SASAT 操作説明書

0.1 版 2010 年 1 月 7 日

富士通株式会社 2010

改版履歴

版数	日付	修正者	内容
0.1	2010/01/07	八木	レビュー版

目次

1. 概要

1.1. 目的

本開発は、SASAT 方式による負荷分散ソフトウェアの試作開発である。
本試作装置を様々なプロトコルに適用してうまく負荷分散できるかどうかを調査し、SASAT 方式の実用性、優位性を実証することを目的とする。
本試作は商用を目的としたものではないため、負荷分散装置としての機能は限定される。
実証中に見つかると思われる問題点の修正や改善はその都度行っていく。

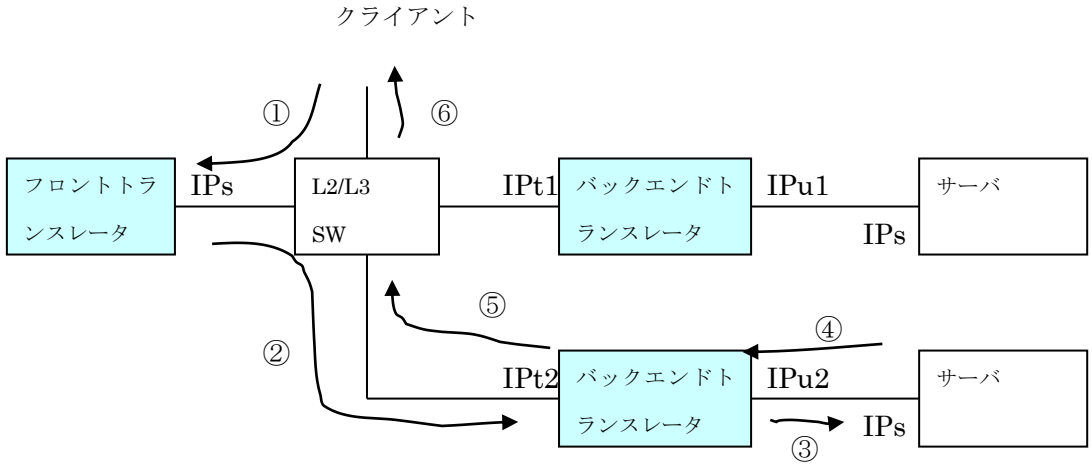
1.2. 参照資料

「SASAT 方式機能仕様書」富士通株式会社

1.3. 機能概要

SASAT 方式における負荷分散装置はフロントトランスレータとバックエンドトランスレータの 2 種類のトランスレータによって構成される。
各トランスレータの処理内容は、「SASAT 方式機能仕様書」を参照。

参考図



※①～⑥はパケットの流れ順を示す

本試作における負荷分散機能の概要を以下に示す。
＊試作のため機能は限定されているが、SASAT 方式による制限ではない。

項目	内容
負荷分散方式	クライアント IP による IP パケット静的分散方式（宛先 IP 書き換え）
サポート構成	ワンアーム配置、直接サーバ応答のみサポート
セッション数	セッション管理を行わないため、無制限

振り分け設定	振り分け設定ファイルによって行い、動作中の変更可能
サーバ死活監視	サポートしない
IPv4/v6	両サポート
ログ、統計機能	サポート
対象プロトコル	telnet, http, ftp, https, IPsec など
その他	ARP 要求/ND 応答機能 VLAN 未サポート IP マルチキャスト未サポート

1.4. システム構成

本試作でサポートするシステム構成について述べる。
今回の試作では、下表における“直接サーバ応答構成、ワンアーム配置の L2/L3 構成”（網掛け部）のみをサポートする。

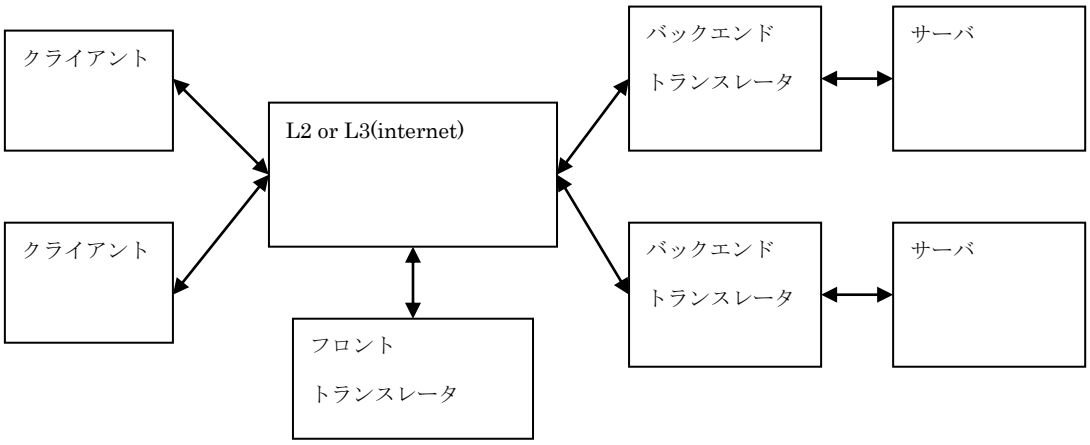
ワンアーム配置のため､フロントトランスレータのインターフェースは１つだけ使用する。
バックエンドトランスレータは 2 本のインターフェースを使用してサーバと外部とのブリッジを行う。

負荷分散装置構成一覧と本試作によるサポート範囲

代表 IP とサーバ IP のネットワーク	構成	配置	備考
異なる	ルート*1	ワンアーム	
		通過	IPCOM:通過型ルーターモード
	直接サーバ応答	ワンアーム	
同一	ブリッジ*2	ワンアーム	
		通過	IPCOM:通過型ブリッジモード
	ルート	ワンアーム	IPCOM:ワンアーム型デフォルト GW 設定
		通過	
	直接サーバ応答	ワンアーム	IPCOM:MAC アドレス変換と同等

