
AGGRESSOR NETWORK

AGGRESSOR NETWORK 操作説明書

リリース **0.1**

株式会社 **bitset**

2023 年 09 月 29 日

目次:

第 1 章	概要	1
1.1	仕様	2
1.2	製品構成	3
1.3	別途、用意する機材	5
第 2 章	セットアップ	6
2.1	接続手順	8
2.2	起動	9
2.3	終了	9
第 3 章	操作説明	10
3.1	画面構成	10
3.2	メイン画面 (0:main)	11
3.3	ヘルプ画面 (1:HELP)	14
3.4	通信量画面 (2:I/O Stat)	15
3.5	eth0 画面 (3:eth0) / eth1 画面 (4:eth1)	16
第 4 章	コマンド	19
4.1	コマンド一覧	19
4.2	一般操作	21
4.3	フィルター設定	25
4.4	パケットキャプチャ	36
4.5	システム設定	40
第 5 章	補足	41
5.1	補足事項	41
5.2	改定履歴	42
5.3	問い合わせ先	42

第1章 概要

本製品はさまざまなネットワーク障害を発生させるネットワークエミュレータで、帯域制御、パケットフィルタ（遅延/損失/重複/破損/順序変更）を発生させることができます。また USB メモリにパケットキャプチャ結果を保存可能です。

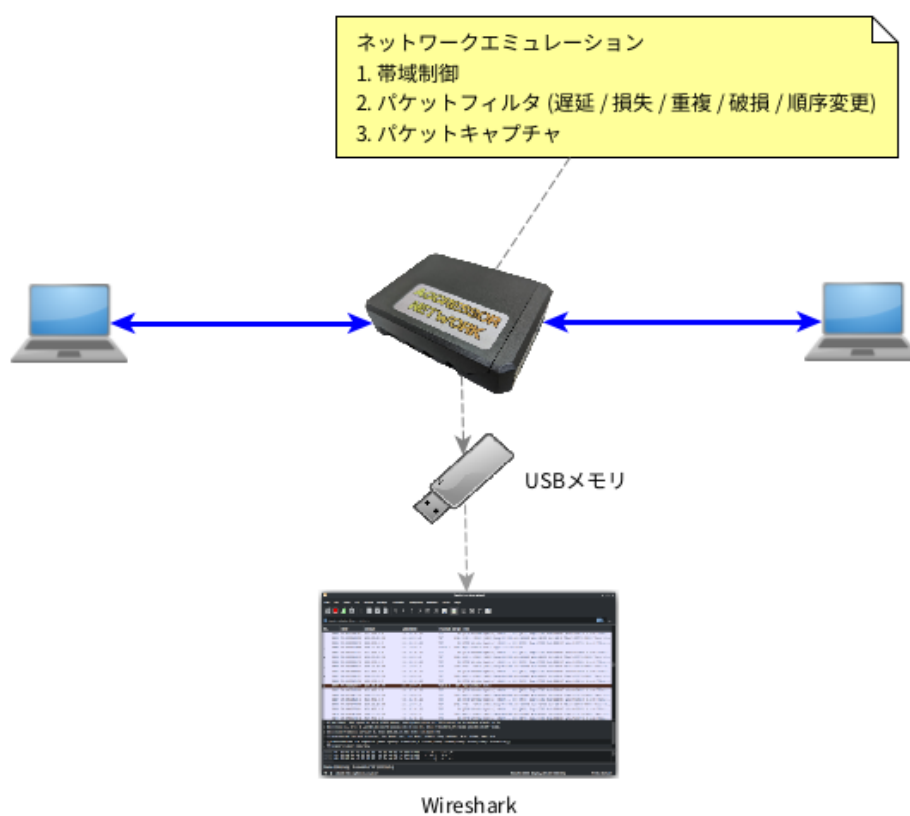


図 1.1: 使用イメージ

1.1 仕様

表 1.1: ネットワーク

ネットワーク	有線
ポート数	2 ポート (本体, USB アダプタ)
最大速度	1Gbps
IP アドレス	使用しません

注釈: 本製品は IP アドレスを使用せず、ネットワーク HUB と同様にパケットを転送します。

表 1.2: パケットフィルター

遅延	遅延時間を指定。ジッター (ゆらぎ) 指定可能
損失	発生率を指定
重複	発生率を指定
破損	発生率を指定
順序変更	発生率を指定
帯域制御	通信速度を指定 (bps - Gbps)

表 1.3: パケットキャプチャ

データ保存形式	PCAP ... Wireshark で表示可能
データ保存先	USB メモリ
USB メモリ	FAT フォーマット

表 1.4: その他

表示	HDMI モニタ、解像度: フル HD
入力デバイス	USB キーボード (US/JP 切替可)
電源	USB Type-C 5V/3A 以上
RTC	内蔵 (交換用電池 : CR1225)

1.2 製品構成

表 1.5: 本体に含まれるもの

機材	写真	補足
1. 本体		<ul style="list-style-type: none">• Ethernet Gbit Ethernet x1• USB3 x2• USB2 x2• microHDMI x2• USB Type-C (電源専用) x1• RTC 内蔵
2. USB ネットワークアダプタ		<ul style="list-style-type: none">• USB3.0 Gbit Ethernet 対応
3. microHDMI - HDMI 変換アダプタ		
4. USB 電源		<ul style="list-style-type: none">• USB Type-C 5V/3A 以上

1.2.1 ポートの構成

(1) USB, Ethernet

青い USB ポートが USB3, 黒いポートが USB2 です。



図 1.2: Ethernet, USB2, USB3, Ethernet

表 1.6: Ethernet ポートのデバイス名

ポート	イーサネットデバイス名
Ethernet ポート	eth0
USB ネットワークアダプタ	eth1

表 1.7: USB ポートの用途

ポート	用途
USB3 (青)	USB ネットワークアダプタ USB メモリ
USB2 (黒)	キーボード

(2) 電源、HDMI

microHDMI は 2 ポートありますが、どちらに接続しても構いません。



図 1.3: USB 電源, microHDMI

1.3 別途、用意する機材

以下の機材は製品に含まれないので、お客様でご用意ください。

表 1.8: お客様でご用意して頂くもの

機材	数量	仕様
キーボード	1	USB, 英語 or 日本語
ディスプレイ	1	解像度: フル HD, HDMI 対応
Ethernet ケーブル	2	
USB メモリ	1	USB3.0 対応が望ましい パケットキャプチャする場合のみ必要

