Swarm Incentives

Questions

Q: What if a chunk is not in the nearest node and needs to be retrieved with a certain number of hops? How is it retrieved?
A: The middleman node downloads and provides the chunk. However, the chunk sum balance will cancel out: whatever the middleman pays for the chunk from one of the farther away nodes, it will charge the nodes which receive this chunk; the same will happen up until the final destination node.

Q: What if a chunk gets lost during file retrieval? What happens to the billing? (the user could pay for a number of chunks but receive less than that; in that case they would be paying for a whole file without having access to it)
A: (Aron) you only ever pay for chunks you receive. After all you are doing the accounting directly with your connected peers. However, if you get 99% of an encrypted file and the last chunk is missing, then you'd already have paid for 99% of the file even though it is of no use to you. -- see also erasure codes below.

Q: What if a file can not be completely retrieved?
A: (Dani) at the chunk level, participants have no way of knowing which chunk belongs to what file. The situation is analogous to packet loss; it's bandwidth-accounting, not file-delivery accounting. Availability could be improved by erasure codes - chunks can be recovered even if erased (research paper).

Q: Do users need to pay for knowing the price of a download? A manifest needs retrieval; in this case the user would need to pay to know how much they will have to pay to download a file.
A: Proposed solution: pre-paying for data (e.g. 100 GB) similarly to how mobile data plans work. Pre-payment is done to your own checkbook contract; not using all the pre-paid traffic when disconnecting means you can get your money back.

Q: What if we use a different message type for price requests?
A: Dani does not think we need this, it is just HTTP head request, the cost is negligible. Using *bzz* instead of *bzz-raw* will allow us to do a HTTP HEAD request, which will get the file size without actually downloading the file. An alternative is parsing JSON (manifest "size" field).

Q: Can I ask any node for the price? What if nodes have different prices?
A: In the first experiment we will (Vitalik), Read: Radical markets (book),
(aron): ???

Q: What is "fork content"? (see here)

A: Fork is a verb. You can fork content from someone else’s dapp simply by embedding the hash in your own manifest or ENS name. For example, currently theswarm.eth site is at 7c121cec08576aff9a202d3853a50b5960006fb58faa6cb9e733f12cd6a8e9c4. I can now take this hash and include it in my manifest as the hash assigned to some key, and now the exact same content will be part of my swarmsite too. I have in effect forked the website… and I can do this to lots of content without ever downloading any of it.

Q: With what price(s) do we want to start?
A: (Aron) how about 1 accounting token per bit of data?

Q: How does billed syncing affect the user when uploading?
A: If you are syncing to somebody, you are passing chunks with postage stamps on them to nodes who are closer to the address. So you are getting paid for syncing but only if they have postage stamps on them (otherwise you would be paid for spamming). Basically this means we need to add cost to uploading because uploading needs to add postage for each chunk. The cost is proportional to:

- Size of the file
- Amount of time for which you are paying
- Priority of the file (if swarm is full it would be garbage collected)
- Cheap initially - voluntary fee - as long as there is redundant storage, it will be cheap.
- Price dynamically adjusted for upload

Q: Are we using hard deposits in MVP? (initial deposit)
A: Not needed.
(Aron) what do you mean by hard deposit?

Q: Are we charging per byte or billing by chunk? Some chunks can be partially empty (e.g. the last chunk of a file).
A: We are internally billing per chunk. For encrypted and entanglement the chunks are always full (always padded).
(Aron) we should probably bill by bit. That way we don’t have to worry about what size chunks are or even if chunk size can vary now or in future.

Q: What are advantages/disadvantages of using different payment channel settlement outside of Swap (soft/hard deposits) if the latter works?

- There are some things Swap cannot do and it has very limited guarantees. However, this is only going to become relevant long after the MVP.
- If the need is justified, how do we interface with other mechanisms?
    - Through some RPC?

A: Let us distinguish between Swap (the accounting protocol itself) and the back and forth payments using a chequebook. It should be a really simple step of using Swap to account for data back and forth while letting the ‘payment’ step either be a call to the chequebook system or the raiden channel or any other settlement.

Q: What is the status of the *Generalised swap swear and swindle games* paper?
A: it is in pretty solid state up to what is written. The sections not written should be cleaned up and it can be considered complete. Nothing is ever final :)

Q: How do I monitor the economic performance of my node?
A: My suggestion would be to display the amount of traffic (e.g. in megabytes or gigabytes) you can use without contributing any resources to the network. If your node is doing well, this number will increase.

Q: How to monitor that the accounting/economing parts of the MVP are working properly?
A: Good question. I think, we will need to first define proper working (i.e. set measurable objectives) and then figure out ways of monitoring.