parc

A Xerox Company

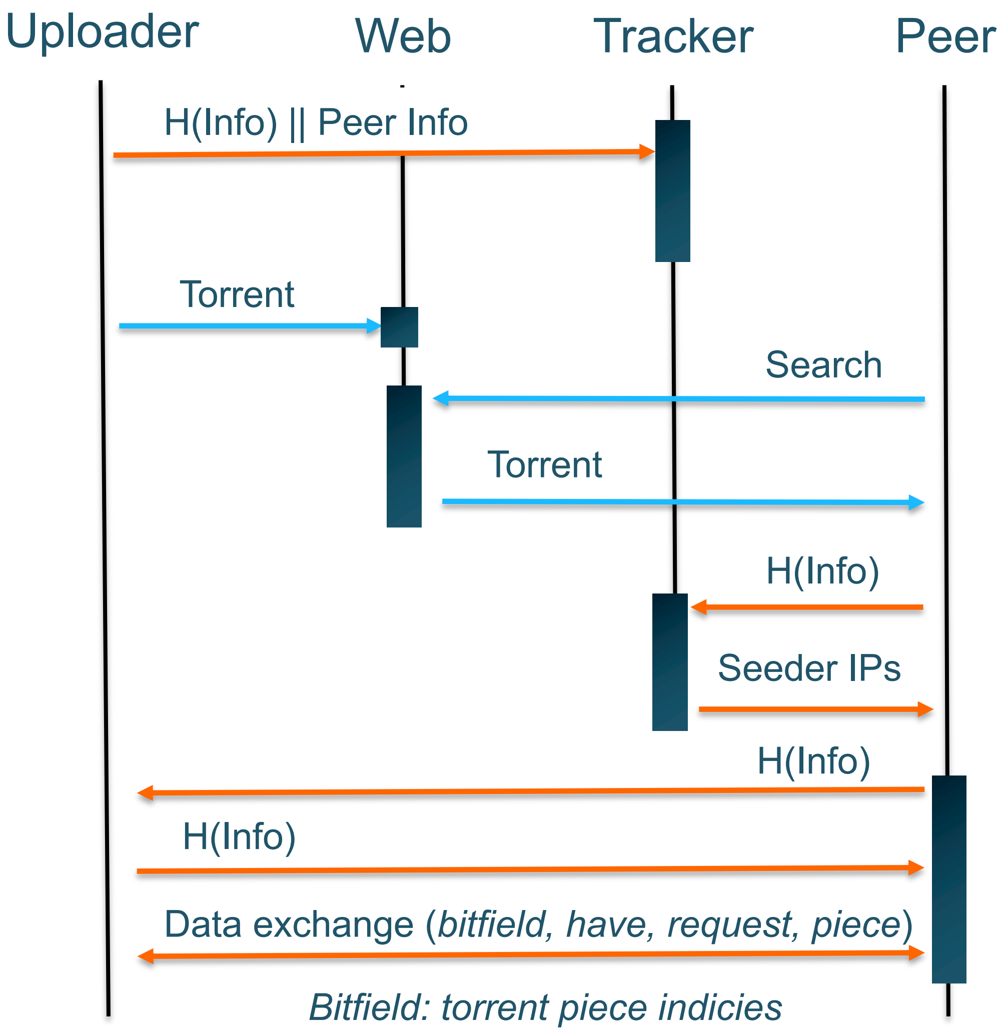
PROJECT ORIGIN
A peer-to-peer object store for CCN

Marc Mosko

Palo Alto Research Center

Dagstuhl ICN Seminar 2016

BitTorrent



Torrent

```
Announce: <tracker>
(… opt metadata …)

Info:
   piece length: <2^x>
   pieces:
      <list of SHA-1>
   name: <filename>
   length: <total bytes>
   (… opt keys …)
```