*1 ルート構成 – 負荷分散装置がルータとして動作する
*2 ブリッジ構成 －負荷分散装置がブリッジとして動作する

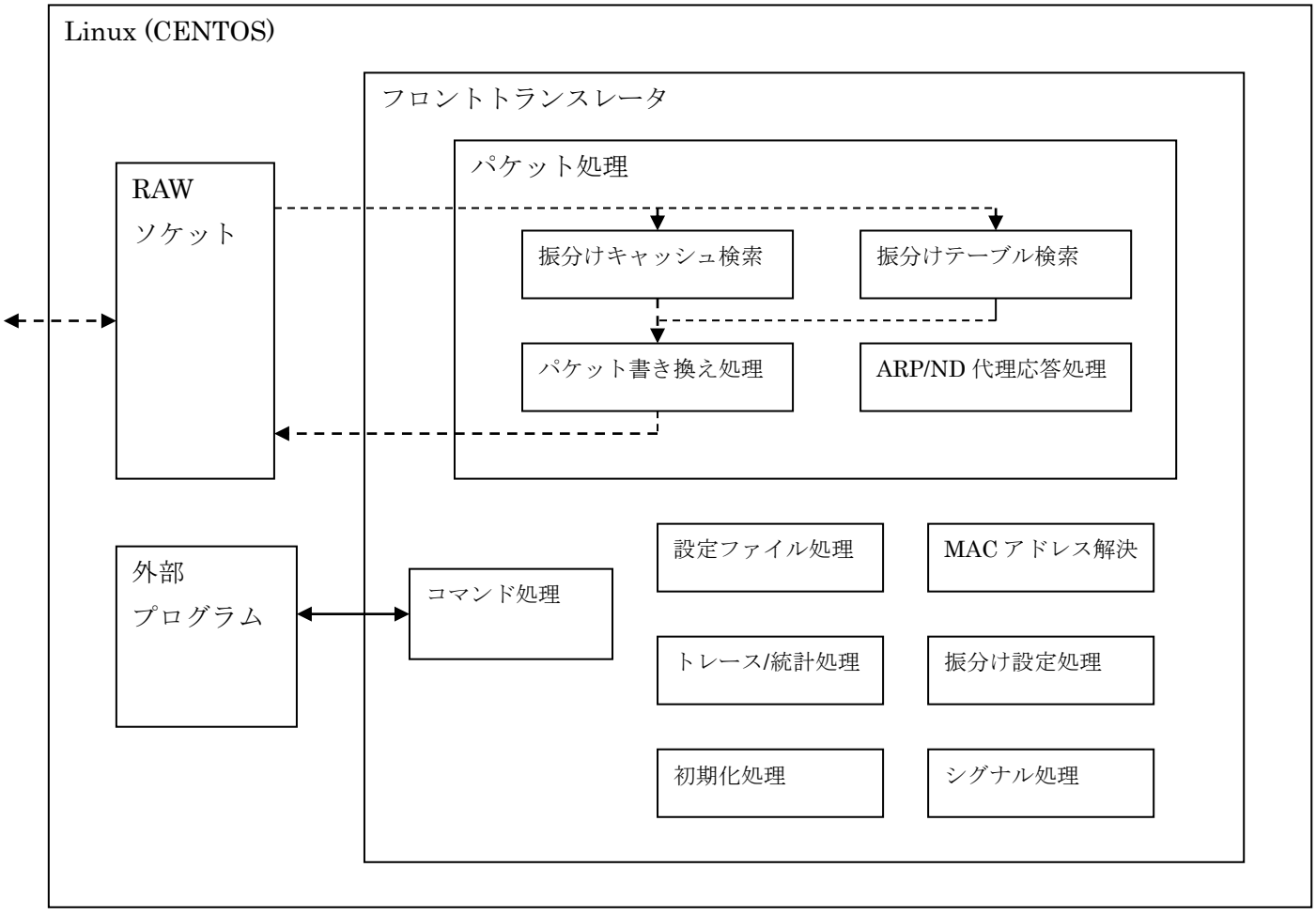
サポート構成



1.5. ソフトウェア構成

本開発におけるソフトウェア構成を以下に示す。

フロントトランスレータ



主な機能の概要

	機能名	概要
1	振分けテーブル検索処理	送信元 IP をキーに、振分け設定ファイルに記述された内容のとおりに振分け先を決定する処理。
2	振分けキャッシュ検索処理	振分けテーブル検索でヒットした場合に作成されるキャッシュファイルのハッシュ検索処理。　振分けテーブルの検索に先立ち行う。
3	パケット書き換え処理	振分けキャッシュテーブルを参照して、宛先 IP およびチェックサム値を書き換え、RAW ソケットへ送信する処理。
4	ARP/ND 代理応答処理	代表 IP が仮想 IP（実際に OS に振られたものではない）の場合、OS 機能の代わりに ARP 応答および ND（neighbor advertise）を代理で行う機能。　※本試作では、仮想 IP と実 IP の両方をサポートし、切り替えは動作設定ファイルの設定によって行う。
5	MAC アドレス解決処理	バックエンドトランスレータへ送信する場合の MAC アドレスを解決する処理。
6	振分け設定処理	振分け設定ファイルを読み込み、振分けテーブルおよびサーバ情報テーブルを作成、削除を行う処理
7	設定ファイル処理	各種動作設定を記述する動作設定ファイルを読み込み、動作モードを設定する処理
8	コマンド処理	外部プログラムからのコマンドを処理する機能。
9	シグナル処理	シグナル（HUP）を受信する処理（設定再読み込み用）

