Appendix B
Network Resource Reservation API General Overview and Specifications

The proposed RESTful API outlined here is intended as a design for the communication of P2P network topologies to network controllers. Through this communication, a P2P tracker server (like TrackerServer.app) could attempt to reserve bandwidth resources for its operation.

Resource types and basic parameters for RESTful API:
1. Controller information
   • Total available bandwidth
   • Total available for reservation
   • Remaining bandwidth for reservation
   • Sessions (list)
2. Session
   • ID
   • Link to tracker server
   • Session total reservation
   • List of connections (links)
3. Connection
   • ID
   • Source (link to Peer resource)
   • Destinations (List of links to Peer resources)
   • Reserved Bandwidth
4. Peer
   • ID
   • Available bandwidth (upload and download)
   • IP address

The API is RESTful, which means that it operates by using the HTTP verbs to access and modify resources rather than using “commands.” Responses are also resources or lists of resources and links to resources. Many requests and responses contain JSON dictionary payloads that encode information about resources. It is also feasible for requests to optionally allow CGI argument style HTTP requests with no payload.

The Request and response formats are as follows
Request:
```json
{
  "resource-type:
  {
    "param1":value,
    "param2":value, ...
  }
}
```
Response error-free:
{
  "stat":"OK",
  "resource-name":
  {
    "param1":value, ...
  }
}

Response with error:
{
  "stat":"Error",
  "error":
  {
    "Code":123,
    "message":"Message text"
  }
}

API Requests

**GET /controller**

Parameters: None. Controller is specified by what we're contacting.

Example response:
{
  "controller":
  {
    "id":12345
    "avail-bw":x
    " reservable-bw":y
    "remaining-bw":z
    "sessions":
    {
      "id": 413847
      "link":linkToSession
    }, ...
  }
}

**GET /sessions**

Parameters: None
Response is simply a list of links to session resources.

**GET /session/<session-ID>**
Parameter in request (ID).
Example response:
{
    "session":
    {
        "id": 53143
        "tracking-addr":"127.0.0.1
        "reservation":x
        "connections":
        {
            "id": 43125
            "link":linkToConnection
        }, ...
        ]
    }
}

GET /connection/<connection-ID>
Parameter is in the request (ID).

The response is formatted much like the response above. Keys are the parameters of a Connection resource described earlier as well as a “peers” list which contains IDs and links to Peer resources.

GET /peer/<peer-ID>
Parameter is in the request (ID).

The response to this is simply a JSON dictionary like the above responses that contains the values of the Peer resource as described above.

POST /session and PUT /session/<session-ID>
Parameters are given in a JSON payload. They are the same as the above parameters of a Session resource. Any parameters not included in the payload are assigned default values or values are set by the controller.
Putting to an ID of a session that already exists causes the controller to update the parameters included in the JSON payload for the session of that ID.
The response to both of these commands is the same as for GET /session/<session-ID>

POST /connection and PUT /connection/<connection-ID>
These are calls for creating and updating connections. They require a JSON payload with the same parameters as the Connection resource described above. The two commands work just like the session POSTs. Posting to /connection creates a new connection while putting a connection ID updates a connection. The response is the same as for GET /connection/<connection-ID>

POST /peer and PUT /peer/<peer-ID>
These are calls for creating and updating peers. They require a JSON payload with the same parameters as the Peer resource described at the top. The two commands work just like the session and connection POSTs. Posting to /peer creates a new peer while putting a peer ID updates that peer. The response is the same as for GET /peer/<peer-ID>