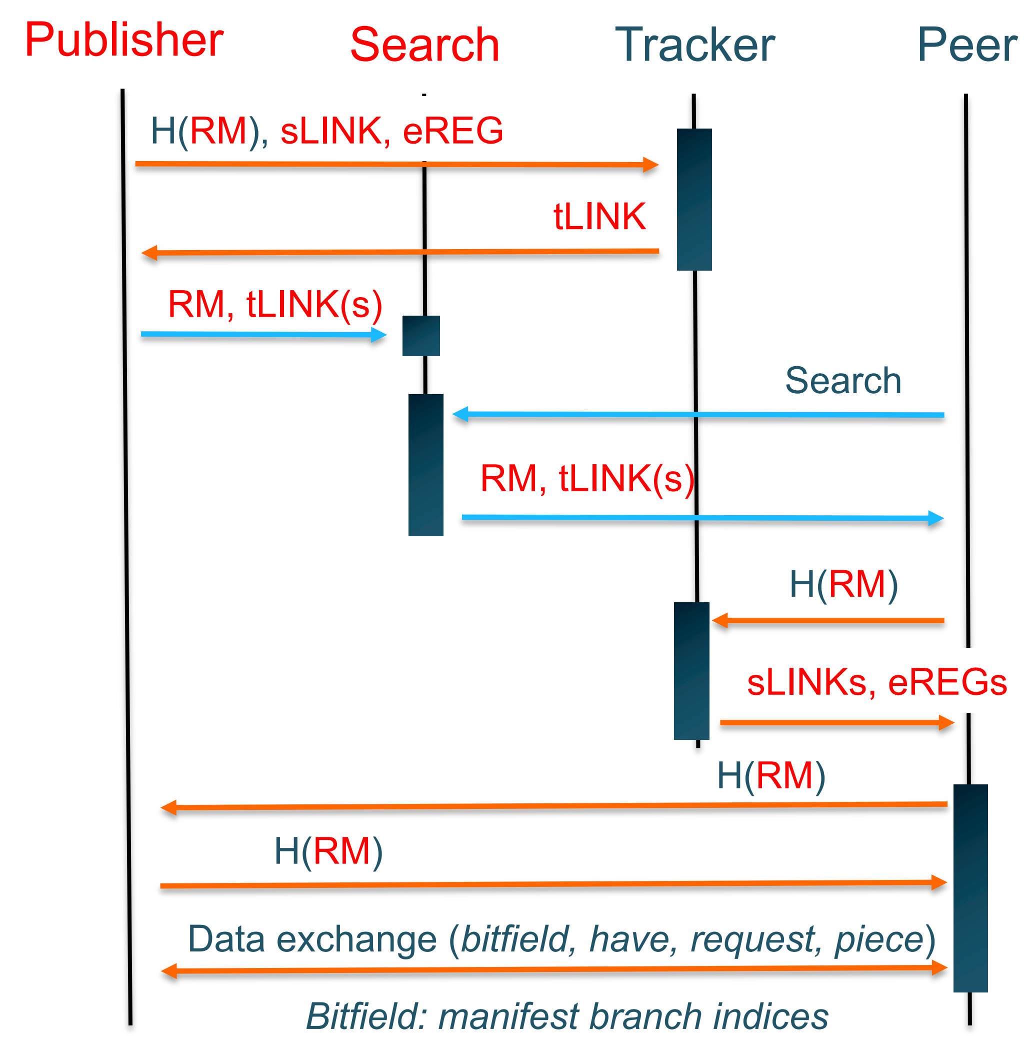
Peer Info

```
Peer_id:  <20-byte string>
IP: <implicit from TCP>
Port: <listen port>
```

→ BT Protocol

→ Other Protocol

CCNx ~~BitTorrent~~



Publisher

Search

Tracker

Peer

H(RM), sLINK, eREG

tLINK

RM, tLINK(s)

Search

RM, tLINK(s)

H(RM)

sLINKs, eREGs

H(RM)

H(RM)

Data exchange (*bitfield, have, request, piece*)

Bitfield: manifest branch indices

Root Manifest (RM) [flic]

```
Name: <data name>
(… opt metadata …)
Branches:
    size: <data length>
    hash: <SHA256>
Signature: <publisher>
```

Seeder/Tracker LINK (sLINK, tLINK)

```
Name:  <s/t name/ver>
KeyId: [<s/t keyid>]
Hash:  <H(RM)>
[eRegKey: <pubkey or link>]
Signature: <s/t sig>
```

Endpoint Registration (eREG)

```
Name:  <s name/ereg/ver>
[Tracker: <t name>]
Endpoints:
    *( eid, ename, … )
Signature: <sig>
```

Protocol

Other Protocol

Properties from LINKs

1. Seeder Links
   A. Signed by seeder with H(RM), indicates seeder accepts delegation of H(RM). Can include validity period.
   B. Name = seeder identity (matches the signing certificate).
   C. Can include the public key of the seeder (or pointer to it or cert), to be used with CCNxKE if desired.
2. Tracker Links
   A. Signed by tracker with H(RM), indicates tracker accepts delegation of H(RM). Can include validity period.
   B. Gives routable prefix of the tracker.
3. Trust
   A. KeyIds -> trust anchors (via KRS or Locator to cert). CCNxKE also usable with KeyId.

Properties (Peer Protocol)

1. Uses encoding of manifest tree to identify pieces (branches) that exist on the downloader (peer), could be more efficient than BT.
   A. Have everything is string “1”.
   B. Have half a binary tree = “010” (or “001”).
2. BT downloads chunks of pieces using (index, offset, length) tuple. Cannot verify download until piece assembled (e.g. 256 KB). CCNx has hash of each chunk, so chunk-by-chunk verification possible.
3. Downloading the hierarchical manifest is part of peer protocol, or could be downloaded via other means.
4. One could use a sync or other protocol instead.

Keeping seeders up-to-date

1. Seeder LINK
   A. Used to accept (or advertise) a RM.
   B. Can include a delegated key for verification of eREG, e.g. use an ephemeral (short) ECDSA key (valid for life of sLINK or eREGs, whichever shorter).
2. Endpoint Registration (eREG)
   A. Associates a seeder identity to zero or more network endpoints.
   B. Endpoint: routable prefix, could also have overlay info (e.g. an IP).
   C. May (should) have a validity period. May (should) identify the Tracker.
   D. Only latest version matters.
   E. A seeder may have multiple simultaneous endpoints.

Future work

1. Distributed trackers
   A. Let trackers use eREG technique too.
2. Nameless root manifests
   A. Use a technique with nameless root manifest and co-signed publisher LINK so the whole manifest can be served as a nameless object tree.
3. Analysis
   A. How do hierarchical manifests differ from linear piece list?
   B. How deep should a peer go in the manifest for the bitfield? BT usually uses 256KB for a piece. If we have 20 links/manifest to 1KB of data, stopping 2-levels up from leaves would be like 400KB pieces.
4. Implementation!
   A. We only have the design inspired by nameless objects and early PARC work on custodian routing.

THANK YOU

parc®

A Xerox Company