＊振分け検索処理について

代表 IP 宛ての IP パケットを受信したとき、振分け先を決定する処理を実行しパケットを書き換えてサーバ（バックエンドトランスレータ）へ送信する。

このとき検索する振分け先情報は、振分けテーブルと振分けキャッシュテーブルの 2 種類に分けられる。

振分けテーブルは、ユーザが用意する振分け設定ファイルの情報を保持するテーブルであり、優先順にリンクされている。

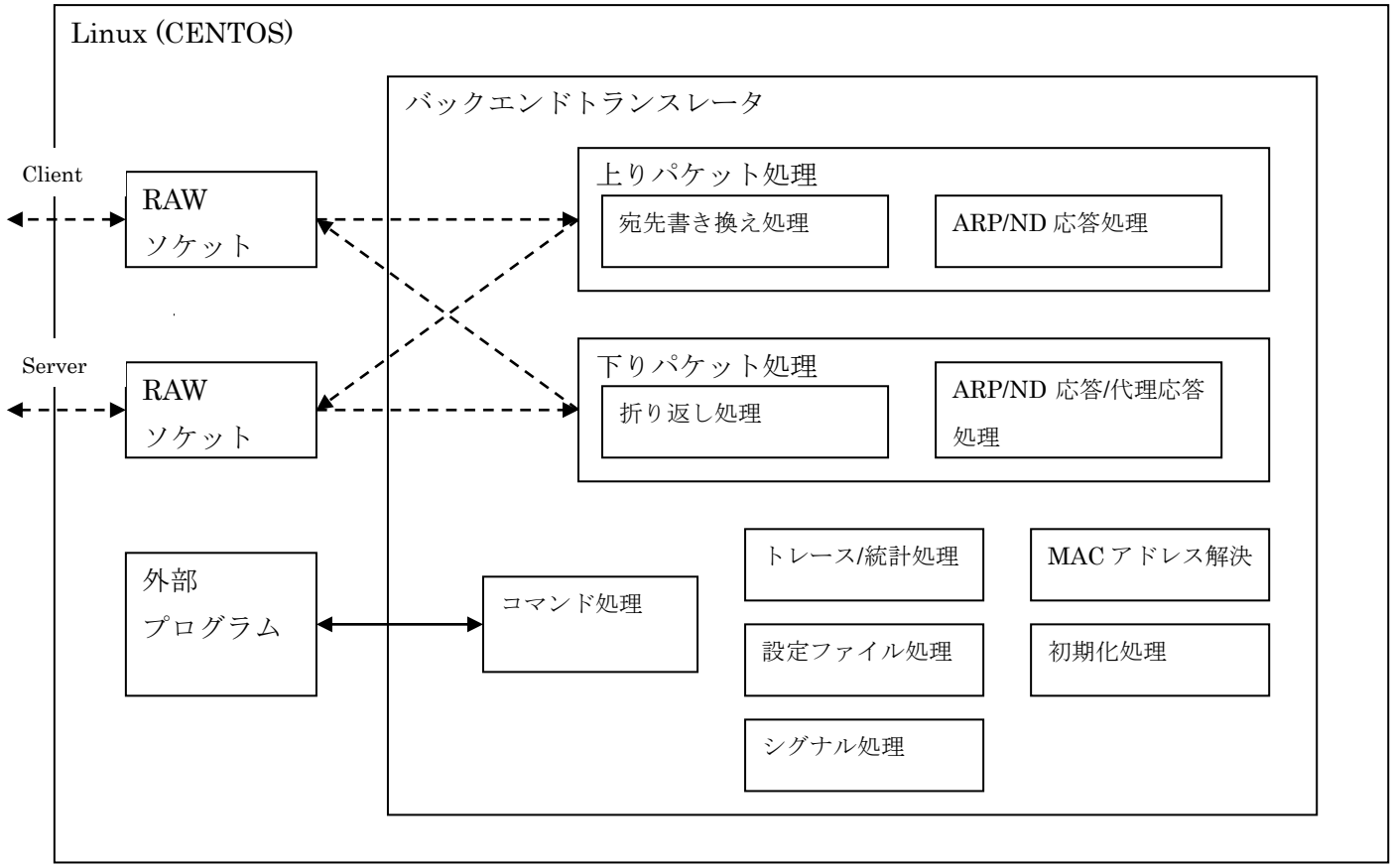
振分けキャッシュテーブルは、一度検索した振分け設定ファイルを高速に検索するために、クライアント毎の情報をハッシュに登録したものである。

＊仮想 IP モードと実 IP モードについて（実験用）

仮想 IP モードはトランスレータが使用する IP アドレスは OS に設定されなくトランスレータが仮想的に認識するアドレスである。　トランスレータが使用するアドレスは動作設定ファイルに記述される。

一方実 IP モードは OS に設定された IP アドレスを使用するもので、トランスレータは OS 設定を読み込んで自動的に認識する。

バックエンドトランスレータ



主な機能の概要

	機能名	概要
1	宛先書き換え処理	フロントトランスレータからの上り（クライアント->サーバ）パケットの宛先 IP,MAC アドレスをサーバのものに変換し、IP チェックサムを更新する処理。
2	折り返し処理	サーバからの下り（サーバ->クライアント）パケットの処理。
3	ARP/ND 応答処理（上り）	上りインターフェースの IP が仮想 IP の場合、OS 機能の代わりに ARP 応答および ND（neighbor advertise）を代理で行う機能。※本試作では、仮想 IP と実 IP の両方をサポートし、切り替えは設定ファイルの設定によって行う
4	ARP/ND 代理応答処理（下り）	サーバが下りパケット送信のための MAC アドレス解決を代理で行う機能。　サーバからの ARP/ND パケットを受信したら、代理で送信し、結果をサーバへ通知する。 また、上りと同じく ARP/ND 応答処理も行う。
5	MAC アドレ	サーバへパケットを送信する場合に MAC アドレスを解決する処理。

	ス解決処理	
6	動作設定ファイル処理	各種動作設定を記述するファイルを読み込み、動作モードを設定する処理
7	コマンド処理	外部プログラムからのコマンドを処理する機能。
8	シグナル処理	シグナル（HUP）を受信する処理（設定再読み込み用）

1.6. 動作環境

以下の環境で試作・検証を行う。

マシン：

FMV-LOOX N/D15 (CPU ATOM N280 1.6GHz, メモリ 1GB)

ハイパースレッディング ON

OS：

CENTOS5.4 (カーネル Linux 2.6.18-164.6.1.el5)

コンパイラ：

gcc バージョン 4.1.2　i386 版

2. トランスレータ操作手順

フロント、バックエンドトランスレータの起動や操作手順について記述する。

2.1. 事前処理および前提条件

2.1.1. OS 設定

・cpuspeed デーモンを停止する（以下コマンドを入力して恒久的に停止)。

```
# chkconfig --level 35 cpuspeed off
```

停止させる理由は、atom の SpeedStep 機能(EIST)が動作することで CPU の TSC 値（time stamp counter）のクロックが変化してしまい、ログ時刻が不正確になるため。

2.1.2. インターフェース設定

・トランスレータの各インターフェースに IP アドレスを設定する。
※仮想 IP モードを使用する場合にも実 IP アドレスが必要。

・iptables / ip6tables 設定（仮想 IP とする場合）
トランスレータが動作しているマシンのアプリケーションが通信を行うことを防ぐため、フィルタ設定が必要になる。
トランスレータが使用するインターフェースに限定し、全入力パケットを drop、出力パケットは ping/ping6 のみ通過させる。

・ルート設定
必要なルート設定を行う

2.1.3. トランスレータのインストール

パッケージは最終的に SRPM で提供予定であり、以下説明は暫定版となる。
※作業には root 権限が必要。

2.1.3.1. コンパイルとディレクトリ作成

・トランスレータソースは圧縮 tar 形式で提供される。これを適当なディレクトリに展開する。

#tar zxvf ファイル名

front, backend, common ディレクトリが作成されて、それぞれ front 用ソース、backend 用ソース、共通ファイルが展開される。

・フロントトランスレータを作成する場合は front ディレクトリに移動、バックエンドトランスレータの場合、backend に移動し、以下コマンドを入力するとコンパイルとディレク

トリの作成、ファイルコピーが行われる。
#make install

作成されるディレクトリとファイルは以下のとおり

ディレクトリ	ファイル名	説明
/opt/sasat/sbin	sasat_f（フロント) sasat_b（バックエンド）	バイナリファファイル
/var/opt/sasat/etc	sasat.conf（共通） sasat.policy（フロント）	設定ファイル関連
/var/opt/sasat/log	なし	ログ

＊ディレクトリが存在しない場合は作成される。
＊バイナリファイルはインストール時に上書きされる。
＊設定ファイルはインストール時に該当ファイルが存在した場合、処理しない（そのまま)。
　ファイルが存在しない場合、空ファイル（sasat.policy）またはデフォルト設定ファイル（sasat.conf）が作成される。

