C-THRU™ FAQ

$Rev: 8$

This document provides answers to frequently asked questions about C-THRU. It is broken down into the following types of questions:

Technical Questions

What platforms does C-THRU support?

At the moment C-THRU supports any POSIX system. It was developed on NetBSD and written in C, and great care has been taken with respect to portability. The client contains no NetBSD or Linux-specific code and is ISO C90, ANSI C89 and POSIX compliant.

What other operating systems can C-THRU support?

C-THRU can be ported easily to any POSIX compliant operating system as required; it was developed with portability in mind. The front end of the library and various components within it have been designed modularly so that porting the library to other non-POSIX operating systems is straightforward.

Is there a version of C-THRU for Windows?

There is currently not a version of C-THRU for Windows. This will be undertaken as required by customers.

Can C-THRU be linked to a non-threaded application?

Yes, it does not matter if the software C-THRU is linked to does or does not use threads, as long as the operating system C-THRU is running on supports threading.

Is there a Java compatible version of C-THRU?

No. However is would be possible to develop a Java version of C-THRU with a re-write of the client side library in Java.

What protocols can C-THRU support?

C-THRU is protocol agnostic: so it supports any protocol that can be run over any type of network which C-THRU can support. (Despite the company's name, C-THRU is not in any way specific to VoIP, but can be used to improve the quality of VoIP systems too).

What API does C-THRU present?

Currently, the Berkeley socket API as defined by POSIX, introducing a new protocol family called `PF_CTHRU` (as opposed to `PF_INET`).

What changes need to be made to link C-THRU to an application?

Very little. C-THRU is designed so that the core features of the library can be used with minimum fuss for the developer. Typically this will require about two lines of code to be changed before an application or piece of software is compiled with C-THRU.

Bubblephone Limited
Sussex Innovation Centre
Science Park Square
University of Sussex
Brighton BN1 9SB
Tel +44 (0) 1273 704 535
http://www.bubblephone.com/

Do changes always need to be made to an application to use C-THRU?

No, not in all cases. There are several ways to make use of C-THRU; whether the application needs to be modified or not depends on the reasons why the application is interfacing with C-THRU. Typically:

- No application features are being added and C-THRU is being used for transparent benefits. A proxy style interface fulfils these requirements; see the use-cases document for bridging, routing and application-layer proxies. In these situations it is the proxy which is linked against C-THRU, not the applications directly.
- For features added to an application, such as codec scaling based on C-THRU's feedback to a VoIP application, the application will need modifying anyway. Alongside the new features, interface code will need to be added to acquire data to be passed along to these features.

Hence any changes made to interface with C-THRU are in situations where the application would be modified anyway, for the purpose of adding new features. Otherwise, the benefits of adding C-THRU are transparent. As each application's requirements differ, please contact us to discuss options.

How does C-THRU select between different Connection Methods?

Based on what C-THRU has deduced about the network resources available, it is able to select the Connection Method that best matches the application's requirements. It is able to switch connection types and even networks in real time to maintain the best possible connection.

This process is referred to as Method Selection and it is performed by the method selection algorithm. Method Selection operates by graph reduction of the possible search space and is itself very efficient.

How are requirements passed to C-THRU?

Developer and end-user requirements are passed to the method selection algorithm via C-THRU's API.

What characteristics of the traffic can be prioritised?

If a characteristic of the connection can be defined, then it can be considered in the decision making of the method selection algorithm.

Popular characteristics include:

- Monetary cost of a particular connection per unit of data.
- The bandwidth of a connection.
- The available bandwidth on a connection real time.
- Heuristic stenography by frequency distribution.
- The need to traverse firewalls.
- The packet sizes on that connection.
- Optimal data compression over that protocol for type of data sent.

What connection methods are available?

A core set of Connection Methods have been developed, each of these representing a de-facto way in which data can be tunnelled. More are being developed all the time; Connection Methods are also being developed specifically to meet individual customer requirements.

Can the application request data about network characteristics?

Yes, any information that C-THRU can categorise can be fed back to the application if requested. One example that has proven particularly popular is bandwidth use. This can allow application specific choices to be made, such as only sending essential data or changing the compression ratio for scalable traffic such as video streams. This can change on the fly, throughout the lifetime of the session.

What happens if C-THRU switches from one type of network to another?

The application will continue to perform without interruption.

Switching between networks types, e.g. from Wi-Fi to TETRA is invisible to the application. C-THRU is able to switch Connection Methods across networks and network types seamlessly during data transmission, and do this transparently from the application’s perspective.

Does C-THRU's searching cause a lot of network traffic?