注釈: 上記以外に、本機の両ネットワークに接続するネットワーク機器が必要です。

第2章 セットアップ

周辺機器の接続から起動・終了までの手順を説明します。

本体と各機器は以下のように接続します。

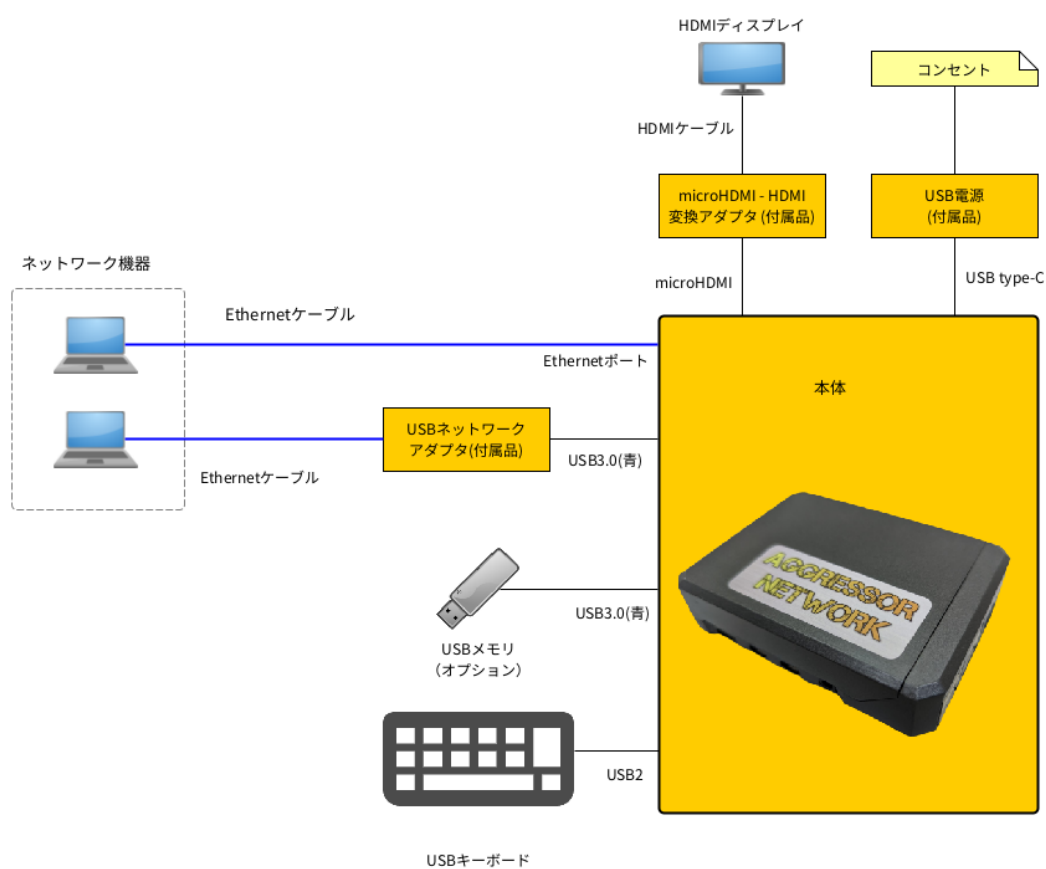


図 2.1: 接続図

- ネットワーク機器は PC を直接接続して、HUB 経由で接続しても構いません。

2.1 接続手順

本装置に以下のように各ケーブルを接続します。



図 2.3: 本体への接続

1. ネットワークケーブルの接続: eth0

本体のネットワークポートに Ethernet ケーブルを接続します。

2. USB ネットワークアダプタの接続: eth1

USB3(青) のポートに USB ネットワークアダプタを接続します。

また、アダプタの先に Ethernet ケーブルを接続します。

3. キーボードの接続

USB2(黒) のポートに キーボードを接続します。

4. ディスプレイの接続

- 付属の microHDMI ・ HDMI 変換アダプタを使って HDMI ディスプレイを接続します。
- どちらのポートに接続しても大丈夫です。

4. 電源の接続

USB Type-C ポートに USB 電源を接続します。

2.2 起動

1. USB 電源のスイッチを ON にします。
2. ディスプレイに起動画面が表示されるまで待ちます。

2.3 終了

1. USB 電源のスイッチを OFF にします。

注釈: 本装置は電源ブチ切り対応なので、シャットダウン等の操作は不要です。

第3章 操作説明

3.1 画面構成

本装置の実行画面です。



図 3.1: 実行画面

画面は次の5つに分割されます。

表 3.1: 画面構成

画面タイトル	本書での名称	機能
0 main	0: メイン画面	コマンド入力画面です。
1 HELP	1: ヘルプ画面	ヘルプを表示します。
2 I/O Stat	2: 通信量画面	毎秒の通信量を表示します。
3 eth0	3:eth0 画面	eth0 の通信状態を表示します。(iftop / tcpdump 切り替え)
4 eth1	4:eth1 画面	eth1 の通信状態を表示します。(iftop / tcpdump 切り替え)

NETWORK EMULATOR USB

0 "main"

NETWORK EMULATOR

copyright (c) bitset.jp 2023 <https://bitset.jp>

--- 18:20:46 ---

```
[0] eth0:  qdisc netem 801d: root refcnt 6 limit 1000
Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
backlog 0b 0p requeues 0
[1] eth1:  qdisc netem 801e: root refcnt 2 limit 1000
Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
backlog 0b 0p requeues 0
```

a > _

ネットワークデバイス選択状態

フィルター情報

キー入力は全てこの画面で行い、以下のことができます。

1. ネットワークデバイスの選択
2. フィルターの設定
3. パケットキャプチャ
4. システム設定
5. ヘルプ表示の指示

3.2.1 フィルター情報表示

ENTER を押すか、コマンドを実行するたびに最新のフィルター情報を表示します。

```
--- 17:49:29 ---
[0] eth0:  qdisc netem 8003: root refcnt 6 limit 1000
    Sent 19814 bytes 244 pkt (dropped 0, overlimits 0 requeues 0)
    backlog 0b 0p requeues 0
[1] eth1:  qdisc netem 8004: root refcnt 2 limit 1000
    Sent 90879 bytes 933 pkt (dropped 0, overlimits 0 requeues 0)
    backlog 0b 0p requeues 0
```

図 3.3: フィルター設定情報

- 1 行目 ("[0] eth0:... ", "[1] eth1:... ")
フィルター設定です。フィルターの詳細は [フィルター設定](#) を参照願います。
- 2-3 行目 (青文字)
ネットワークデバイス毎の通信統計です。

表 3.2: 表示項目

項目	意味
Sent	送信バイト数、パケット数
dropped	破棄
overlimit	キューが一杯になった回数
requeues	再送のためのキューイング
backlog	T.B.D

3.2.2 ネットワークデバイスの選択

プロンプトのネットワークデバイスの選択状態を表します。

表 3.3: プロンプト

プロンプト	選択状態
0 >	eth0 を選択
1 >	eth1 を選択
a >	両方 (eth0, eth1) を選択

デバイス選択コマンド (0 / 1 / a) で切り替えることができます。

0: eth0 を選択:

```
a > 0 [ENTER]
0 >
```

1: eth1 を選択:

```
0 > 1 [ENTER]
1 >
```

a: 両方 (eth0, eth1) を選択:

```
0 > a [ENTER]
a >
```

3.3 ヘルプ画面 (1:HELP)

コマンドのヘルプを表示します。

```

1 "HELP"
--- FILTER OPTION ---
- delay TIME (JITTER) (PERCENT)
  ex: delay 100ms 20ms 10%
- loss PERCENT
  ex: loss 10%
- duplicate PERCENT
  ex: duplicate 10%
- corrupt PERCENT
  ex: corrupt 10%
- reorder PERCENT
  ex: reorder 10%
- rate RATE
  ex: rate 64kbps
--- FILTER OPERATION ---
- 0/i/a Select network
  (0:eth0/1:eth1/a:all)
- edit Show editing settings
- update Write filter settings
- reset Reset interface settings
--- PACKET CAPTURE ---
- ls show usb file list
- cap start Start packet capture
- cap stop Stop packet capture
--- OPERATION ---
- ifconfig [0] Show ifconfig
- clear [0] Clear screen
- help [1] Show command usage
  ex: help command
- stat [2] Reload I/O Stat
- iftop [3/4] Show iftop
- tcpdump [3/4] Show tcpdump
- height [3/4] height(10-30)
--- SETTING ---
- rtc Set hardware clock

