

組み込みLinuxにおける ネットワーク帯域制御

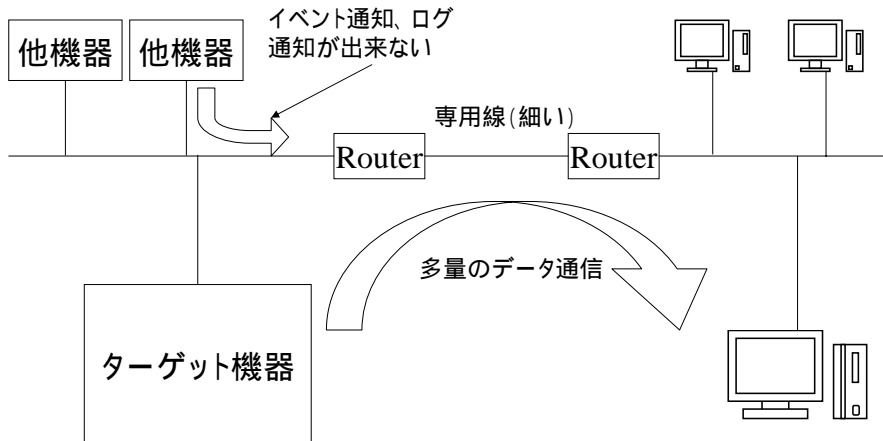
三菱電機株式会社
先端技術総合研究所
八木孝介



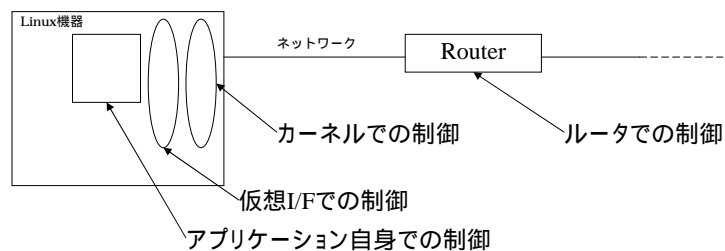
概要

- Linux-2.4.20のQoS関連機能を使用
- tcコマンド、cbq.initスクリプトを中心に使用して設定を行った
- Sh4プラットフォームで動作確認を行った
- 期待する制御が出来ていることを確認した
- シンプルなネットワーク帯域制限機能として十分使用に耐える？

想定場面

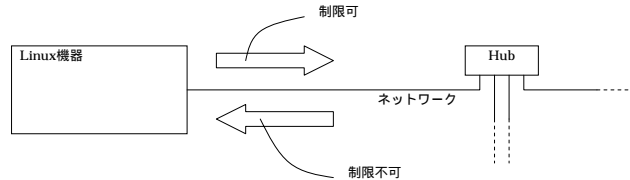


トラフィック制御を行う位置



- ルータによるトラフィック制御(もっとも一般的)
- カーネルによるトラフィック制御(カーネルの機能として組み込まれている。高機能だが設定は複雑。)
- アプリケーション自身による制御 (Apacheなど一部のソフトで対応可能。自作アプリの場合は仕込みが必要。横断的な制御は困難。)
- Netnice等の仮想インターフェースを使用(まだ一般的でない)
機器単独で実現できて、負荷の軽いカーネルでの制御を選択した。

Linuxカーネルで出来ることと出来ないこと



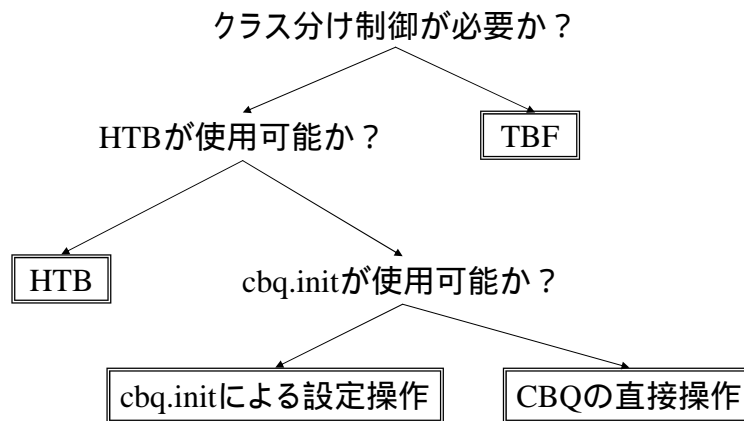
- カーネルの機能として帯域制限、優先順位制御が可能
- 出ていくパケットは制御可能
- 入ってくるパケットは制御不可能(入ってくるパケットに関しては受け取り後に捨てることは可能)
- インタフェース、IP、ポート番号、netfilterのルール等を条件として制御することが出来る
- アプリケーション、ユーザ等を条件として制御することは出来ない

基本的にはカーネルのネットワークQoS機能はLinuxでルータを作ることを目的としている。そのため、端末機器としてみた場合、アプリケーションごとの設定や、ユーザごとの設定という概念が存在しないため不便なことも多い。

