See if I am interested in this Peer

Send ‘Request’ for a piece

Receive ‘Piece’

No ‘Piece’ Received

(because other peer coincidently reset its preferred neighbors)

Handle this case  
(How?)

Received ‘Bitfield’

Update bitfield struct for that peer

Send ‘Interested’

Send or Received Handshake to 1 client

Received ‘unchoke’

Receive ‘Choke’

Wait for ‘Unchoke’

Wait for ‘Bitfield’

Received ‘Have’  
(Who is sending it?)

Send ‘Not Interested’  
(then what?)

Listen to TCP connection

Create Thread (1 thread per client. Does both send and receive)

Thread created

Sending work

Receiving work

Receiving work  
(Red colored boxes imply that I don’t know that to do at that point)

Open in MS Word (Google Docs messes with the figure arrows)

Still interested

Not interested anymore

Idle

Receive ‘Request’

Ignore the request

Timeout: Unchoking interval

Timeout: OUI

Test Bandwidth

Idle

P nbr

Reset OU nbr

Thread represents OU nbr

Send ‘Unchoke’

Bandwidth bad

Bandwidth good

Thread doesn’t represent OU nbr

Idle

OU nbr

Send ‘Unchoke’

P nbr

Receive ‘Request’

Send ‘Piece’

Timeout: Unchoking interval

Test Bandwidth

P nbr

Idle

Send ‘Choke’

Bandwidth good

Bandwidth bad

OU nbr

Receive ‘Request’

Send ‘Piece’

Timeout: Unchoking interval

Timeout: OUI

Test Bandwidth

OU nbr

Reset OU nbr

Thread represents OU nbr

Add to P nbr list

Bandwidth bad

Bandwidth good

Thread doesn’t represent OU nbr

Idle

OU nbr

If OU nbr not in P nbr list

Timeout: OUI

Reset OU nbr

P nbr

P nbr

If OU nbr P nbr list

Send ‘Bitfield’ Conditionally

Sending work  
(Colored boxes are states)   
P nbr: preferred neighbor  
OU nbr: Optimistically Unchoked neighbor