```

図 3.4: ヘルプ画面 (コマンド一覧)

「0: メイン画面」で `help` コマンド名 と入力すると、そのコマンドのヘルプを表示します。

```
a > help rate
```

```

1 "HELP"
rate - Filter Option

Control outband traffic rate.

USAGE:
  rate speed

OPTIONS:
  speed Traffic rate.
  ex: 100Mbit, 64kbps
    - (default) bit / second
    - bit      bit / second
    - bps      byte / second

    - K      Killo
    - M      Mega
    - T      Giga

EXAMPLES:
0 > rate 100Mbit ... set rate
0* > update ... Write to NIC
--- 22:24:43 ---
[0] eth0: qdisc netem 8009: root
  refcnt 6 limit 1000 rate 100Mbit
  Sent 182 bytes 3 pkt (dropped 0,
  overlimits 0 backlog 0b 0p
  requeues 0

NOTICE:
  Execute update command to reflect
  on the NIC.

```

図 3.5: ヘルプ画面 (help rate)

注釈: 使用可能なコマンドは [コマンド一覧](#) を参照してください。

3.4 通信量画面 (2:I/O Stat)

ネットワークデバイス (eth0, eth1) の各送受信量を 1 秒毎に更新します。

```

2 "I/O Stat"
>>dstat -t --net -N eth0,eth1
----system---- --net/eth0----net/eth1-
   time      | recv  send:  recv  send
24-04 18:20:46 |    0    0 :    0    0
24-04 18:20:47 |  86B  78B :  64B  86B
24-04 18:20:48 | 346B   0 :    0  60B
24-04 18:20:49 |1278B 54B :    0  54B
24-04 18:20:50 |1310B   0 :    0    0
24-04 18:20:51 |1218B   0 :    0    0
24-04 18:20:52 | 450B   0 :    0    0
24-04 18:20:53 | 746B   0 :    0  60B
24-04 18:20:54 | 384B   0 :    0    0

```

図 3.6: 通信量画面

時刻, eth0(受信, 送信), eth1(受信, 送信) の並びで表示します。また、通信量によって数字の色が変化します。

表 3.4: 通信量の色

色	単位
赤	Byte
黄	Kbyte
緑	Mbyte

注釈: 起動時にネットワークデバイスが非アクティブ(=接続されていない) 場合、起動後にネットワークに接続しても通信量が表示されません。その際は、「0: メイン画面」で `stat` コマンドを実行して下さい。

3.5 eth0 画面 (3:eth0) / eth1 画面 (4:eth1)

ネットワークデバイスの通信状態 (iftop, tcpdump のいずれか) を表示します。

- iftop

TCP コネクション単位での帯域を表示します。

- tcpdump

ネットワークパケットを表示します。

3.5.1 表示切り替え

「0: メイン画面」からコマンド入力で表示を切り替えます。

(1) iftop

3 "eth0"										4 "eth1"									
		12.5kb	25.0kb	37.5kb	50.0kb	62.5kb						12.5kb	25.0kb	37.5kb	50.0kb	62.5kb			
192.168.100.233	=>	192.168.100.176	0b	0b	0b		192.168.100.176	=>	224.0.0.22	160b	40b	40b							
	<=		3.78kb	4.39kb	4.39kb			<=		0b	0b	0b							
192.168.100.176	=>	224.0.0.22	160b	40b	40b		0.0.0.0	=>	224.0.0.22	0b	40b	40b							
	<=		0b	0b	0b			<=		0b	0b	0b							
224.0.0.22	=>	0.0.0.0	0b	0b	0b		192.168.100.1	=>	224.0.0.1	0b	36b	36b							
	<=		0b	40b	40b			<=		0b	0b	0b							
192.168.100.1	=>	224.0.0.1	0b	36b	36b														
	<=		0b	0b	0b														
TX: cum: 0b peak: 0b rates: 0b 0b 0b										TX: cum: 40b peak: 160b rates: 0b 40b 40b									
RX: 4.51kb 8.61kb 3.94kb 4.51kb 4.51kb										RX: 76b 160b 160b 76b 76b									
TOTAL: 4.51kb 8.61kb 3.94kb 4.51kb 4.51kb										TOTAL: 116b 160b 160b 116b 116b									

図 3.7: iftop のみ

手順:

```
0 > a      ... 両方のデバイスを選択
a > iftop  ... 表示切り替え
```

(2) tcpdump

-3 "eth0"	-4 "eth1"
16:53:13.799679 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 60	16:52:55.998018 IP 0.0.0.0 > igmp.mcast.net: igmp
16:53:13.806659 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 0	16:52:57.894105 IP 192.168.100.100 > igmp.mcast.net: igmp
16:53:13.983251 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 120	16:52:58.014130 IP 192.168.100.233 > igmp.mcast.net: igmp
16:53:13.992397 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 0	16:52:59.902907 IP 192.168.100.162.9999 > 255.255.255.255.9999: UDP, length 29
16:53:14.435622 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 60	16:53:09.540773 IP 192.168.100.1 > all-systems.mcast.net: igmp
16:53:14.435708 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 60	16:53:09.843707 IP6 fe80::2ae9:8eff:fedc:9631 > 2404:7a84:62e0:1d00:47b7:e188:2
16:53:14.443053 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 0	8a6:e266: ICMP6, neighbor solicitation, who has 2404:7a84:62e0:1d00:47b7:e188:2
16:53:14.601533 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 120	8a6:e266, length 32
16:53:14.607800 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 0	16:53:09.844965 IP6 2404:7a84:62e0:1d00:47b7:e188:28a6:e266 > fe80::2ae9:8eff:f
16:53:14.973391 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 60	edc:9631: ICMP6, neighbor advertisement, tgt is 2404:7a84:62e0:1d00:47b7:e188:2
16:53:14.973471 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 60	8a6:e266, length 24
16:53:14.981976 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 0	16:53:13.349440 IP 192.168.100.162.mdns > 224.0.0.251.mdns: UDP, length 40
16:53:14.993549 IP6 fe80::43c3:ceb:2cde:7b16 > fe80::2ae9:8eff:fedc:9631: ICMP6,	16:53:13.358221 IP 192.168.100.162.mdns > 224.0.0.251.mdns: UDP, length 398
neighbor solicitation, who has fe80::2ae9:8eff:fedc:9631, length 32	16:53:13.374044 IP 0.0.0.0 > igmp.mcast.net: igmp
16:53:14.993667 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 0	16:53:14.993587 IP6 fe80::43c3:ceb:2cde:7b16 > fe80::2ae9:8eff:fedc:9631: ICMP6
16:53:14.995256 IP6 fe80::2ae9:8eff:fedc:9631 > fe80::43c3:ceb:2cde:7b16: ICMP6,	neighbor solicitation, who has fe80::2ae9:8eff:fedc:9631, length 32
neighbor advertisement, tgt is fe80::2ae9:8eff:fedc:9631, length 24	16:53:14.995282 IP6 fe80::2ae9:8eff:fedc:9631 > fe80::43c3:ceb:2cde:7b16: ICMP6
16:53:15.102071 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 84	neighbor advertisement, tgt is fe80::2ae9:8eff:fedc:9631, length 24
16:53:15.145835 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 0	