Linuxで利用可能なキューイング規則 (qdisc: Queueing Discipline)

	qdisc	概要
デフォルト	pfifo_fast	シンプルで高速なデフォルトFIFO。パケットの優先度フラグによる処理を行う。
クラスレス	TBF	帯域制限用。設定した通信量以下に抑える。瞬間的には設定値を超えることあり。
	sfq	特定の通信が帯域を占有しないように公平に分配する。
クラスフル	CBQ	帯域制限、優先度制御用。様々な条件でクラスを分け制御を行う。設定が難しい。
	HTB	帯域制限用。様々な条件でクラス分けを設定し、帯域制限を行う。
	prio	優先度制御用。様々な条件でクラス分けを設定し、優先度順の送信を行う。
特殊用途	bfifo	統計用
	pfifo	統計用
	CSZ	
	DSMARK	
	RED	
	GRED	

qdiscの選択 (帯域制限の場合)

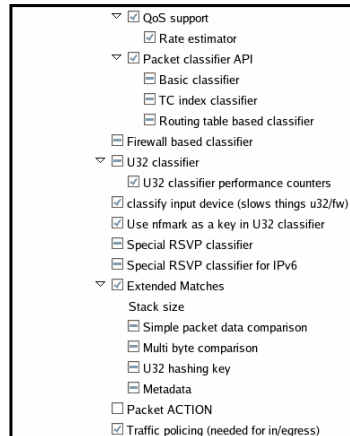
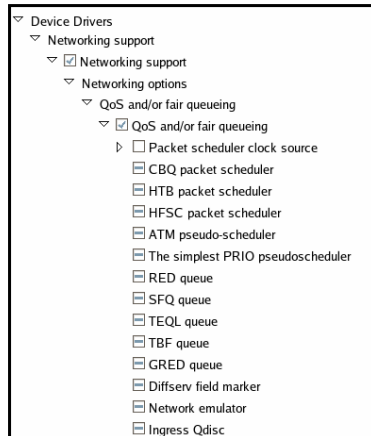


カーネルの設定

“traffic control Mini-Howto”によると、カーネル2.2系では次の機能を有効にする必要がある。

- Prompt for development and/or incomplete code/drivers
- QoS and/or fair queuing
- CBQ packet scheduler
- TBF queue
- QoS support
- Rate estimator
- Packet classifier API
- Routing table based classifier
- Firewall based classifier
- U32 classifier
- Ingress traffic policing

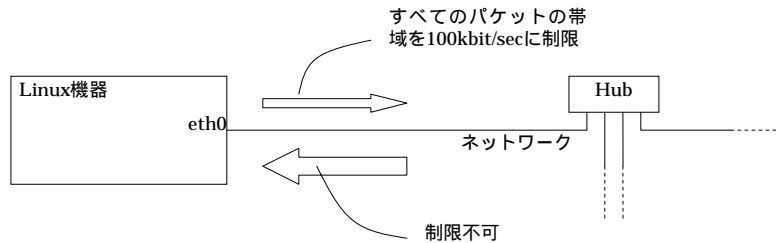
カーネルの設定



ツール類

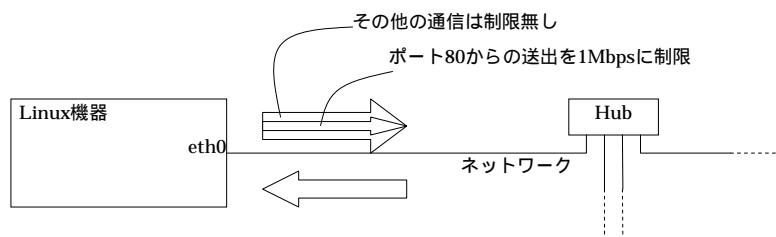
- iproute2
ifcfg、ip、rtmon、netstatなど、ネットワーク機能設定のためのツールパッケージ
- tc
QoS機能を設定するためのコマンド
- iptables
パケットフィルタリング機能を設定するためのコマンド。tcと組み合わせて高度なQoSを設定することも可能。
- cbq.init
qdiscの一つであるcbqの設定を簡易化するためのスクリプト。

設定例1



- すべての通信を等しく制御するにはクラスレスのqdiscが適している。
- `tc qdisc add dev eth0 root tbf limit 15Kb buffer 10Kb/8 rate 100kbit`

設定例2



- 条件による制限を行う場合はクラスフルなqdiscが適している
- CBQはcbq.initスクリプトを用いると設定が簡略化できる
- cbq.initの設定ファイル(/etc/sysconfig/cbq/cbq-10.http)
DEVICE=eth0,100Mbit,10Mbit
RATE=1Mbit
WEIGHT=100kbit
RULE=:80,

