Deploy your own P2P network

Dobrica Pavlinušić
http://blog.rot13.org
FOI Security Symposium, 2011-09-23 http://fsec.foi.hr

fsec

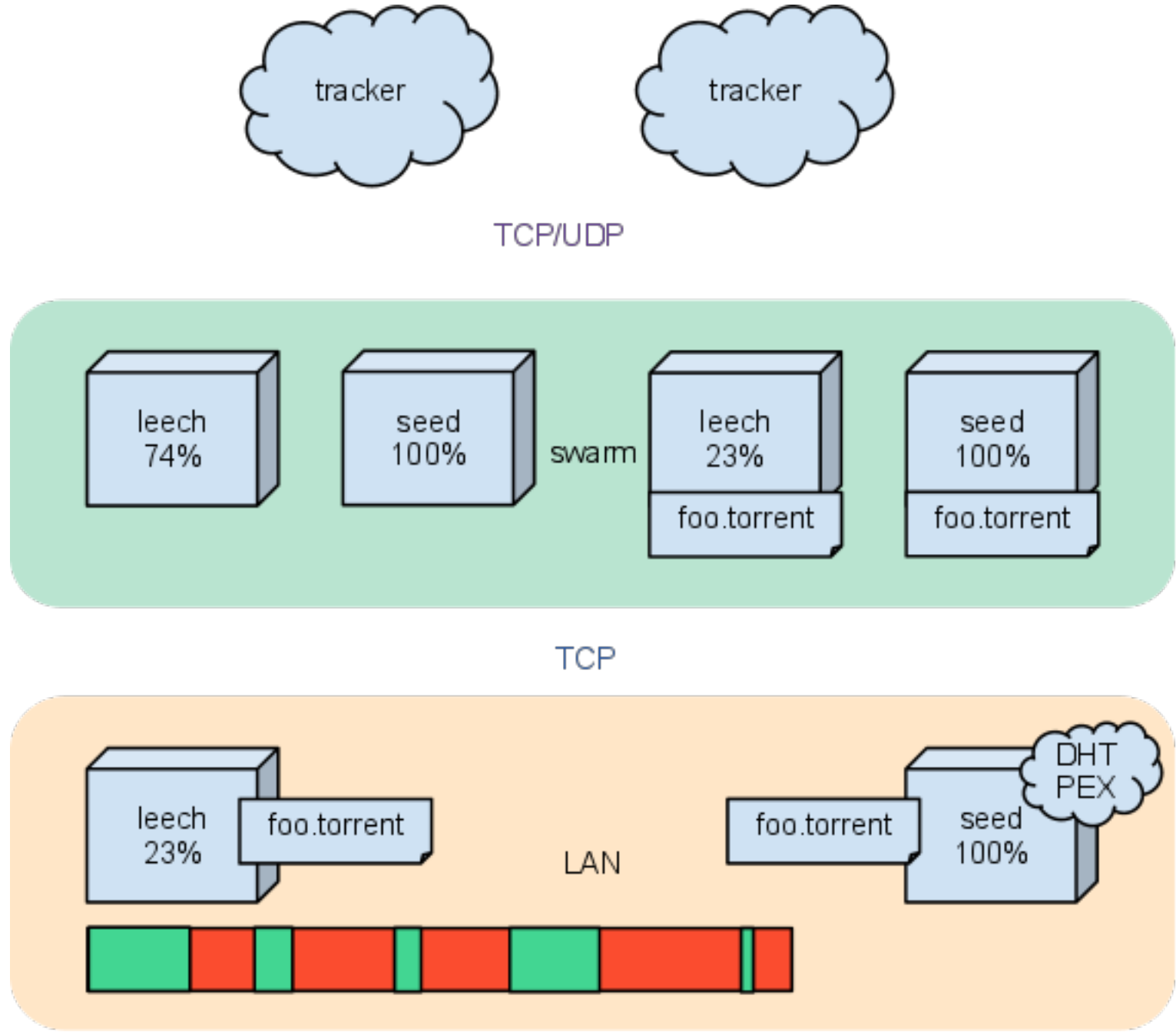
Static file distribution

If it's good for w@rez and p00rn, it must have legitimate uses:

- static file distribution (Linux distributions, software updates, disaster recovery)
- sharing chunks between clients
  - get chunks already available on local LAN (behind NAT!)
  - distributed upload
  - aggregated bandwidth usage

Use existing protocol: **Bit Torrent** (~50 % of Internet traffic, so it must work) but build private P2P cloud controlling all parts of it.
Don't create private swarm if you don't have to - be part of public torrent swarm if possible to get all benefit of peers!

Think about future: IPv6 support, scalability, etc.


tracker
tracker
TCP/UDP
leech
74%
seed
100%
swarm
leech
23%
foo.torrent
seed
100%
foo.torrent
TCP
leech
23%
foo.torrent
LAN
foo.torrent
seed
100%
DHT
PEX

BitTorrent overview BEP 0003

- static .torrent file (SHA1 of info value)
  - announce (tracker URL)
  - comment, creation_date
  - info
    - name (MD5)
    - piece_length (2^18 = 256 K)
    - pieces (20*SHA1 of chunks)
    - length
- tracker
  - HTTP get protocol: info_hash, peer_id, ip, port, uploaded, downloaded, left, event
- BitTorrent client

http://bittorrent.org/beps/bep_0003.html

BitTorrent protocol

http://wiki.theory.org/BitTorrentSpecification **real detailed, up-to-date protocol documentation, not fake introduction!**

.torrent

- info
  - announce-list
  - private (disable PEX DHT!)
  - created_by

Tracker

- HTTP/HTTPS protocol
- scraping

Peer wire protocol (TCP)

Bencoding, Algorithms, Extensions

Torrent tracker

Many, many choices, mostly php scripts...
Wanted simpliest possible solution (without RDBMS if at all possible) with support for multiple instances, ergo:
http://erdgeist.org/arts/software/opentracker/

Documentation needs a bit of love: some options available are not documented, and some documented options don't work in recent version

Compilation from CVS repository

Public tracker at http://openbittorrent.com/

BitTorrent client

Leeching - have torrent, download
Seeding - have torrent and whole file, upload

Transmission

- http://www.transmissionbt.com/
- nice JSON-based API
- I failed to make file seeding work :-(

rTorrent

- http://libtorrent.rakshasa.no/
- XML/RPC API (over socket or port), RTPG protocol
- needs .rtorrent.rc to make API work
    - scgi_local=/tmp/rtorrent.socket

I c@n haz t0rrent neTw0rk?

```
$ opentracker -i $ip_tracker1 -p 6969 -s 9696
$ opentracker -i $ip_tracker2 -p 6969 -s 9696

$ rtorrent -O directory=$srv/md5 \
  -O session=$var/session \
  -O schedule=watch_directory,5,5,load_start=$var/watch/*.torrent \
  -O schedule=tied_directory,5,5,start_tied= \
  -O schedule=untied_directory,5,5,close_untied= \
  -O scgi_local=$var/socket \
  -O bind=$ip_seed -O port_range=6890-6999 \
  -O dht=auto -O dht_port=6881 \
  -O peer_exchange=yes \
  -O download_rate=15 -O upload_rate=15

$ mktorrent --announce http://$ip_tracker1:6969/announce \
  --announce http://$ip_tracker2:6969/announce \
  --no-date --verbose --output $file.torrent $file

$ ln -s $file.torrent $var/watch/ # seed!
```

Conclusion

- It's possible to build Torrent network at home!
- Protocol is not documented as well as we would like
  - it would be nice to have torrent files which don't use filename as part of SHA1 (thus, md5 filenames)
  - it would be handy to have chunks shared, even between different torrent files (since we allready have SHA1 sums)
  - PEX and DHT are black art (and important for NAT scenarios)