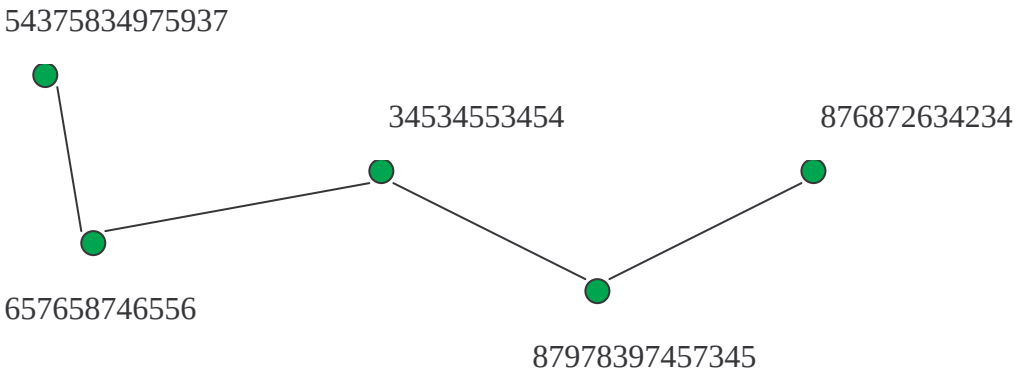
Why is the ANtsP2P DC version born?

I want to explain in a few words my last choice to create a hybrid net.
The base idea can be resumed in the following concept: if we have a net of, say, 5 nodes and if this net is secure as we intend to be secure the MUTE or the ANts net, then a node that uncovers its identity (ID/IP relationship) must not be a threat for other nodes. This is logic… how a net can be considered secure if an evil node that unveils its identity is a threat for every other node?
So actually we have this situation… as in many other net we want to give a differential service: if a node wants to be anonymous it can be anonymous, but if it wants to have fast transfers, also giving aid to other anonymous nodes it should have the possibility to choose this option.

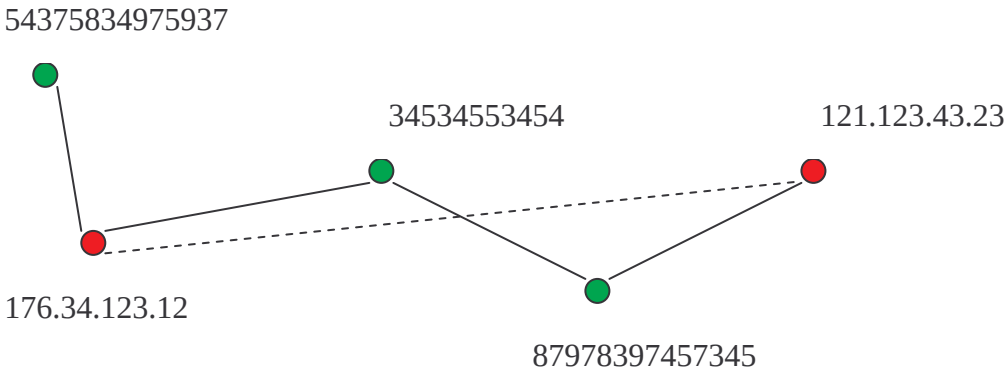
Obiously a mere bridge from emule or kazaa to Ants is not possible. Ants has a particular logic that makes nodes very uncoupled. Every communication on the net is asynchronous and based on timeouts, not the same in the most known 2nd gen p2p nets. This is why the halfnode concept has been introduced. The halfnode is the component that lets us merge the 2nd gen logic with the 3rd gen logic. A halfnode is not a simple 2nd gen node, it is much more. It can run in two distinct modalities, anonymous mode, and not anonymous mode.
A halfnode running as anonymous is nothing more than a simple Ants node as we know it. It uses ID addresses to reach destinations inside the net by means of many hops though other nodes. When a halfnode is switched to non-anonymous mode, it will expose its ip address in the messages it produces (so in the source field of a message any node will be able to see the ID and also the IP address), also when it will answer to queries it will give back to the requestor not only its ID address, but also its IP. So if the requestor is a halfnode too (that is if the query issuer can expose its IP too) a direct connection will be possible.
Actually only if BOTH the requestor and the source of a file have selected the non-anonymous running modality a direct connection will be extablished, otherwise everything will work the same way it works in the classic Ants network, and the anonymity of those not exposig their ips will be preserved.
A very interesting thing is the side effect produced by this approach… as soon as the net begin exchanging messages, node will collect infos about the topology as we know, so we will have halfnodes and anonymous nodes running together and also working together in a way that improve transfers in a significant way… why? Because by means of the hafnodes in the net, we will be able to have the so called “far hops”… if you request a file from a source that is far and that expose its IP address and if you want to remain anonymous, we said everything will work the same way it worked in the non DC network… this is not completely true. Infact if the request message travels through a non-anonymous node this will be bounced directly to the source of the file, thus improving a lot the transfer speed, without harming the requestor privacy. In a dual way, if the requestor exposes its IP address but the source wants to remain anonymous, we will have improvement too, cuz it’s likely that the messages from the source to the requestor will find shorter ways being bounced directly to the requestor itself (though the source privacy is preserved).

Let’s give some pictures of this:



Using the normal Ants algorithm 4 hops are needed to get from 54375834975937
to the file source 876872634234.

But if we have some nodes exposing their IP addresses for some reason…



Once a message gets from the node 543… to the node
176.34.123.12, then this is able to create a direct connection to the source, shorting up the overall hops number.

The same way of reasoning is possible for the dual case where the node 54375834975937
Is the source, while the node 121.123.43.23 is the requestor, not interested in preserving anonymity. The source will preserve its anonymity, because it doesn’t expose its IP address, while the requestor will be able to get better speeds, due to the direct proxying performed by the node 176.34.123.12.

So actually these are the main changes introduced since protocol 0.4.9s. Everything has to be tested and made working, cuz the protocol probably is very buggy, many bugs are still present from the former changes so we have to fix both old and newer bugs… this will require time, but I’m sure that the final result will be a very fast network, for both non-anonymous and anonymous purpouses.