設定例 2

- 生成された制御コマンド

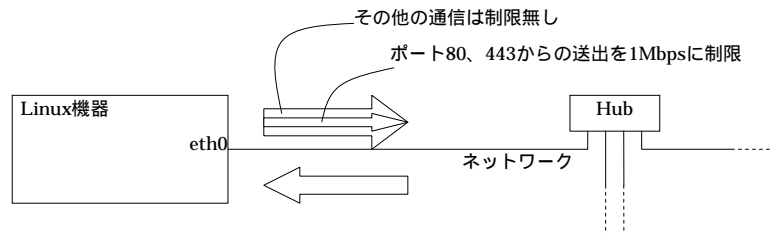

```
/sbin/tc qdisc del dev eth0 root
/sbin/tc qdisc add dev eth0 root handle 1 cbq bandwidth 100Mbit avpkt
1000 cell 8
/sbin/tc class change dev eth0 root cbq weight 10Mbit allot 1514

/sbin/tc class add dev eth0 parent 1: classid 1:10 cbq bandwidth 100Mbit
rate 10Mbit weight 1Mbit prio 5 allot 1514 cell 8 maxburst 20 avpkt
1000 bounded
/sbin/tc qdisc add dev eth0 parent 1:10 handle 10 tbf rate 10Mbit buffer
10Kb/8 limit 15Kb mtu 1500
/sbin/tc filter add dev eth0 parent 1:0 protocol ip prio 100 u32 match ip
sport 80 0xffff classid 1:10
```

cbq.initについて

- tcコマンドでcbqを使用するためのヘルパースクリプト
- GPLライセンス
- マニュアルはコードの中に書かれている
- 書式: `cbq.init command`
 SysV init互換のコマンド形式
 一部のコマンドにはroot権限が必要
- コマンド
 - `start` 設定ファイルの内容に従って帯域制限を開始します。
 - `stop` 帯域制限を停止します。
 - `restart` 帯域制限の再起動を行います。
 - `list` 設定内容を表示します。
 - `stats` 統計情報を表示します。
 - `compile` 設定ファイルの内容に従ってコマンド列を出力します。
- 設定ファイル
 - 配置ディレクトリ: `/etc/sysconfig/cbq/`
 - ファイル名の書式: `cbq-xxx.name`
 xxxはクラスを表す数字。0002 ~ FFFFの範囲。
 nameは人間が認識出来るように付ける好きな文字列
 ファイル名の例: `cbq-10.http`
- 兄弟版の`htb.init`もある

設定例3



- `cbq.init`の設定ファイル(/etc/sysconfig/cbq/cbq-10.http)
DEVICE=eth0,100Mbit,10Mbit
RATE=1Mbit
WEIGHT=100Kbit
RULE=:80,
RULE=:443,

設定例3

- 生成された制御コマンド
`/sbin/tc qdisc del dev eth0 root`
`/sbin/tc qdisc add dev eth0 root handle 1 cbq bandwidth 100Mbit avpkt 1000 cell 8`
`/sbin/tc class change dev eth0 root cbq weight 10Mbit allot 1514`

`/sbin/tc class add dev eth0 parent 1: classid 1:10 cbq bandwidth 100Mbit rate 10Mbit weight 1Mbit prio 5 allot 1514 cell 8 maxburst 20 avpkt 1000 bounded`
`/sbin/tc qdisc add dev eth0 parent 1:10 handle 10 tbf rate 10Mbit buffer 10Kb/8 limit 15Kb mtu 1500`
`/sbin/tc filter add dev eth0 parent 1:0 protocol ip prio 100 u32 match ip sport 80 0xffff classid 1:10`
`/sbin/tc filter add dev eth0 parent 1:0 protocol ip prio 100 u32 match ip sport 443 0xffff classid 1:10`

cbq.initによるより複雑な設定

- 設定ファイルを複数作成することにより、異なった条件で制限をかけることが出来る (httpは10Mbps、ftpは5Mbps等)
- 親クラスを設定することで、いくつかの設定を束にした設定も可能 (httpは8Mbps、ftpは5Mbpsまでだが、両方合わせて10Mbps以下に抑える等)
- クラス間でバンド幅の貸し借りもできる

今回のターゲットと結果

- SH4(7751R)
- Linux 2.4.20 (MontaVistaLinux)
- Ethernet 10/100Mbps
- 帯域制限の設定によりほぼ想定通りの性能が得られた
- 帯域制限によるCPU負荷の増加は見られなかった (通信量減少による負荷軽減の方が効果が大きかった)

まとめ

- LinuxカーネルのネットワークQoS機能を組み込み機器に利用した
- 用途を限れば組込用のネットワーク制御として十分使用可能
- 設定が複雑なので補助ツールの利用が有効である

参考文献

- 『ネットワークトラフィックを制御せよ』、UNIX USER誌、2005年3月号
- “traffic control Mini-Howto”,
< <http://www.linux.or.jp/JF/JFdocs/traffic-control.html> >
- “cbq.init README”, cbq.init付属文書(ソースコード内)
< <http://sourceforge.net/projects/cbqinit> >
- “Linux Advanced Routing & Traffic Control HOWTO”,
< <http://lartc.org/> >
- htb.init
< <http://sourceforge.net/projects/htbinit> >