図 3.8: tcpdump のみ

手順:

```
0 > a          ... 両方のデバイスを選択
a > tcpdump    ... 表示切り替え
```

(3) 混在

左 (eth0), 右 (eth1) で異なる表示にすることも可能です。

-3 "eth0"	-4 "eth1"
12.5Kb	46
25.0Kb	17:22:27.186885 ARP, Request who-has 192.168.100.201 tell 192.168.100.1, length
37.5Kb	46
50.0Kb	17:22:27.206979 ARP, Request who-has 192.168.100.240 tell 192.168.100.1, length
62.5Kb	46
192.168.100.233	=> 192.168.100.100 0b 0b 0b 0b 1.24Kb
224.0.0.251	=> 192.168.100.162 0b 0b 0b 0b 0b
239.255.255.250	<= 192.168.100.122 0b 470b 117b 17:22:27.227098 ARP, Request who-has 192.168.100.221 tell 192.168.100.1, length
224.0.0.250	=> 192.168.100.162 0b 0b 0b 0b 46
192.168.100.1	<= 192.168.100.122 0b 367b 367b 17:22:27.247197 ARP, Request who-has 192.168.100.202 tell 192.168.100.1, length
224.0.0.251	=> 192.168.100.123 0b 0b 0b 0b 46
224.0.0.251	<= 192.168.100.123 0b 0b 0b 0b 15b 17:22:27.267220 ARP, Request who-has 192.168.100.103 tell 192.168.100.1, length
224.0.0.251	=> 192.168.100.4 0b 0b 0b 0b 1.16Kb 17:22:27.287322 ARP, Request who-has 192.168.0.11 tell 192.168.100.1, length 46
	<= 192.168.100.4 0b 0b 0b 0b 305b 17:22:27.307456 ARP, Request who-has 192.168.100.154 tell 192.168.100.1, length
	=> 192.168.100.4 0b 0b 0b 0b 46 17:22:27.327544 ARP, Request who-has 192.168.100.155 tell 192.168.100.1, length
	46
TX:	cum: 0b peak: 0b rates: 0b 0b 0b 17:22:28.542896 IP 192.168.100.162.mdns > 224.0.0.251.mdns: UDP, length 40
RX:	326KB 14.7Kb 2.91Kb 4.07Kb 3.89Kb 17:22:28.543135 IP 192.168.100.162.mdns > 224.0.0.251.mdns: UDP, length 214
TOTAL:	326KB 14.7Kb 2.91Kb 4.07Kb 3.89Kb 17:22:29.585189 IP 192.168.100.1 > all-systems.mcast.net: igmp

図 3.9: iftop(左:eth0), tcpdump(右:eth1)

手順:

```
a > 0          ... eth0 を選択
0 > iftop      ... eth0 の表示切り替え
0 > 1          ... eth1 を選択
1 > tcpdump    ... eth1 の表示切り替え
```

3.5.2 表示サイズ変更

「0: メイン画面」から **height** コマンドで表示サイズ (行数) を変更できます。

手順:

```
0 > height 20      ... 10 - 35 に範囲で指定
```

注釈: デバイスの選択状態に関係なく、画面 3,4 同時に変更されます。

第4章 コマンド

4.1 コマンド一覧

本装置で使用可能なコマンドは以下の通りです。コマンドは「0: メイン画面」から入力します。

表 4.1: 一般操作

コマンド	機能	引数	対象画面
ifconfig	ネットワークデバイスの設定表示	---	0
clear	表示クリア	---	0
help	ヘルプを表示する	コマンド	1
stat	I/O Stat の再起動	---	2
iftop	iftop を表示する	---	3, 4
tcpdump	tcpdump を表示する	---	3, 4
height	画面 3,4 の表示行数を設定する	行数	3, 4

表 4.2: フィルター操作

コマンド	機能	引数
0 / 1 / a	設定対象のネットワークデバイス選択 0:eth0 / 1:eth1 / a:all(eth0 & eth1)	---
edit	未反映のフィルター設定表示	---
update	フィルター設定適用	---
reset	フィルター設定解除	---

表 4.3: フィルター設定

コマンド	機能	引数
delay	遅延	遅延時間, {ばらつき}, {発生率 (%)}
loss	損失 (パケットロス)	損失率 (%)
duplicate	重複	発生率 (%)
corrupt	破損	発生率 (%)
reorder	順序変更	発生率 (%)
rate	通信速度	帯域 (速度)

表 4.4: パケットキャプチャ

コマンド	機能	引数
ls	ファイル一覧表示 (USB メモリ)	---
cap start	キャプチャ開始	ローテート間隔 (秒)
cap stop	キャプチャ停止	---

表 4.5: システム設定

コマンド	機能	引数
rtc	システム時刻設定	設定時刻 (YYYY-MM-DD hh:mm:ss)
keyboard	キーボードレイアウト変更	レイアウト (us / jp / hhk)

注釈: 次節以降で各コマンドの使用方法、仕様について説明します。

4.2 一般操作

4.2.1 ifconfig: ネットワーク設定表示

選択中のネットワークデバイス (eth0, eth1) の設定を表示します。

表 4.6: ifconfig で表示するデバイス

カーソル	表示内容
0 >	eth0 のみ表示
1 >	eth1 のみ表示
a >	eth0, eth1 両方を表示

書式:

```
> ifconfig
```

実行例:

```
a > ifconfig
eth0: flags=4419<UP,BROADCAST,RUNNING,PROMISC,MULTICAST> mtu 1500
    inet6 fe80::dea6:32ff:fe91:b629 prefixlen 64 scopeid 0x20<link>
    ether dc:a6:32:91:b6:29 txqueuelen 1000 (Ethernet)
    RX packets 35014 bytes 14800637 (14.1 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 6140 bytes 676011 (660.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4419<UP,BROADCAST,RUNNING,PROMISC,MULTICAST> mtu 1500
    inet6 fe80::2e0:4cff:fe01:5350 prefixlen 64 scopeid 0x20<link>
    ether 00:e0:4c:01:53:50 txqueuelen 1000 (Ethernet)
    RX packets 4536 bytes 489618 (478.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 23361 bytes 11606631 (11.0 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

図 4.1: ifconfig 出力例

4.2.2 clear: メイン画面のクリア

「0: メイン画面」を表示クリアします。

書式:

```
> clear
```

4.2.3 help: ヘルプ表示

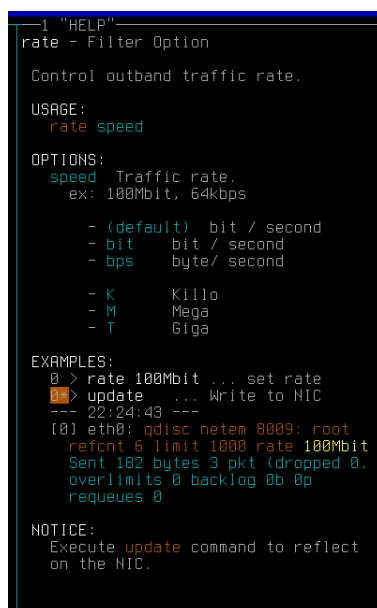
ヘルプ表示します。ヘルプは「1: ヘルプ画面」に表示します。

書式:

```
> help {コマンド名}
```

実行例:

```
> help rate
```



```
1 "HELP"
rate - Filter Option

Control outband traffic rate.