2.1.4. 動作設定ファイル

動作設定ファイルは各トランスレータの動作モードなどを指示したファイルである。
下記フォーマットにより一行一項目で記述し、指定ディレクトリに配置する。
ファイル名は　sasat.conf

key=data（=の前後にスペースは入れない）

トランスレータは起動時にこのファイルを読み込む。　変更する場合はトランスレータの再起動が必要。

トランスレータが認識する key および data は以下のとおり。

	key	data	意味	F/B *1	デフォルト *2
1	in.ifname	eth0 など	入力で使用するインターフェース名	F,B	eth0
2	eg.ifname	eth1 など	出力で使用するインターフェース名	B	eth1
3	ud_file	ファイルパス名	外部コマンドとの通信用ソケット名	F,B	/dev/shm/.sasat
4	vip_mode	0 or 1	仮想 IP モードかどうか 仮想 IP を使用しない場合はイン	F,B	1(使用する)

			ターフェースに設定された IP を使用する。*3		
5	in.ipaddr4	IP(v4)	入力側仮想 IP（IPv4))	F,B	vip_mode=1 の場合、v4 と v6 のどちらかが必要 *4
6	in.ipaddr6	IP(v6)	入力側仮想 IP（IPv6)	F,B	同上
7	eg.ipaddr4	IP(v4)	出力側仮想 IP（IPv4)	B	同上
8	eg.ipaddr6	IP(v6)	出力側仮想 IP（IPv6)	B	同上
9	svr.ipaddr4	IP(v4)	サーバ IP（IPv4)	B	v4 と v6 のどちらかが必要
10	svr.ipaddr6	IP(v6)	サーバ IP（IPv6)	B	同上

*1 F（フロントトランスレータ）で使用、B（バックエンドトランスレータ）で使用

*2 記述が無い場合のデフォルト値

*3 仮想 IP を使用する場合、トランスレータの IP は OS には設定されなく、トランスレータのみが使用する。 そのため、トランスレータは外部からの ARP 要求/ND による MAC アドレス解決要求に応答する必要がある。 仮想 IP を使用しない場合、トランスレータが使用する IP は OS 設定のものであるため、ARP 要求/ND は OS が処理する。 その場合、アプリケーションがクライアントと通信することを避ける必要があるため、ND を除く全ての IP 受信と ND と ping を除く IP 送信を iptables, ip6tables の設定により破棄する必要がある。

今回の試作では、実験のために両方のモードを実装する（仮想 IP を優先)。

*4 バックエンドトランスレータの場合、ingress, egress, サーバ IP の全てのアドレスが有効の場合のみ有効となる。

2.1.5. 振分け設定ファイル

振分け設定ファイルはフロントトランスレータが使用するもので、クライアントからのパケットを複数のバックエンドトランスレータに振り分ける振分けポリシーを記述する。
振分けは、クライアントの IP アドレス(v4,v6)のみで決定される。
ポリシーは一行一項目で記述し、指定ディレクトリに配置する。
ファイル名は sasat.policy

＊このファイルはトランスレータ動作中に変更して動作に反映させることが可能である。
通信中に変更した場合、振分け先サーバが途中で変更されることになり問題が生じることがあるが、今回の試作では考慮しない。

① 記述フォーマット

デリミタはカンマで以下のように記述する。

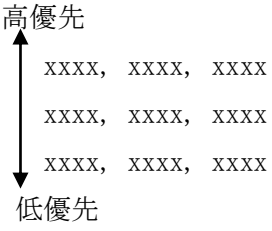
“ターゲット IP”，”マスク値”，”バックエンドトランスレータ IP”

・クライアントの IP アドレスを“マスク値”でマスクしたときの値が、”ターゲット IP”に一致した場合、”バックエンドトランスレータ IP”に送信する。
・“バックエンドトランスレータ IP”が ALL0 の場合、パケットを破棄する。
・IPv4 と v6 ポリシーは混在記述可能。ソフトウェアは処理中に分離する。
・フォーマットにエラーがある場合、その行は無視されてログに記録される。
・記述例は以下のとおり

```
#コメント
10.0.0.0, 255.0.0.1, 10.24.8.1
10.0.0.1, 255.0.0.1, 10.24.8.2
0.0.0.0, 0.0.0.3, 10.24.8.3
0.0.0.1, 0.0.0.3, 10.24.8.4
0.0.0.2, 0.0.0.3, 10.24.8.5
0.0.0.3, 0.0.0.3, 10.24.8.6
::1, ::03, ff02::1
0.0.0.0, 0.0.0.0, 0.0.0.0
# 最後の行では全ての条件にヒットしないとき明示的に破棄している
```

② 優先順位

先頭に近い行が高優先となる。



③ デフォルト動作

全てのポリシーに不一致の場合、パケットは破棄する。

明示的に破棄するポリシーを設定した場合とは統計情報が異なる。

2.1.6. 自動起動設定

トランスレータを自動起動する場合、/etc/rc.sysinit の最後の行に以下文を追加する。

・フロントトランスレータの場合

```
/opt/sasat/sbin/sasat_f -d
```

・バックエンドトランスレータの場合
/opt/sasat/sbin/sasat_b -d

コマンドオプション”-d”はデーモンとして動作することを意味し、“-d”が無い場合は通常コマンドとして動作する（デバッグ用途）。

2.1.7. サーバ設定

サーバにはIPアドレスおよびルート設定を手動で行う。
L3構成の場合、デフォルトゲートウェイはバックエンドトランスレータのIPを設定する。

2.2. 起動

事前設定を行ったあと、マシンの再起動またはコマンド入力によりトランスレータを起動する。
再起動の場合トランスレータ起動は前述のとおり/etc/rc.sysinitで自動的に行われ、待ち受け状態となる。
トランスレータは起動時の動作設定読み込み中にMACアドレス解決を行うため、以下の順で起動するのが望ましい。　ただし、順番が異なっても特に問題はない。

