Содержание

5 Bittorent control option

BTRT - BitTorrent Proxy Tracker

Description

BTRT is a BitTorrent tracker allowing to optimize the network traffic distribution in favor of reducing traffic to external networks by increasing the amount of intranet traffic. Reducing the network traffic volume to external networks is achieved by prioritization of peer IP addresses in response to *announce* requests: according to the configuration IP addresses of the operator's network peers are replied in the first place.

BTRT replies to standard BitTorrent client HTTP requests: *announce* and *scrape*, also it designed to handle non-standard *forwarded_announce* requests, more details on it see here.

Hardware requirements

The main requirement is imposed to the RAM size. BTRT holds the data about all torrent files and peers and more. To calculate the required amount of RAM you can use as basis of 512 bytes per peer.

Software requirements

BTRT runs on CentOS 6. To install the BTRT the following packages should be installed on the system:

- `mysql-server`, version 5.1.73 or later
- `mysql-connector-c++` version 1.1.5 or later

Use cases

Local retracking

The simplest way to use BTRT is to use it as a local retracker. Several trackers that the BitTorrent client should access by sending *announce* requests can be specified in torrent-files. Special name `retracker.local` is used for local retracking. Local retracking is provided by forwarding requests to the `retracker.local` on BTRT.

"Adjacent" retrackers data using

Local retracking allows to redistribute traffic only within the operator's network and provides a

reduction in volumes of traffic to external networks only through operator's customers. The operator often has a contractual relationship with partners which implies a lower traffic cost in the networks of partners. If the tracker is launched in the partner's network, the use of peers registered on this server allows to reduce the traffic volumes in external networks. When processing *announce* requests BTRT enables to use not only local peers data but also to request the data from retrackers operating in partner networks. In response to *announce* request in this case BitTorrent client is sent the IP addresses of peers registered both in operator's network and in partner networks resulting in redistribution of traffic generated by BitTorrent clients.

Handling of requests to external trackers

BTRT handles the *forwarded_announce* requests generated by DPI platform. *Forwarded_announce* requests stand for *announce* requests sent to external BitTorrent trackers, which in turn are intercepted by DPI platform and forwarded to the BTRT.

When the *forwarded_announce* request is processed, BTRT, if necessary, sends the request to an external (third-party) tracker for the list of peers, but a list of peers is formed according to the peers priorities specified in the configuration (local peers, partner peers, etc.) and then the list is sent to the BitTorren client. So having enough peers in operator and partner networks, it's possible to significantly limit the bandwidth when communicating with the external peers, so to redistribute the traffic in such a way that the intranet traffic will be increased.

Installation

The packet and its dependencies should be installed in order to properly install the application initially. It could be done by applying the command:

```
sudo yum install btrt
```

BTRT uses MySQL database. To perform further installation steps, you must run the service **mysqld** :

```
sudo service mysqld start
```

User account and MySQL schema should be created before the first launch of BTRT (existing user account can also be used). It is highly recommended to create specific MySQL account for the BTRT. To do so, you need to connect to the MySQL server with superuser privileges:

```
mysql -hlocalhost -uroot -p
```

and execute the account and schema creation commands providing user password:

```
CREATE USER 'btrt'@'localhost' IDENTIFIED BY 'password_for_btrt_user';
CREATE DATABASE btrt;
GRANT ALL PRIVILEGES ON btrt.* TO 'btrt'@'localhost';
```

After the account is created, you have to finish the privileged MySQL session by pressing the key combination **Ctrl-D** . and create objects in the BTRT data schema, providing the name and password

of the previously created MySQL user:

```
mysql -hlocalhost -ubtrt -p < /etc/btrt/btrt.db_creation.sql
```

Before the first start, you have to create an /etc/btrt/btrt.conf configuration file based on the /etc/btrt/btrt.conf.default configuration template. More details on configuration is here.