USAGE:
  rate speed

OPTIONS:
  speed Traffic rate.
  ex: 100Mbit, 64kbps

  - (default) bit / second
  - bit       bit / second
  - bps       byte/ second

  - K        Killo
  - M        Mega
  - T        Giga

EXAMPLES:
  0 > rate 100Mbit ... set rate
  0* > update ... Write to NIC
  --- 22:24:43 ---
  [0] eth0: qdisc netem 8009: root
        refcnt 6 limit 1000 rate 100Mbit
        Sent 182 bytes 3 pkt (dropped 0,
        overlimits 0 backlog 0b 0p
        requeues 0

NOTICE:
  Execute update command to reflect
  on the NIC.
```

図 4.2: ヘルプ画面 (help rate)

4.2.4 stat: 通信量画面の再起動

「2: 通信量画面」を再起動します。

書式:

```
> stat
```

注釈: 起動時にネットワークデバイスが非アクティブ (=接続されていない) 場合、起動後にネットワークに接続しても通信量が表示されません。その際は、本コマンドを実行して下さい。

4.2.5 iftop

「3:eth0 画面」, 「4:eth1 画面」の表示を iftop コマンドに切り替えます。

書式:

```
> iftop
```

--3 "eth0"										--4 "eth1"									
	12.5Kb		25.0Kb		37.5Kb		50.0Kb		62.5Kb		12.5Kb		25.0Kb		37.5Kb		50.0Kb		62.5Kb
192.168.100.233		=>	192.168.100.176		0b		0b		0b	192.168.100.176		=>	224.0.0.22		160b		40b		40b
		<=			3.78Kb		4.39Kb		4.39Kb			<=			0b		0b		0b
192.168.100.176		=>	224.0.0.22		160b		40b		40b	0.0.0.0		=>	224.0.0.22		0b		40b		40b
		<=			0b		0b		0b			<=			0b		0b		0b
224.0.0.22		=>	0.0.0.0		0b		0b		0b	192.168.100.1		=>	224.0.0.1		0b		36b		36b
		<=			0b		40b		40b			<=			0b		0b		0b
192.168.100.1		=>	224.0.0.1		0b		36b		36b										
		<=			0b		0b		0b										
TX: cum: 0b peak: 0b rates: 0b 0b 0b										TX: cum: 40b peak: 160b rates: 0b 40b 40b									
RX: 4.51Kb 8.61Kb 3.94Kb 4.51Kb 4.51Kb										RX: 76b 160b 160b 76b 76b									
TOTAL: 4.51Kb 8.61Kb 3.94Kb 4.51Kb 4.51Kb										TOTAL: 116b 160b 160b 116b 116b									

図 4.3: iftop

4.2.6 tcpdump

「3:eth0 画面」, 「4:eth1 画面」の表示を tcpdump コマンドに切り替えます。

選択中のデバイスのみ表示を切り替えます。

書式:

```
> tcpdump
```

--3 "eth0"		--4 "eth1"	
16:53:13.799679 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 60		16:52:56.990018 IP 0.0.0.0 > igmp.mcast.net: igmp	
16:53:13.806659 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 0		16:52:57.094105 IP 192.168.100.100 > igmp.mcast.net: igmp	
16:53:13.983251 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 120		16:52:58.014130 IP 192.168.100.233 > igmp.mcast.net: igmp	
16:53:13.992397 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 0		16:52:59.902987 IP 192.168.100.162.9999 > 255.255.255.255.9999: UDP, length 29	
16:53:14.435622 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 60		16:53:09.540773 IP 192.168.100.1 > all-systems.mcast.net: igmp	
16:53:14.443053 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 0		16:53:09.843707 IP6 fe80::2ae9:8eff:fedc:9631 > 2404:7a84:62e0:1d00:47b7:e188:28a6:e266: ICMP6, neighbor solicitation, who has 2404:7a84:62e0:1d00:47b7:e188:28a6:e266, length 32	
16:53:14.601533 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 120		16:53:09.844955 IP6 2404:7a84:62e0:1d00:47b7:e188:28a6:e266 > fe80::2ae9:8eff:fedc:9631: ICMP6, neighbor advertisement, tgt is fe80::2ae9:8eff:fedc:9631, length 32	
16:53:14.607880 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 0		16:53:13.350221 IP 192.168.100.162.mdns > 224.0.0.251.mdns: UDP, length 398	
16:53:14.973391 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 60		16:53:13.374044 IP 0.0.0.0 > igmp.mcast.net: igmp	
16:53:14.979471 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 0		16:53:14.993507 IP6 fe80::2ae9:8eff:fedc:9631 > fe80::43c3:ceb:2cde:7b16: ICMP6, neighbor solicitation, who has fe80::2ae9:8eff:fedc:9631, length 32	
16:53:14.993549 IP6 fe80::43c3:ceb:2cde:7b16 > fe80::2ae9:8eff:fedc:9631: ICMP6, neighbor solicitation, who has fe80::2ae9:8eff:fedc:9631, length 32		16:53:14.995256 IP6 fe80::2ae9:8eff:fedc:9631 > fe80::43c3:ceb:2cde:7b16: ICMP6, neighbor advertisement, tgt is fe80::2ae9:8eff:fedc:9631, length 24	
16:53:14.993667 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 0		16:53:15.102071 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 0	
16:53:14.995256 IP6 fe80::2ae9:8eff:fedc:9631 > fe80::43c3:ceb:2cde:7b16: ICMP6, neighbor advertisement, tgt is fe80::2ae9:8eff:fedc:9631, length 24		16:53:15.145835 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 0	
16:53:15.102071 IP 192.168.100.100.50064 > 192.168.100.233.ssh: tcp 0			

図 4.4: tcpdump

4.2.7 height

「3:eth0 画面」, 「4:eth1 画面」の表示行数を変更します。

選択中のデバイスのみ表示を切り替えます。

書式:

```
> hegith 行数
```

表 4.7: height コマンドの引数

引数	省略	値
行数	必須	10 - 35

実行例:

```
> height 20
```

4.3 フィルター設定

4.3.1 フィルターの適用範囲

フィルター設定はネットワークデバイスの出力 (送信側) に対して適用されます。

例えば、遅延オプションを片方のデバイスにしか設定しない場合、一方向の packets にのみ遅延が発生し、逆方向の packets には遅延が発生しません。

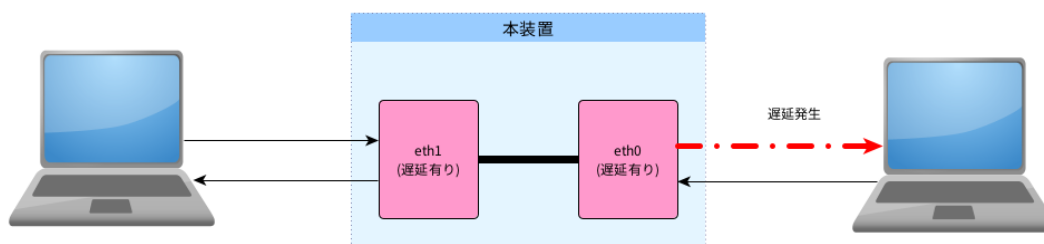


図 4.5: 片方向の遅延

そのため、双方向に遅延を発生させる場合は、両デバイスに対して同じフィルター設定を行って下さい。

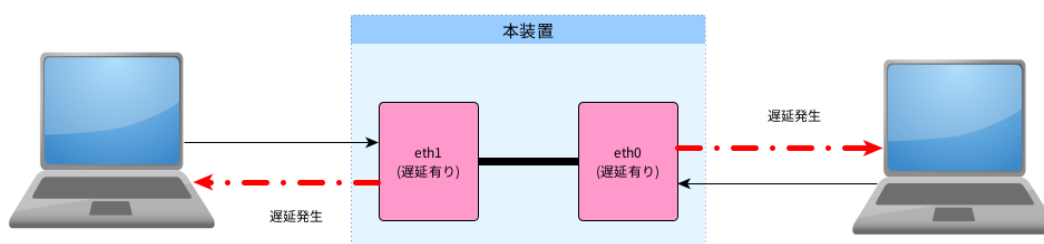


図 4.6: 双方向の遅延

4.3.2 編集

4.3.3 編集コマンド

フィルター・オプションは一旦編集バッファに格納され、*edit* コマンドで編集中の内容を確認できます。*update* コマンドを実行するとフィルター・設定が変更されます。また、*reset* コマンドで解除されます。

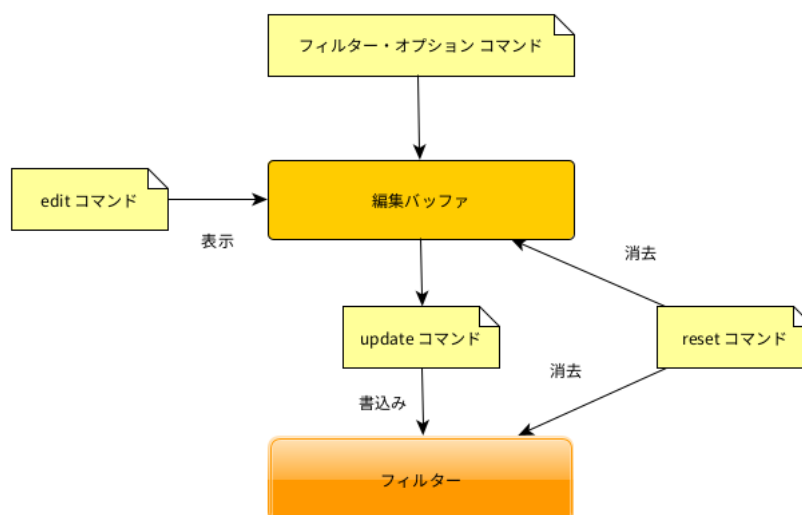


図 4.7: 編集コマンド

(1) edit

編集集中でフィルター未適用の内容を表示します。

```
a > rate 64kbps ... 設定
a* > edit ... 確認
[0] eth0: delay:- loss:- duplicate:- corrupt:- reorder:- rate:['64kbps'] 編集バッファ
[1] eth1: delay:- loss:- duplicate:- corrupt:- reorder:- rate:['64kbps'] の内容
```

図 4.8: edit コマンド

(2) update

編集中の内容をフィルターに適用します。

```
a*> update
execute: sudo tc qdisc change dev eth0 root netem rate 64kbps
execute: sudo tc qdisc change dev eth1 root netem rate 64kbps
--- 15:06:40 ---
[0] eth0:  qdisc netem 8003: root refcnt 6 limit 1000 rate 512Kbit
        Sent 2784 bytes 34 pkt (dropped 0, overlimits 0 requeues 0)
        backlog 0b 0p requeues 0
[1] eth1:  qdisc netem 8004: root refcnt 2 limit 1000 rate 512Kbit
        Sent 46830 bytes 264 pkt (dropped 0, overlimits 0 requeues 0)
        backlog 0b 0p requeues 0
```

図 4.9: update コマンド

上記の例では、帯域が 64Kpbs(=512Kbit/second) に設定されています。

(3) reset

適用済み、及び、編集中のフィルタ設定をリセットします。

```
a > reset
--- 15:06:43 ---
[0] eth0:  qdisc netem 8005: root refcnt 6 limit 1000
        Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
        backlog 0b 0p requeues 0
[1] eth1:  qdisc netem 8006: root refcnt 2 limit 1000
        Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
        backlog 0b 0p requeues 0
a > _
```

図 4.10: reset コマンド

4.3.4 フィルター・オプション

フィルター・オプションを入力すると一旦編集バッファに格納されます。*update* コマンドを実行するとフィルター設定に反映されます。(前節参照)

(1) delay: 遅延

パケットの遅延設定を行います。

書式:

```
> delay 遅延時間 {ばらつき} {発生率 (%)}
```

表 4.8: delay オプションの引数

引数	省略	値
遅延時間	必須	単位: s: 秒 / ms: ミリ秒 / us: マイクロ秒
ばらつき時間	可	単位: s: 秒 / ms: ミリ秒 / us: マイクロ秒
ばらつき発生率	可	単位: %

実行例:

- a. 100msec の遅延を発生させる。

```
> delay 100ms
```

- b. 100msec±20msec の遅延を発生させる。

```
> delay 100ms 20ms
```

- c. 100msec±20msec の遅延を 50%、残りは 100msec の遅延を発生させる。

```
> delay 100ms 20ms 50%
```

注意:

ping コマンドのようにパケット往復が発生する通信の場合、フィルターの適用範囲によって全体の遅延時間に違いが発生します。

a. 双方向に遅延を設定した場合

```
a > delay 100ms
```

往復で 200ms の遅延が発生します。

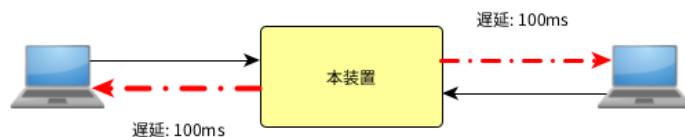


図 4.11: 両方向の遅延

往復で 100ms にする場合は、50ms の遅延設定にしてください。

b. 片方向に遅延を設定した場合