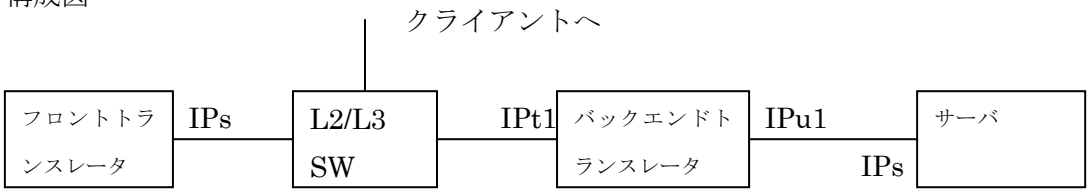
起動順：
サーバ　→　バックエンドトランスレータ　→　フロントトランスレータ

各トランスレータは、起動時にプロセスIDを/var/run/sasat.pidファイルに記録する。

2.3. パケット処理

各トランスレータにおいてパケットは以下のように処理される。

構成図



① フロントトランスレータ

仮想 IP を使用しない場合の前提条件：

・フロントトランスレータで使用するインターフェースに代表 IP（IPs）を OS に設定*

・iptables, ip6tables の設定で受信、送信 IP パケット破棄設定（icmp, icmpv6 は通過）

＊仮想 IP を使用する場合、IPs の代わりに何らかのアドレスを設定する必要がある

トランスレータ処理

対象プロトコル	処理内容
IPv4,v6（送信先：IPs）	宛先 IP を IPt1 に書き換えて送信
IPs への ARP 要求/ND （仮想 IP を使用する場合）	応答（仮想 IP を使用しない場合、OS が処理する）
その他	破棄

OS 処理（仮想 IP を使用しない場合）

対象プロトコル	処理内容
IPs への ARP 要求/ND	OS が処理（iptables で通過設定）
その他	iptables で破棄

② バックエンドトランスレータ

仮想 IP を使用しない場合の前提条件

・フロントトランスレータ側インターフェースに IPt1、サーバ側インターフェースに IPu1 を設定

・iptables, ip6tables の設定で受信、送信 IP パケット破棄設定（icmp, icmpv6 は通過）

＊仮想 IP を使用する場合、IPt1, IPu1 の代わりに何らかのアドレスを設定する必要がある

共通の前提条件

・サーバのデフォルト GW に IPu1 を設定

トランスレータ処理（入力側）

対象プロトコル	処理内容
IPv4,v6（送信先：IPt1）	宛先 IP を IPs に書き換えて、サーバへ送信
IPt1 への ARP 要求/ND （仮想 IP を使用しない場合）	応答（仮想 IP を使用しない場合、OS が処理する）
その他	破棄

トランスレータ処理（出力側）

対象プロトコル	処理内容
IPv4,v6（送信元：IPs）	書き換えなしで通過
IPu1 への ARP 要求/ND （仮想 IP を使用しない場合）	応答（仮想 IP を使用しない場合、OS が処理する）
IPu1 以外の ARP 要求/ND	代理応答（トランスレータが MAC アドレス解決を行って、応答する）
その他	破棄

OS 処理（仮想 IP を使用しない場合）

対象プロトコル	処理内容
IPt1, IPu1 の ARP 要求/ND	OS 処理（iptables で通過設定）
その他	iptables で破棄

2.4. 設定変更

動作設定ファイルまたは振分け設定ファイルを変更する場合の手順について。

動作設定ファイル変更：
動作設定ファイルを変更する場合トランスレータの再起動が必要となる。
動作設定ファイルを修正後、停止シグナル送信およびコマンドによる起動を行う。

停止のシグナルを送信は root 権限で以下のコマンドを打つ。
#kill -9 ‘cat /var/run/sasat.pid’

ログは全て消えるため、必要な場合はコマンド入力前に取得する。

振分け設定ファイル変更：
振分け設定ファイルはフロントトランスレータ動作中の変更が可能である。
振分け設定ファイルを修正後、シグナルまたはプログラムによるコマンド送信でトランスレータに更新を通知する。
フロンとトランスレータは通知を受け取ると、振分け設定ファイルを読み込んで内部情報を更新する。

シグナルを送信する場合は root 権限で以下のコマンドを打つ。
#kill –HUP　‘cat /var/run/sasat.pid’

コマンド送信は unix domain ソケットによってトランスレータと通信することが可能である。　コマンド等は 2.7 項を参照。

2.5.　サーバの追加・削除

サーバの追加および削除手順について

①　サーバ追加の場合
・サーバとバックエンドトランスレータを用意・起動する。
・バックエンドトランスレータの動作設定ファイルを作成・編集して再起動する。
・フロントトランスレータの振分け設定ファイルを編集し、設定反映させる。

②　サーバ削除の場合
・フロントトランスレータの振分け設定ファイルを更新し、設定反映させる。
・サーバを停止する。

2.6.　統計・ログの確認

トランスレータの動作状態、アクセスしているクライアントの情報はトランスレータの統計・ログを取得することで確認できる。
この情報は、トランスレータへのコマンドによって取得（ファイル出力）が可能である。
取得コマンドは何時でも発行可能であるが、コマンドにより出力されるファイル /opt/sasat/log/sasat.log でファイル名固定のため、必要ならばコマンド出力前に古いファイルをセーブして上書きされるのを防ぐ必要がある。
統計・ログには複数のセクションがあるため、コマンドでどれを取得するか指定する。
起動失敗するエラーは内部ログではなく syslog に出力される。

ログファイル内容：
統計ファイルは内容を示す 16 文字(終端無)の識別子により、フロントの場合 6 つ、バックエンドの場合 4 つのセクションに分けられる。

1. 統計セクション
   内部統計情報（振分け設定に付随するものは 4 項）
   フロントとバックエンドでは内容は異なる（詳細は未定）
2. mlog セクション
   メッセージログ。起動時や非常事態のみ記録されるため数は少ない。 最大 128 個（仮）。
   128 個を超える場合、古いものは上書きされる。
   ヘッダ部（１つ）とデータ部（MAX128 個）の２つのサブセクションに分けられ、それぞれの先頭に 16 文字の識別子が付けられる。
3. event log セクション
   イベントログ。 通常動作のイベントを記録する。最大 4096 個（仮）。
   4096 個を超える場合、古いものは上書きされる。
   ヘッダ部（１つ）とデータ部（MAX4096 個）の２つのサブセクションに分けられ、それぞれの先頭に 16 文字の識別子が付けられる。
4. クライアント情報
   ・クライアント毎の上りパケット数
   ・クライアント毎の下りパケット数（バックエンドのみ）
   ※保持するクライアント数には上限あり
5. 振分け統計（フロントトランスレータのみ）
   各振分け設定に関連する以下の数をカウント
   ・各振分け設定のヒット数（パケット数）
   ・各振分け設定のヒット数（クライアント数）
6. サーバ統計（フロントトランスレータのみ）
   ・各サーバ（バックエンド）へのパケット数

