QL Network Protocols

Standard QL Handshake

The Standard QL handshaking network protocol is compatible with the Sinclair Spectrum protocol. It comprises 11 phases

	sender	**receiver**
a) scout		
1) gap	waiting for 3ms for activity, if activity occurs: restart	
2) wait		waiting for activity (a scout)
3) scout	send a scout of duration < 530us, if contention occurs: restart	wait for 530us
b) header		
4) hactiv	set net active 22us	wait for active
5) hbytes	for each byte 11.2us start (inactive) bit, 8*11.2us data bits, 5*11.2us stop (active) bits	for each byte wait for start (inactive) bit, read 8 data bits, if fails: restart
6) hackw	wait for 2.5ms for active, if not active: restart	set net active 22us
7) hackbt	wait for start bit, read 8 data bits, if error: restart	send 11.2us start bit 8 data bits 00000001
c) data		
8) dactiv	set net active 22us	wait for active
9) dbytes	for each byte 11.2us start (inactive) bit, 8*11.2us data bits, 5*11.2us stop (active) bits	for each byte wait for start (inactive) bit, read 8 data bits, if fails: restart
10) dackw	wait for 2.5ms for active, if not active:	set net active 22us

```
                restart

  11) dackbt    wait for start bit,        send 11.2us start bit
                read 8 data bits,          8 data bits 00000001
                if error: restart
```

The entire protocol is synchronised by a period of inactivity at least 2.8ms long.

The header is eight bytes long in the following format:

destination station number
sending station number
block number (high byte)
block number (low byte)
block type (0 normal, 1=last block of file)
number of bytes in block (0 to 255)
data checksum
header checksum

If the number of bytes in a block is 0, 256 data bytes are actually sent.

The checksums are formed by simple addition: if there are two single bit errors in the most significant bit (the most common type of error) within one block, then the errors will pass undetected.

If the block number received in a header is not equal to the block number required, then the header and data block are acknowledged but ignored.

The protocol is not proof against a failure on the last block transmitted where the receiver has accepted the block, but the sender has missed the acknowledge. In this case the sender will keep re-transmitting the block until it times out (about 20s).

Toolkit II Broadcast

Toolkit II has a special version of this protocol for network broadcast. This has an extended scout to allow time for the receiver to interrogate the IPC without missing the scout, and it has an active acknowledge / not acknowledge. The protocol has been defined in such a way that future network drivers can be more flexible than the Toolkit II driver.

```
                sender                     receiver
a) scout

  1) gap        waiting for 3ms for
                activity, if activity
                occurs: restart

  2) wait       waiting for activity
                (a scout) every 20ms
                check IPC for BREAK
```

3) scout	send a scout of duration < 530us, if contention occurs: restart	wait for 530us
4) scext	send a scout extension of 5ms active	
b) header		
5) hbytes	for each byte 11.2us start (inactive) bit, 8*11.2us data bits, 5*11.2us stop (active) bits	for each byte wait for start (inactive) bit, read 8 data bits, if fails: nack
6) hwait	leaving net active, wait 1ms	
c) data		
7) dbytes	for each byte 11.2us start (inactive) bit, 8*11.2us data bits, 5*11.2us stop (active) bits	for each byte wait for start (inactive) bit, read 8 data bits if fails: nack
8) dack	inactivate net and wait 1ms for active: if fails, restart	within 500us set net active and wait 5ms, do any processing required and when ready for next packet, inactivate and restart
d) Not acknowledge		
9) nack	wait for inactive	wait for 2.8us of active or inactive, if inactive: restart
10) nackw	wait 500us for active: timeout is ok, active is fail	wait 200us for active, if active: restart, if inactive activate 500us (nack)

A broadcast acknowledge is 5ms active followed by more than 400us inactive. A broadcast not acknowledge is no response or 5ms active followed by 200us to 300us inactive, followed by more than 200us active.

Toolkit II Server Protocol

The Toolkit II server protocol is physically the same as the Standard QL protocol, but the header has been slightly changed to improve the checksum, to allow blocks of up to 1000 bytes to be sent and to distinguish server transactions. A server header cannot be confused with a standard header.