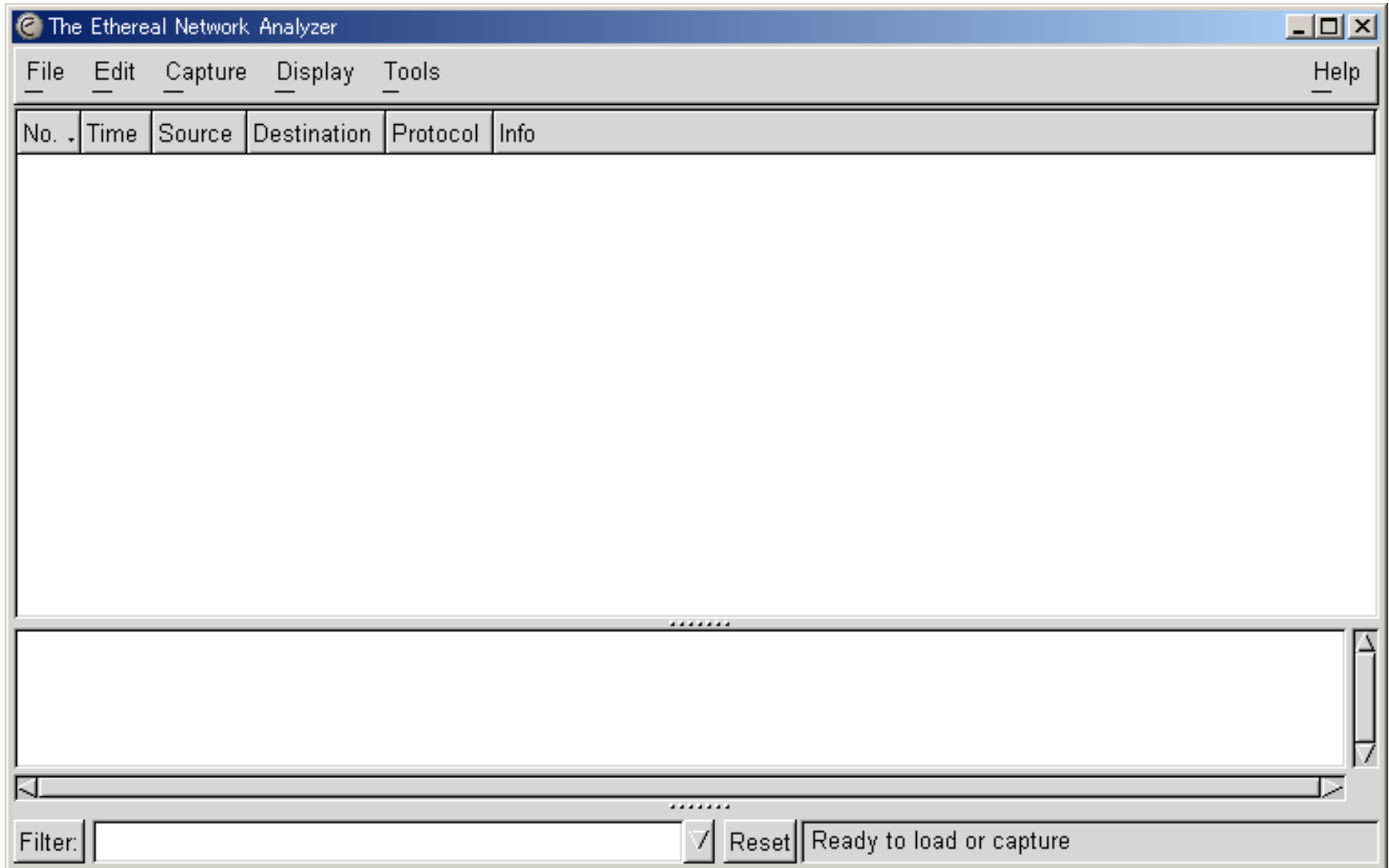
<http://www.ethereal.com/>

WinPcap

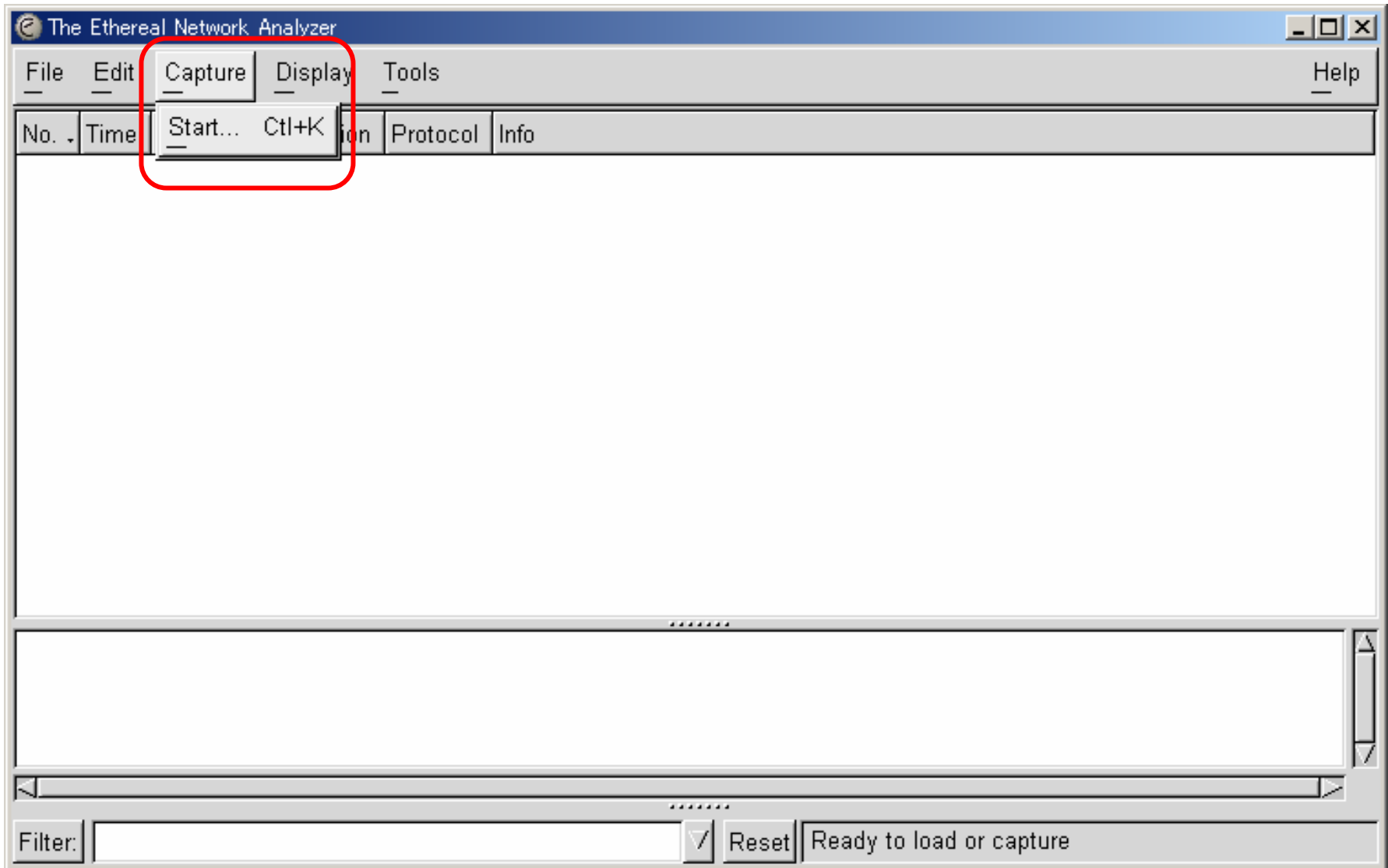
Windows用キャプチャードライバー

<http://netgroup-serv.polito.it/winpcap/install/default.htm>

ethereal(起動時)



ethereal(キャプチャ開始)



ethereal(取り込み 終了)

