

A multicast based peer to peer system

SDR with SAP/SDP has been used successfully with IPv4 and v6 for years now. We envision a similar system for multicast based peer to peer file sharing. This system will be based these main components:

- A protocol for file announcements and file requests
- All announcements and requests would be sent on one or more multicast addresses
- Reply is sent by unicast/multicast
- The main datastream is either sent with multicast and a control channel only in unicast or everything in unicast.
- A multicast block should probably be allocated for these kinds of services to ease load balancing. (As this is not real time data, multicast addresses which do carry real time data can thus be given a preference if the network operator so chooses.)
- It could also, optionally, consist of a network of public key servers accessible through anycast to enable easy sharing of those keys when the need arises.

A typical file transfer situation could thus perhaps be modelled so:

1. A client node (CN) sends file request starting with hop limit 1 and increasing until it deems it has enough answers to choose from.
2. The request may contain a transfer rate and a bit pattern describing which blocks are wanted.
3. The nodes which has the requested files, referenced as server nodes (SN), replies with block length, multicast address, session ID and transfer rate
4. The CN chooses interesting SN's from their replies.
5. The CN establishes a control channel with the SN.
6. The SN sends a continous stream of packets, probably with RTP.
7. The SN does not send packets continuously from the file, but chooses randomly from the file and those blocks in the file which are most wanted have a higher probability of being chosed over those not wanted.

There are several things which must be tested in real life.

1. Multicast or unicast

- Should the file be sent with multicast and thus reducing overall network load, but with a small/large? penalty in the transfer rate due to packet loss.
- Or should the file be sent with unicast and thus increasing overall network load and congestion and this a small/large? penalty in the transfer rate?

In the interest of scalability one is lead to believe that the multicast approach would maximize the amounts of files possible to transfer with full bandwidth, as long as the files are wanted by several peers. Should only one peer be interested in the file, unicast would probably be a better protocol to ensure transfer rate, but multicast with the unicast control channel will probably serve just as well.

2. File query or announcement

How should the files be announced or queried for. One possible model is the SDR SAP/SDP model, another is the hop limit based model described earlier.

We believe that the hop limit based model is the most scalable as the queries never go further than they have to, and they allow the CN to easily compile a topological map of the answering SNs. The popularity of the unicast based peer to peer systems makes it highly probable that the amount of files wanted would be rather large, and it would probably grow even more.

A SAP/SDP based model would probably be swamped under by the amount of files offered quite quickly once the number of files passed a critical limit. Our hop limit based model has probably such a limit too, but we feel that it is on a order of magnitudes larger than the SAP/SDP model.

3. SSM or ASM multicast

Another question is if the SN should choose randomly from a set of ASM addresses, or if it should choose a random multicast address and then offer it a ASM address.

Scalability makes us prefer the last option. A random ASM address together with hop limiting would probably scale for a while, but a ransom SSM address scheme would probably scale for the foreseeable future.

4. What to send in the unicast control stream

The control stream should probably contain a public key to allow IPsec or other cryptographical signing of files and packets.

The SN can use the stream to inform the CN which packets are sent as they are sent.