フォーマット（内容は全て ASCII 文字列であり、エディタで参照可能）

セクション	識別子(16 文字)	フォーマット
統計	“STATISTICS　　”	“カウント数（10 文字 MAX）”説明(28 文字 MAX)” の繰り返し
mlog	“MLOG　　”	ヘッダ部（” MLOG HEADER　　“16 文字） ・mlog 数（10 進数 4 桁）/ ・開始時間（20 文字 例“2010/01/20 21:10:00”）¥n データ部（” MLOG DATA　　“16 文字） ・シーケンス番号（10 進数 10 桁）/ ・時間（20 文字 例“2010/01/20 21:10:04”）/ ・付属データ（128 文字）¥n

event log	“EVENTLOG ”	ヘッダ部（”EVTLOG HEADER “16 文字） ・evtlog 数（10 進数 4 桁）/ ・開始時間（20 文字 例“2010/01/20 21:10:00”）¥n データ部（”EVTLOG DATA “16 文字） ・シーケンス番号（10 進数 10 桁）/ ・時間（20 文字 例“2010/01/20 21:10:11”）/ ・情報１（16 進数 8 桁）/ ・情報２（16 進数 8 桁）/ ・付属データ（32 文字）¥n
クライアント情報	“CLIENT INFO “	上り（”INGRESS “16 文字） ・IP アドレス/ ・パケット数（16 進数 8 桁）¥n 下り（”EGRESS “16 文字） ・IP アドレス/ ・パケット数（16 進数 8 桁）¥n
振分け統計	“STAT2 ”	・振分け設定番号（10 進 2 桁）/ ・”IP,MASK,SERVER IP”（サイズ不定)/ ・パケット数(16 進 4 桁)/ ・ヒット数（クライアント数）（16 進 4 桁）¥n
サーバ情報	“SERVER INFO “	・バックエンドトランスレータ IP/ ・パケット数(16 進 4 桁)¥n

2.7. トランスレータと外部プログラムの通信

トランスレータは動作設定ファイルで記述された名前の unix domain ソケットをオープン、外部プログラムからのコマンドを待ち受けて実行後結果を通知する。

外部プログラムは、通信するとき unix domain ソケット（AF_LOCAL, SOCK_DGRAM）を作成してコマンドを送信して結果を受け取る。

コマンドのマジックナンバー、コマンドサイズが不正な場合、トランスレータは静かに破棄する。
コマンドコード、付属データが不正な場合、エラーを通知する。

コマンドフォーマット：

byte	値	意味
0	0x74	マジックナンバー(4Byte)
1	0x6e	
2	0x72	
3	0x46	
4	コマンドコード	1 - ログ出力 2 - 振分け設定ファイル更新（フロントのみ）
5	付属データ	
6	0	—
7	0	—

付属データ：

2（振分け設定ファイル更新）の場合、無効

1（ログ出力）の場合、出力内容をビットマップで指示する。

bit 0 -- 統計出力

bit 1 -- mlog 出力

bit 2 -- event log 出力

bit 3 -- クライアント情報出力

bit 4 -- 振分け統計出力

bit 5 -- サーバ情報出力

レスポンスフォーマット： 8Byte

byte	値	意味
0	0x74	マジックナンバー(4Byte)
1	0x6e	
2	0x72	
3	0x46	
4	コマンドコード	受信コマンドと同じ
5	付属データ	受信コマンドと同じ
6	結果	0 -- OK, 0 以外 -- NG
7	理由コード	NG の場合、理由を表示

理由コード：

1: ビジー（処理中）

2: コマンド不正

2.8. 停止

トランスレータを停止するコマンド等はないため、停止するときは kill する。

#kill -9 ‘cat /var/run/sasat.pid’

3. トランスレータ別処理

各トランスレータの処理の詳細について記述する

3.1. フロントトランスレータ処理

3.1.1. 概要

フロントトランスレータは、受信した代表 IP 宛の IP パケットを振分け設定に従って振分ける。　また、設定によっては ARP/ND 処理を行う。
本試作では、ワンアーム構成のみをサポートする。

フロントトランスレータは２つのスレッドで構成される。
①コマンド処理スレッド
　コマンドを受信して処理するスレッド。
②ネットワーク処理（振分け）スレッド
　振分け処理、ARP/ND 処理、振分けの再設定などを行う。

２つのスレッド間での排他処理を行わないで済むように配慮する。

3.1.2. 起動時処理

起動時は以下処理を行う。
・デーモン化処理
・各初期化（tsc クロック読み出し、ログ、サーバ管理テーブル、振分け管理テーブルなど）
・動作設定ファイル読み込み
・振分け設定ファイル読み込み
・スレッド起動
　① ネットワーク処理スレッド
　　・raw ソケット作成
　　・待ち受け
　② コマンド処理スレッド
　　・unix domain ソケット作成
　　・待ち受け

3.1.2.1. デーモン化処理

起動時のオプションで”-d”が指定されている場合、デーモンとして動作する。
daemon 関数で処理。　作業ディレクトリは　/var/opt/sasat　を指定する。

3.1.2.2. 初期化処理

tsc クロック読み出し：

/proc/cpuinfo ファイルの”cpu MHz”値を記録する。これは CPU の持つ TSC(time stamp counter)のクロックであり、TSC 値を実時間に換算する為に使用する。　cpuspeed デーモンが動作すると、cpu 負荷によってクロックが可変になり TSC 値が不正確になるため、停止する必要がある。
ログ : mlog, eventlog 機能の初期化。　ログ領域の確保とログコントロールデータの初期化を行う。

ログコントロールデータは以下の情報を持つ。
　unsigned int seqno;　（シーケンス番号）
　long starttime;　(clock_gettime()による開始時間(秒）)
　uint64_t tsc;　　(starttime とほぼ同時の tsc 値)
　double cpu_clock; (tsc clock)
ログデータは以下の情報を持つ。
(mlog)
　unsigned int seqno; (シーケンス番号)
　uint64_t tsc;　　（tsc 値）
　char m_data[128];　(データ)
(eventlog)
　uint32_t trace_id;　(ログの ID)
　uint64_t time;　　(tsc 値)
　unsigned long info1;　(情報１)
　unsigned long info2;　(情報２)
　char e_data[32];　(データ)

サーバ管理テーブル：
　振り分け先情報を持つサーバテーブルのリンクヘッダの初期化。

振分け管理テーブル：
　振分け情報を持つ振分けテーブルのリンクヘッダの初期化、フリーの振分けキャッシュテーブルのプール作成。

3.1.2.3. 動作設定ファイル読み込み処理

動作設定ファイル読み込み処理では、動作設定ファイル（2.1.4 項）を読み込んで、キーとデータの組み合わせのデータベースを作成、情報を双方向リストとして保持する。

プログラム上で設定の参照が必要な場合、キーを元にデータベースを参照、設定データを読み込み動作を決定する。

3.1.2.4. 振り分け設定ファイル読み込み処理

フロントトランスレータは初期化時および処理が指示されたとき、振分け設定ファイルを読み込み、内部に以下情報として持つ。

① 振分けテーブル

双方向リストで優先度順にリンクされ、振分け情報（IP アドレス、マスク値、サーバテーブルへのポインタ、統計情報など）を持つテーブル。　パケット受信時に振分け先を決定するとき、振分けテーブルの先頭から順番に検索する。

