辕珣开发文档一：
Week 1-2. Strongly typed interfaces and bare-bones implementations will be completed for Protocol A, B, and C. There will
be simple proofs-of-concept for the file-management component, the network-management component, and the verification
component. It is understood that the rules for verification may not be permanently established by this point. Thus, a
modular framework for verification must be provided so that the rules can be easily changed.

Week 3-4. An executable prototype will be ready for launch. The first test-net will be launched by the end of the second
week. Even within a given iteration of Protocol A, B, and C, there can be several test-nets, each of them with a
specific genesis block which may be triggered based on a command-line option. There will be finite-state-machine
routines for introspecting the node implementations and verifying their correctness. There will be a stripped down ASCII
notation for network transactions based on nc(1) so that testing individual transactions by hand may be simplified.

Week 5. At this point, there will be a series of refinements to the existing code and well as experiments with different
versions of protocols. For each version of Protocol A, B, and C, a separate set of headers can be made, and swapped into
the compilation folder so that they can be included during the compilation process. There will be active communication
Dr Xu Gang and others within the Bitcoin community, such as the personnel of Wu Jihan, to ensure that protocols arrived
at may satisfy the wishes of various parties and be easy to configure for, and to build against.

Week 6. Further experiments will continue. There will be rigorous testing by dedicated unit-testers and integration-
testers at the company. At this point, the details of deployment will be considered, as well as the ease of
verification, and the ability to deploy the program to end-users, and also remotely, and in an automated fashion. The
author aims that the resulting program may be distributed as a single executable.

The resulting executable will target two major platforms, Windows AMD64 and Linux i686. The former was chosen because it
is the platform of choice for miners and node operators within the PRC. The

latter was chosen because it is, by the
estimation of the author, the most stable platform, and the one which he would prefer to use for test-nets during
iterative experimentation and testing, and for the seed-servers for public use, upon there being an authorized version
of ELA scheduled for public release.

Protocols

The following protocols exist for BTC and for all node programs of a similar nature: Protocol A, which describes the
structure of the block-chain and the procedure needed for verification, 2) Protocol B, which describes the means by
which an application, including a mining program, can communicate with a node, and Protocol C, which describes the
means of communication between two nodes.

Protocol C is the most complex because it concerns low-level details of transmission along the network, including
difficulties with the discovery of peers and with timing, which is to say, the maximum allowable time elapsed between
communications before changes occur in the network, such as peers ejecting one another or deeming one another invalid.

Protocol C under ELA will be subject to attentional requirements, namely:

1) BTC does not provide a means for UDP hole punching, but this is a necessary feature for casual end-users to be able
to run their nodes upon home computers, so that they may be plugged into the network, thereby increasing the network's
capacity, robustness, and interconnectedness, which may further facilitate communication with applications such as
wallets, commerce programs, and visualization programs.

2) BTC uses magic identifying strings in order to initiate connections, but this carries the risk that a motivated
obstructor, such as the operator of a firewall, may use detection routines to systematically preempt connections
relating to BTC, thereby disrupting the flow of traffic between residents of a given domain and residents outside of
that domain. Certain techniques need to be utilized in order to avoid the use of magic identifying strings. Having a
rotating set of identifying strings may not be enough. A viable solution might require the use of a rotating cipher and

a series of keys which may be selectively tested against unknown inputs to derive meaning.