1 The Idea

Using two neighboring wireless routers, on seperate internet connections, to share bandwidth seems like a low hanging fruit. There are no glaring reasons that might point to it being impossible. The goal of this project is to make two routers (Router A and Router B) collaborate when downloading http GET content (although this could extensibly be anything), in such a way that:

- All logic happens at the router, without client intervention
- Router A and B download seperate halves (using HTTP start/end) and exchange this data between each other wirelessly
- The Destination only sees the initiating router as the Src address.

To accomplish this, openWRT (an open firmware for a variety of wireless routers, most notable the linksys WRT-54 family) will be flashed onto each router. A daemon that will monitor traffic to/from each router in promiscous mode will decide when the neighbor router needs to be used (for traffic such as movies, audio, and software).Each router will inspect the Application layer of all outgoing packets, and use some of the following criteria to determine if the neighbor router's help is needed.:

- The Packet is of type HTTP GET
- The MIME type is video,audio,or application

The router will issue an HTTP HEAD request to the destination URL to determine the size of the file. Any size above a configured threshold will do. The initiating router (Router A), will create two HTTP GET requests. One for the neighbor router (Router B) to send, and one for its self. Each packet will have the HTTP Range header fields set so that each router downloads a different portion of the file.[1]

Router A will open a TCP connection with router B in order to send the HTTP packet over the wireless network. Once B recieves this packet, it will interpret the data as the packet to be sent, and replace the SRC address with it's own before sending it down the wire. The return traffic will be routed from B back to A, either over a TCP connection between the two routers, or through an entry in B's NAT table.[2]

The daemon running on router A will recieve router B's TCP responses from the server, and forge the TCP sequence number in such a way that it can be blended with the first half of the file sent to A. In practice, the two routers are facilitating something akin to parallel downloading through multiple concurrent

[1] With limited caching options on each router, I imagine the implementation of this bit will be tricky. Assembling the file in order for the client is a can of worms best left out for later

[2] libnetfilter queue is a promising userspace kernel module (compatible with openwrt) which would allow us to inspect and modify packets inside the router's packet queue. This could potentially be used to reroute the address of the return packets back to Router A, without a TCP connection.

TCP conections. PD has grown in popularity, with many infant TCP protocols spawning to support and handle it. Normally, PD involves downloading parts of a large file through multiple TCP connections on one client. Here, with router B acting as a TCP client, we get the effect of parallel downloading, but with access to more downstream bandwidth.