② サーバテーブル

単一方リストでリンクされた、バックエンドトランスレータ情報（バックエンドトランスレータの IP、GW IP、GW の MAC アドレス、統計情報など）を持つテーブル。
振分け設定ファイルで指定される振分け先情報分（IPv4, v6 別）作成される。
サーバテーブルを作成するとき、GW IP とその MAC アドレスを OS のルーティングテーブルを参照して決定する（3.1.5 項）。
ここで解決しない場合は、解決していないことを示す情報をテーブルに書き込み、実際にパケットを振り分けるときに解決処理を行う。
それでも解決しない場合はパケットは破棄される。
保持する MAC アドレスは、一度決定すると更新しない。

3.1.2.5. スレッド起動

前述のとおり、フロントトランスレータはコマンド処理スレッドとネットワーク処理スレッドの２つのスレッドで動作する。　実際はコマンド処理スレッドが親となり子スレッドであるネットワーク処理スレッドを起動する。　スレッドの同期は不要。
それぞれのスレッドの動作は以下。

コマンド処理スレッド：

① unix domain ソケットを作成して、待ち受ける(以下オプションなど)。
・ソケットは AF_LOCAL, SOCK_DGRAM
・root 以外のプログラムと通信できるよう umask(000)で作成する
・non blocking 設定する
② pid ファイルを作成（/var/run/sasat.pid）
③ 設定更新用シグナルハンドラ設定
④ コマンド待ち受け

ネットワーク処理スレッド：

① シグナルブロック
② raw ソケットを作成して待ち受ける（以下ソケットオプションなど）
・ソケットは AF_PACKET, SOCK_RAW, ETH_P_ALL 指定
・システム最大バッファを設定（送受信 512KB）
・ソケットの最大バッファを設定（送受信 512KB）
・インターフェースと bind し UP(IFF_UP)する
・non blocking 設定
③ 受信タイムアウトを設定し、パケット待ち受け

3.1.3. コマンド処理

コマンドはコマンド処理スレッドで受信、応答する。
コマンドとそのレスポンスについては、2.7 項を参照。

振分け設定更新コマンド（フロントのみ）：

コマンド処理スレッドが振分け設定更新コマンド（および HUP シグナル）を受信したとき、以下の処理を行う。

① コマンド処理スレッドは、ネットワーク処理スレッドへ停止指示（pthread_cancel）を発行し、停止を待つ。
② ネットワーク処理スレッドは、pthread_cancel によりクリーンアップハンドラが呼び出されるので、その中でソケットのクローズを行い、停止完了状態をフラグによ

り報告する。

③ コマンド処理スレッドは、サーバテーブル、振分けテーブル、振分けキャッシュテーブルの初期化を行ったあと、振分け設定ファイルの再読み込みを実行して振分けテーブル、サーバテーブルの再構築を行い、ネットワーク処理スレッドを再起動、コマンド処理結果を通知する。

ログ出力コマンド：

コマンド処理スレッドは、ログ出力コマンドを受信すると必要な内部ログを読み出してフォーマット変換してファイルに書き出す処理を行う。

このとき、ネットワーク処理スレッドとの排他的な問題が生じないよう注意する。

また、CPU 負荷が上がりすぎないようにスリープを挟むなど行う。

3.1.4. 振り分け処理

① 代表 IP 宛パケットを受信したとき、以下の動作をおこなう。

・送信元 IP アドレスを取り出し、振分けキャッシュテーブルをハッシュ検索する。
ヒットしなかった場合、振分けテーブルを検索して振分け先を決定し、振分けキャッシュテーブル作成またはパケット破棄する。

・振分けキャッシュテーブルから送信先 IP、送信先 MAC を取り出してパケットを書き換える。　このとき、MAC アドレスが未解決の場合、ここで解決する。
※本来、MAC 解決処理は、振分け処理への影響を考慮し、別の処理キューなどを作成して他スレッドで行うべきかもしれないが、今回は試作のためそこまで行わない。