No. Because C-THRU sits between the operating system's traffic management and the application, often it is able to infer the majority of the information it needs from attributes of the traffic, not by sending explicit probes.

Can I use C-THRU to reduce my network traffic?

Yes, in a variety of different ways. Please talk to us about it and we'll help see what fits your requirements.

How big is the C-THRU client library?

C-THRU is designed to run on embedded systems; it is small and uses very little RAM. It also requires a single file for storage.

Does C-THRU provide encryption?

No. C-THRU is a socket library, and encryption is not part of the socket API. This leaves the developer free to use any form of encryption they require.

Can C-THRU help with addressing issues?

Yes, including the following which we have encountered so far:

- Interoperability between segregated networks
- Joining networks which use different address families
- Flattening complex address hierarchies
- NAT traversal and equivalents for non-IP

C-THRU does this by providing its own address system, `AF_CTHRU`, which sits on top of the existing address family(s). C-THRU's address family acts as an overlay network over the underlying connection methods, which allows these networks to then communicate directly as they are all within C-THRU's address space.

This feature is optional and fits transparently within the existing 32 bit IPv4 space, so can support up to four billion devices. Alternatively the underlying address families (such as `AF_INET`) may be used as usual.

Can C-THRU roam?

Yes, including across different physical network types, using `AF_CTHRU` addresses to join the different address spaces. E.g. it could roam between two IP connections or from a satellite connection to TETRA without interruption to the application's connection.

So does C-THRU support P2P applications?

Yes, all client to client communications are peer-to-peer. C-THRU supports all the typical peer-to-peer styles of communication, including supernoding and other more unusual methods of establishing connections if necessary.

What ISO and safety standards does your code conform to?

Currently none, however, the code is written to very strict internal company standards to make this process as smooth as possible when required by a customer.

Logistical Questions

What changes or special equipment is required to roll-out a C-THRU application?

No changes are required for application roll-out; the application is linked against C-THRU and distributed in the normal fashion.

In addition a hardware device called a 'Cache' will need to reside on the customer's network.

What is the Cache and what does it do?

The cache passes updates and new methods to the C-THRU clients at run time. It also keeps a track of the client instances of C-THRU and takes logs of the types of traffic and any desired network trends. These can be presented as logs if required.

How is C-THRU or the application updated?

The application is updated in the usual way the customer would always update their application, including the version of C-THRU it is linked to if required. In addition, the copy of C-THRU the client is linked to can be updated automatically at run time with a download from the Cache. This is an optional feature and is preformed seamlessly in the background without affecting the application's connection.

Any update would be a controlled process so as to ensure compatibility of the components being updated and continuity of the running application.

Does Bubblephone sell copies of the entire system, not just the client and cache?

Yes. A central server administered by Bubblephone sits above the customer's cache and passes down updates and receives logs of relevant data. All this infrastructure may be provided.

Can Bubblephone tell if someone is spoofing clients?

Yes.

What else can be monitored?

Bubblephone is able to monitor the use of threads by the application linked to the C-THRU client. If threads are being used incorrectly by that application, Bubblephone is able to advise the client of this.

Do clients continue operating if the Cache is temporarily unavailable?

Yes.

Business Questions

What intellectual property protection does Bubblephone operate?

Bubblephone has a full IPR protection scheme in place and will be issuing patents on C-THRU and the various technologies it employs before it goes to market.

How easy is it to patent C-THRU's IPR?

Bubblephone is in a tactically advantageous position with regards to patents: Bubblephone is able to patent the algorithms themselves, not the specific implementations of the software; which can often prove difficult or impossible, giving a far wider coverage of patent protection.

Can Bubblephone prove an infringement of C-THRU's IPR or patents?

Bubblephone is in an unusually strong position as it can identify IPR and patent infringement by watching competitors' traffic; C-THRU's movement of traffic is unique so infringement would be easy to spot without the need to look at source code.

Has Bubblephone carried out a full patents search?

Yes. There are no known existing technologies similar to C-THRU. Bubblephone has conducted two separate patent searches and employees keep a continual lookout for potential infringements. So far none have been identified in the years since the company was founded.

What about market competitors?

There are companies in the market that address some of the features/capabilities of C-THRU but not all. For example, companies providing protocol compression can improve connection performance and throughput. However the same companies cannot address connection robustness and other C-THRU capabilities in which case C-THRU would further enhance their value propositions.

Competitors have been approached and most seem interested in potential cooperation as C-THRU is complementary to the majority of existing technologies on the market.

A Harvey Balls competitor comparison chart is available on request, for a detailed analysis of competitors against C-THRU.