```
1 > delay 100ms
```

片方向のみで 100ms の遅延が発生します。

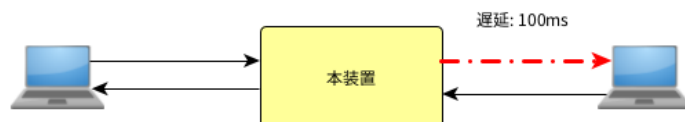


図 4.12: 片方向の遅延

通信量が偏っている場合:

サーバからのダウンロードのように片側の通信量が大きい場合はフィルターの設定方向に注意して下さい。

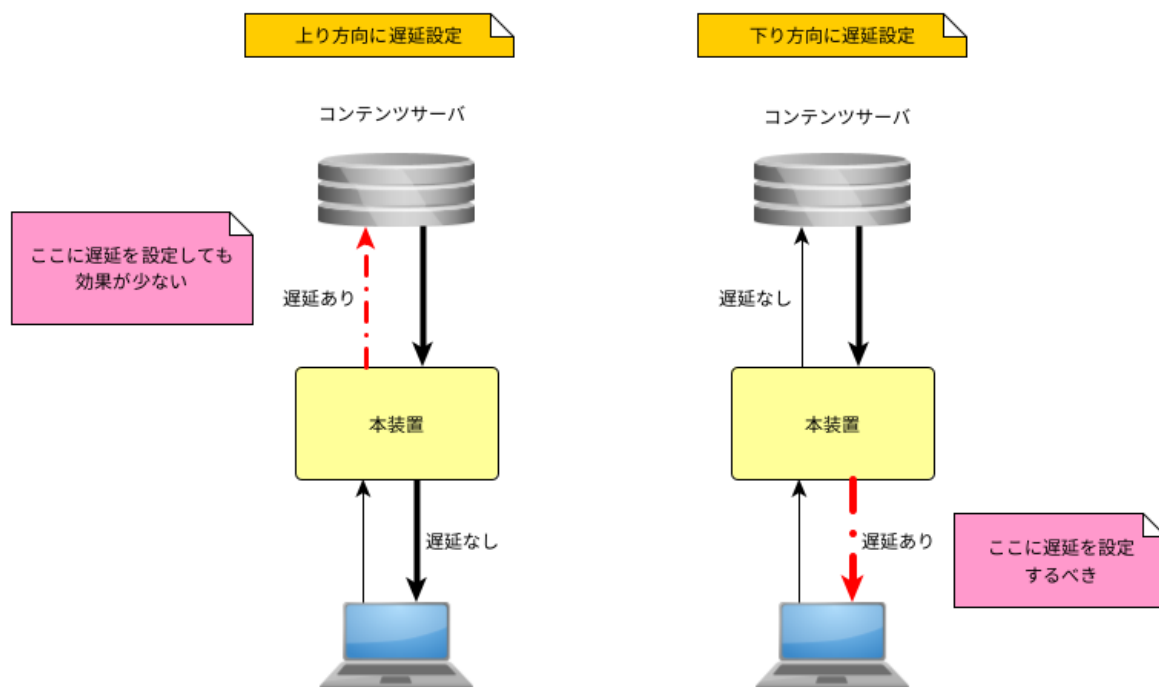


図 4.13: 通信量が偏っている例

(2) loss: 損失 (パケットロス)

パケットロスを発生させます。

書式:

```
> loss 損失率 (%)
```

表 4.9: loss オプションの引数

引数	省略	値
損失率	必須	単位: %

実行例:

- a. 10% の確率でパケットロスする。

```
> loss 10%
```

注釈: 遅延時間と同様にフィルターの適用範囲に注意して下さい。

(3) duplicate: 重複

パケットの重複を発生させます。

書式:

```
> duplicate 発生率 (%)
```

表 4.10: duplicate オプションの引数

引数	省略	値
発生率	必須	単位: %

実行例:

- a. 10% の確率でパケットを重複させる。

```
> duplicate 10%
```

注釈: 遅延時間と同様にフィルターの適用範囲に注意して下さい。

(4) corrupt: 破損

パケットのデータを書き換えます (ランダムな 1 bit を反転)

書式:

```
> corrupt 発生率 (%)
```

表 4.11: corrupt オプションの引数

引数	省略	値
発生率	必須	単位: %

実行例:

- a. 0.1% の確率でパケットを書き換える。

```
> corrupt 0.1%
```

注釈:

1. 遅延時間と同様にフィルターの適用範囲に注意して下さい。
2. TCP/IP パケットの場合はプロトコル・スタックで CRC チェックで誤りを検出してパケットは破棄されます。UDP パケットでチェックサムを無効にしている場合は上位層で誤り検出が必要です。

(5) reorder: 順序変更

パケットの到着順序を変更します。

書式:

```
> reorder 発生率 (%)
```

表 4.12: reorder オプションの引数

引数	省略	値
発生率	必須	単位: %

実行例:

- a. 10% の確率でパケットの到達順序を変更する。

```
> reorder 10%
```

注釈: 遅延時間と同様にフィルターの適用範囲に注意して下さい。

(6) rate: 帯域制御

書式:

```
> rate 帯域
```

表 4.13: rate オプションの引数

引数	省略	値
帯域	必須	単位

表 4.14: 単位

単位	意味
bps	byte/second
bit	bit/second
指定なし	bit/second
K / k	Killo (1,000)
M / m	Mega (1,000,000)
G / g	Giga (1,000,000,000)

表 4.15: 設定例

例	解釈
1000	1K bit/second
1K	1K bit/second
1Kbit	1K bit/second
1Kbps	1K byte/second

実行例:

```
> rate 64kbps
```

注釈: 遅延時間と同様にフィルターの適用範囲に注意して下さい。

4.4 パケットキャプチャ

注釈:

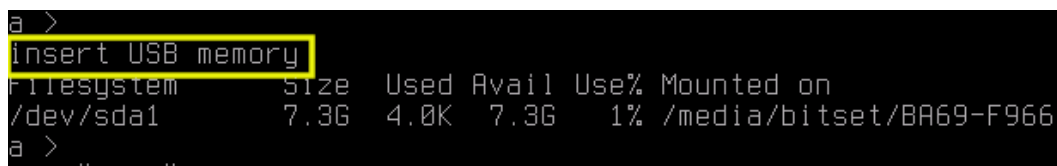
1. パケットキャプチャ機能を使用するには USB メモリが必要です。(USB3.0 対応推奨)
 2. 結果の確認は PC で Wireshark を使用します。
-

4.4.1 キャプチャ手順

(1) 開始

1. USB メモリを本体の USB3.0 ポートに挿入します。

USB メモリを認識すると、*insert USB memory* というメッセージが表示されます。



```
a >
insert USB memory
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1        7.3G  4.0K  7.3G   1% /media/bitset/BA69-F966
a >
```

図 4.14: USB メモリ挿入時のメッセージ

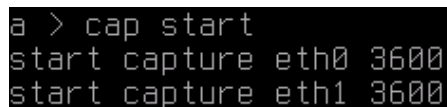
注釈: USB メモリに十分な空き領域があることを確認してください。

2. メイン画面でキャプチャ開始コマンドを入力します。

書式:

```
> cap start {ローテート間隔 (秒)}
```

デフォルトでは 3600 秒 (1 時間) 毎にファイルをローテートします。間隔を変更する場合は引数で指定します。



```
a > cap start
start capture eth0 3600
start capture eth1 3600
```

図 4.15: キャプチャ開始

上記の例では、eth0, eth1 両方のキャプチャを開始しています。

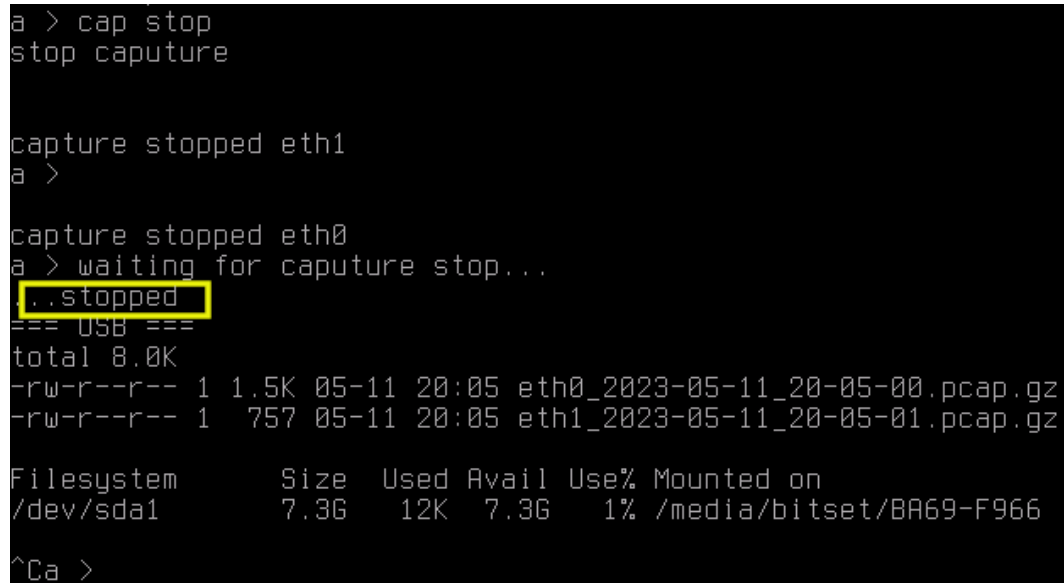
(2) 停止

1. メイン画面でキャプチャ停止コマンドを入力します。

書式:

```
> cap stop
```

2. キャプチャ結果のファイル保存が終わるまで待ちます。



```
a > cap stop
stop caputure

capture stopped eth1
a >

capture stopped eth0
a > waiting for caputure stop...
..stopped
=== USB ===
total 8.0K
-rw-r--r-- 1 1.5K 05-11 20:05 eth0_2023-05-11_20-05-00.pcap.gz
-rw-r--r-- 1 757 05-11 20:05 eth1_2023-05-11_20-05-01.pcap.gz

Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       7.3G   12K   7.3G   1% /media/bitset/BA69-F966

^Ca >
```

図 4.16: キャプチャ停止

`stopped` の後に、USB メモリ内のファイル一覧表示すれば処理完了です。

3. USB メモリを取り出します。

(3) 結果の確認

USB メモリに保存されているキャプチャファイル(*.gz)を Wireshark で開いて確認します。(gz ファイルの解凍は不要です)

ファイルは以下の命名規則で保存されています。

eth{0|1}_{YYYY-MM-DD_hh-mm-ss}.pcap.gz

1. Wireshark の [ファイル]-[開く] メニューでファイルを開きます。



図 4.17: ファイルを開く

2. キャプチャデータを表示します。

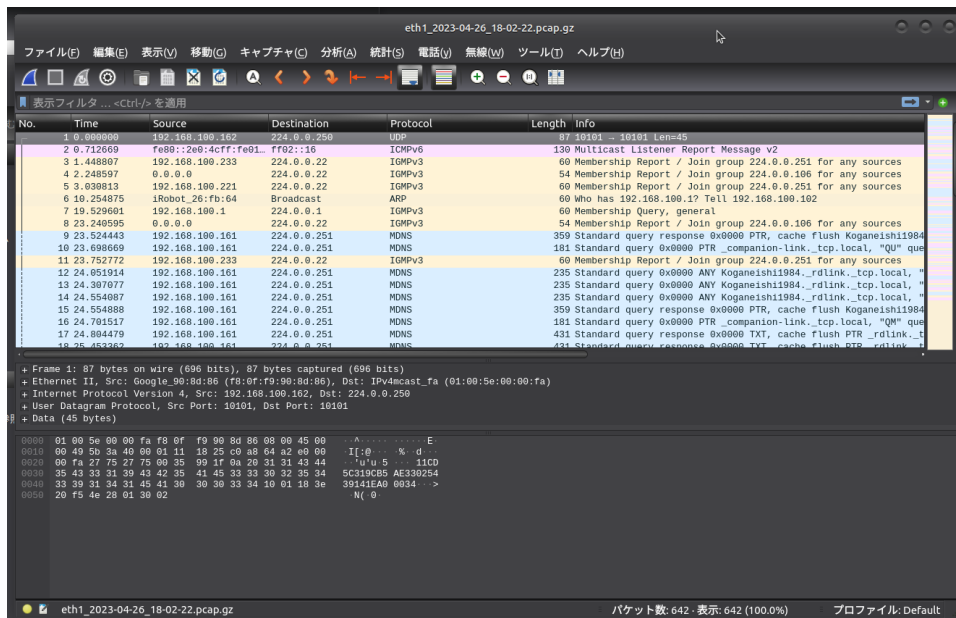


図 4.18: Wireshark での表示

4.4.2 ls: USB メモリのファイル一覧表示

書式:

```
> ls
```

実行例:

```
a >
insert USB memory
Filesystem      Size   Used Avail Use% Mounted on
/dev/sda1        7.3G    20K   7.3G   1% /media/bitset/BA69-F966
a > ls
=== USB ===
total 16K
-rw-r--r-- 1 3.0K 05-12 14:48 eth0_2023-05-12_14-48-03.pcap.gz
-rw-r--r-- 1 2.1K 05-12 14:48 eth1_2023-05-12_14-48-03.pcap.gz
-rw-r--r-- 1 1.5K 05-11 20:05 eth0_2023-05-11_20-05-00.pcap.gz
-rw-r--r-- 1 757 05-11 20:05 eth1_2023-05-11_20-05-01.pcap.gz

Filesystem      Size   Used Avail Use% Mounted on
/dev/sda1        7.3G    20K   7.3G   1% /media/bitset/BA69-F966
```

図 4.19: ls コマンド実行例

「*.pcap.gz」ファイルがパケットキャプチャした結果です。

4.4.3 cap start: キャプチャ開始

書式:

```
> cap start {ローテート間隔 (秒)}
```

表 4.16: cap start コマンドの引数

引数	省略	値
ローテート間隔	可	単位: 秒, 範囲:60 - 3600 (省略時は 3600)

4.4.4 cap stop: キャプチャ終了

書式:

```
> cap stop
```

コマンド実行後、未保存のデータを USB メモリに書き込むので、しばらくお待ち下さい。USB メモリのファイル一覧が表示すれば処理完了です。

4.5 システム設定

4.5.1 rtc: 時刻設定

システム時刻を設定します。時刻は RTC(リアルタイムクロック) に記録され、電源を切った状態でも更新します。

書式:

```
> rtc YYYY-MM-DD hh:mm:ss
```

実行例:

```
> rtc 2023-05-10 12:15:00
```

警告: 設定変更後、システムが再起動します。

4.5.2 keyboard: キーボードレイアウト変更

書式:

```
> keyboard {layout}
```

layout は以下のいずれかを指定して下さい。

表 4.17: layout 一覧

layout	言語	キーボードのタイプ
jp	日本語	PC-105
us	英語	PC-105
hhk	英語	PFU Happy Hacking keyboard

実行例:

```
> keyboard us
```

警告: 設定変更後、システムが再起動します。

第5章 補足

5.1 補足事項

5.1.1 RTC の電池交換

注釈:

- 電池は CR1225 を使用して下さい。
 - 交換は電源を切った状態で行って下さい。
-

交換手順:

1. 本体の上蓋をスライドさせて外します。
2. 電池を交換します。



図 5.1: RTC

3. 交換後は `rtc` コマンド で時刻を再設定して下さい。

5.2 改定履歴

表 5.1: 改定履歴

Version	日付	改定内容
0.1	2023-06-01	初版作成

5.3 問い合わせ先

表 5.2: 問い合わせ先

販売	株式会社 bitset
住所	569-1123 大阪府高槻市芥川町 2-14-7 ロイヤル高槻 2F
担当	宮村
mail	info@bitset.jp
tel	050-7119-2899
Web	https://bitset.jp
サポート時間帯	平日: 09:00～18:00 (土日祝日休み)

—以上