② ARP 要求/ND（代表 IP 向け）を受信したとき、かつ仮想 IP 動作の場合

ARP/ND を自分の MAC アドレスで応答する。

振分けテーブルについて：

ユーザが用意する振分け設定ファイルの情報を保持するテーブルで、振分け設定ファイルで設定された分のテーブルを持つ。

高優先の情報から逐次検索して情報がヒットした場合、その情報でパケット書き換えを実施する。

一度ヒットすると、その振分け先、書き換え内容、IP チェックサム差分を振分けキャッシュテーブルとして保持する。

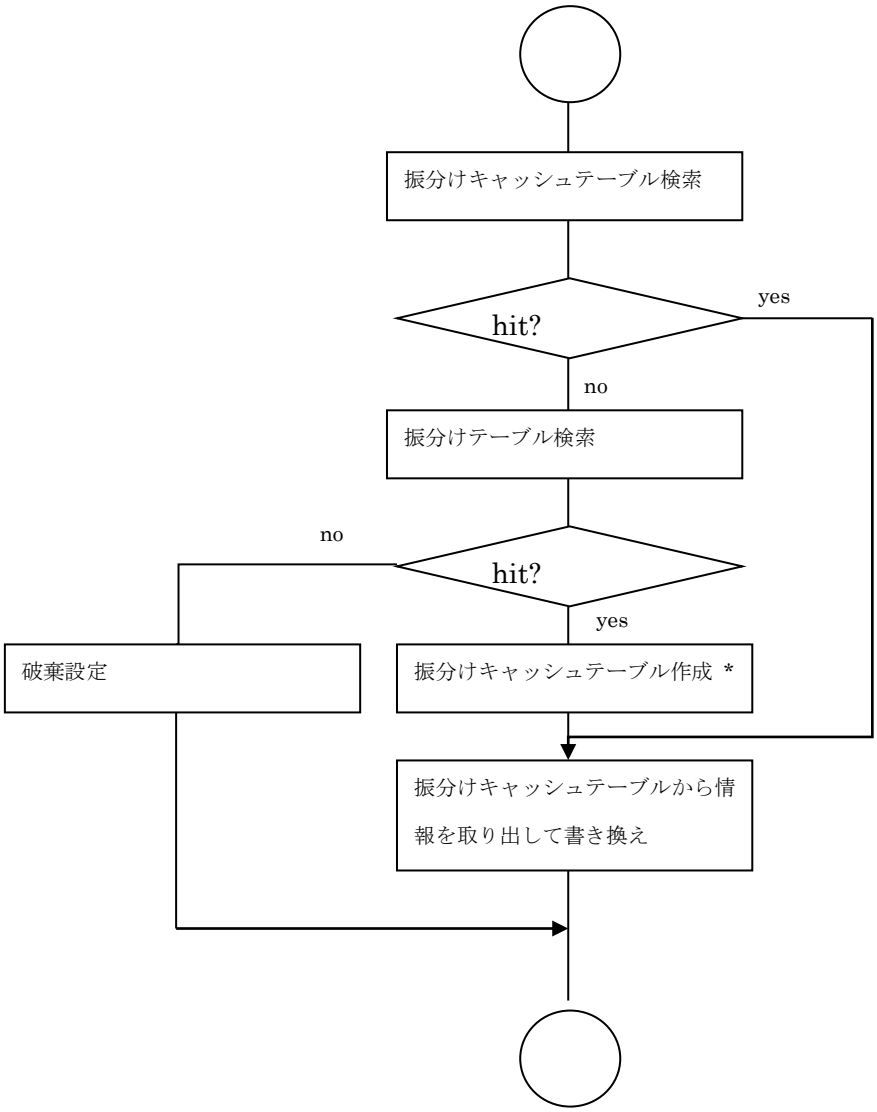
振分けキャッシュテーブルについて：

前述のように振分けテーブルの検索結果をキャッシュするテーブル。送信元 IP アドレスのハッシュ検索により高速検索できる。これにヒットしなかった場合、振分けテーブルの検索が行われる。

振分けキャッシュテーブルは初期化時に大量に確保してプールしておき、必要になった

場合は取り出して使用する。　使いきって空きが無くなった場合、一旦全キャッシュを削除する。

振分け検索のフロー



＊振分けテーブルで“破棄”設定されている場合、“破棄”設定された振分けキャッシュテーブルが作成される。　振分けテーブルにヒットしなかった場合、振分けキャッシュテーブルは作成されずに破棄される。

3.1.5. MAC アドレス解決処理

宛先のMACアドレスはルーティングソケット（AF_NETLINK, SOCK_RAW, NETLINK_ROUTE）を使用して、OSからゲートウェイIPアドレスとMACアドレスを取得する。
MACアドレスが取得できない場合、ping/ping6を送信してOSにMACアドレスを解決させる。

3.2. バックエンドトランスレータ処理

3.2.1. 概要

バックエンドトランスレータは、２つの NIC を使用してフロントトランスレータとサーバとのブリッジを行う。

バックエンドトランスレータはフロントトランスレータのような振分けテーブル、振分けキャッシュテーブルは持たないが、ログ目的でクライアント情報テーブルをクライアント毎（上り、下り別）に作成する。　これは、振分けキャッシュテーブルのように初期化時にプールを作成、使用時はそこから取り出してハッシュ登録する。

バックエンドトランスレータは３つのスレッドで構成される。

・コマンド処理スレッド
　コマンドを受信して処理するスレッド。
・入力側ネットワーク処理スレッド
　入力側では受信した特定の宛先 IP パケットを実サーバへ転送する。　また、設定によっては ARP/ND 処理を行う。
・出力側ネットワーク処理スレッド
　サーバからのパケットを素通しする。
　サーバは MAC 解決をバックエンドトランスレータ経由で行うため、バックエンドトランスレータは IP アドレスを含めてフレーム書き換えの必要がない。
　サーバからの ARP 要求/ND を代理で処理してサーバに応答する。

　３つのスレッド間での排他処理を行わないで済むように配慮する。

3.2.2. 起動時処理

起動時は以下処理を行う。
・デーモン化処理
・各初期化（tsc クロック読み出し、ログ、クライアント情報テーブルなど）
・動作設定ファイル読み込み
・スレッド起動
　① 上りネットワーク処理スレッド
　　・raw ソケット作成
　　・待ち受け
　② 下りネットワーク処理スレッド
　　・raw ソケット作成
　　・待ち受け

③ コマンド処理スレッド

・unix domain ソケット作成

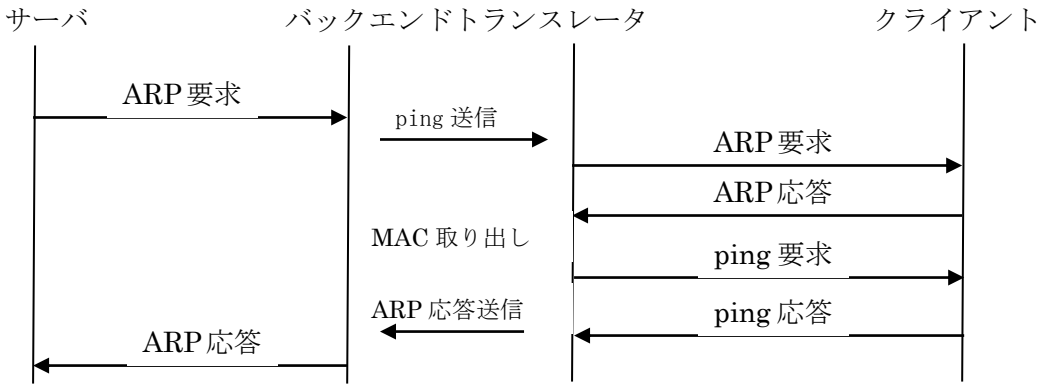
・待ち受け

3.2.2.1. スレッド起動

前述のとおり、バックエンドトランスレータはコマンド処理スレッドと上りネットワーク処理スレッド、下りネットワーク処理スレッドの３つのスレッドで動作する。　実際はコマンド処理スレッドが親となり子スレッドであるネットワーク処理スレッドを起動する。
２つのネットワーク処理スレッドは、待ち受け前に逆側のネットワーク処理スレッドの初期化が完了している必要があり（逆側のソケットを使用するため)、初期化完了の同期を取る。
それぞれのスレッドの初期化動作は、フロントトランスレータと同じであるため割愛する。

3.2.3. ARP/ND 代理応答

L2 構成において下りインターフェース（サーバ側）では、サーバは宛先の MAC アドレスを解決するために ARP 要求/ND を直接宛先に送信する。
このフレームをそのまま中継することはできないため、バックエンドトランスレータは代理応答処理を行う。



① サーバからの ARP 要求/ND を受信したとき、かつ解決する MAC がバックエンドトランスレータ以外だった場合、バックエンドトランスレータの実 IP からクライアント宛に ping/ping6 を送信して MAC アドレスを解決する。
*ping を使用するのは処理の簡略化のため。

② ping 送信後、ルーティングソケットを使用してクライアントの MAC アドレスを OS から取り出し、サーバ宛の応答パケットを作成し送信する。
MAC アドレスが取得できない場合、応答パケットは送信しない。

3.2.4. クライアント情報テーブル

クライアント情報テーブルは通信したクライアント情報をクライアント単位（上り、下り別）で保持するテーブルである。
上り、下り別にハッシュ登録され、ヒットカウンタを持つ。
また、IPv4 の上りでは IP チェックサム差分情報も持つ。

以上