Updating

To update a previously installed application, the tracker service should be stopped:

```
sudo service btrt stop
```

then the updated btrt packet should be installed:

```
sudo yum update btrt
```

When upgrading from version 2.1 and below, you need to update the database structure:

```
mysql -hlocalhost -ubtrt -p < /etc/btrt/btrt.db_upgrade_2.1.0.sql
```

The consistency of the configuration has to be checked due to the possible changes therein caused by upgrading, and then to launch the tracker service:

```
sudo service btrt start
```

Configuration

The main configuration file is /etc/btrt/btrt.conf. The configuration in this file is specified in the YAML format.

Формат YAML

The base YAML elements are:

- "Key-Value" pair, for example:
  `level: "info"`
- value sequence, for example:
  `- "127.0.0.0/8"`
  `- "192.168.0.0/16"`

It is possible to create complex data structures by using the basic elements.

One of the key principles describing complex structures behind the YAML format is the rule of formatting nested elements using the indents from the beginning of a line. The same hierarchy level

elements should have the same number of leading spaces, for example, the following text:

```
#
# Logging parameters
logging:
    path:           "/var/log/btrt"     # Path to log files, default:
/var/log/btrt
    level:          "info"              # Possible levels are:
                                        #   critical, error, warning, info,
diagnostic, debug
                                        # default logging level is "info"
    switch_time:    "01:00:00"          # Log switching time in format
"HH:MI:SS", default: "01:00:00"
```

can be interpreted as data structure `logging` containing three fields: `path`, `level` and `switch_time`.

As pointed out above, given that the guiding principle for editing documents in YAML format **is to use a certain number of whitespace characters to form indents instead of using tabs**.

The main configuration file /etc/btrt/btrt.conf

The main configuration file contains the description of most parameters used by the BTRT. Initially, when the package is installed, the configuration template /etc/btrt/btrt.conf.default file is created. It describes all of the settings and options with corresponding comments and can be used to create the working configuration.

To create a working configuration, the following parameters have to be modified at least:

Description of connection to the MySQL database:

```
database:
    url:        "tcp://127.0.0.1:3306"  # URL for connection to MySQL,
default: "tcp://127.0.0.1:3306"
    user:       "root"                  # User name, default: "root"
    password:   "root"                  # Password
    schema:     "btrt"                  # Schema name, default: "btrt"
```

Description of the IP addresses pools:

```
    peer_groups:
        # It is useful to have definition for peers in local network.
        # It can be used when peer from the local network is registered on
non-local tracker (neighbor, forwarded tracker)
        -   name:       "local peers"
            priority:   0
            cidr_file:  "/etc/btrt/local_cidrs.conf"
            cidr_list:
                -
```

```
        # neighboring peers is used to identifying range of IP-addresses of
peers connected to neighboring trackers
        -   name:       "neighboring peers"
            priority:   1
            cidr_file:  "/etc/btrt/neighboring_cidrs.conf"
            cidr_list:
                -
```

Description of the "neighbors" trackers:

```
    neighbors:
        -   name:                   "the nearest retracker"
            refresh_interval:       1800
            peers_expiry_interval:  7200
            announce_request:
"http://retracker.mgts.by/announce?info_hash={info_hash}&peer_id=01234567890
12345678A&port=6882&downloaded=0&uploaded=0&left=4194304&event=started&compa
ct=1&numwant=1000"
```

Control

The main executable tracker file is `/usr/sbin/btrt`.
To display the prompting message, the following command should be launched:

```
btrt help
```

The tracker operates as a service. So to start the service, execute the following command:

```
sudo service btrt start
```

to stop the service:

```
sudo service btrt stop
```

The log files are created accrording to the settings specified in configuration file during the tracker operates. Log files are created in the directory `/var/log/btrt` by default.

Troubleshooting

- How often hashlist.bin is updated?

crontab -l - hashlist update every 10 minutes

- Does DPI generate forwarded requests or does it only handle hashlist.bin? In other words, how can I make BTRT take peers from an adjacent tracker?

See the above solution "Adjacent" retrackers data using. In the DPI statistics log there is a section

“Detailed statistics on BitTorrent”. The first digit stands for how many handshakes are processed. The second one is how many are blocked in favor of local distributions