PROTOCOL ARCHITECTURES TO EFFICIENTLY DELIVER IP SERVICES OVER HF RADIO

Steve Kille
Isode Ltd.
14 Castle Mews, Hampton
TW12 2NP
United Kingdom
steve.kille@isode.com

SUMMARY

There is broad interest to provide a range of modern IP applications services over HF Radio. NATO have standardized an "IP Client" protocol in STANAG 5066[1] Annex F.12 which is a mandatory part of STANAG 5066 and can be used to deliver any IP service over HF Radio. This paper starts with an analysis of using this protocol and explains why it is always sub-optimal and for very many applications will deliver unusably poor performance.

Then the paper looks at an approach to delivery XMPP services over HF Radio which worked well down to 300 bits per second in a trial funded by UK MoD and reported on it in a presentation "UK MOD XMPP OVER HF PILOT" to the HF Industry Association in February 2019[2]. The architectural choices needed to achieve this performance are considered.

Providing an approach that will work for a wider range of applications is considered, leading to analysis of the STANAG 5066 ARQ reliable data transfer service. The paper looks at how this basic service can be extended to provide efficient general-purpose datagram and streaming services, which Isode has published in the open specification "SIS Layer Extension Protocol (SLEP)" (S5066-APP3)[3]. The paper explains why this extra layer is necessary, and how it can be used to provide efficient delivery of a wide range of IP Services over HF.

1 GOALS FOR IP SERVICES OVER HF

There are a myriad of applications that run over IP, many of which could be operated over HF Radio and would be useful. This paper looks at architectures to achieve this, some of which optimize applications and others which do not require the end application to be changed.

2 STANAG 5066 IP CLIENT

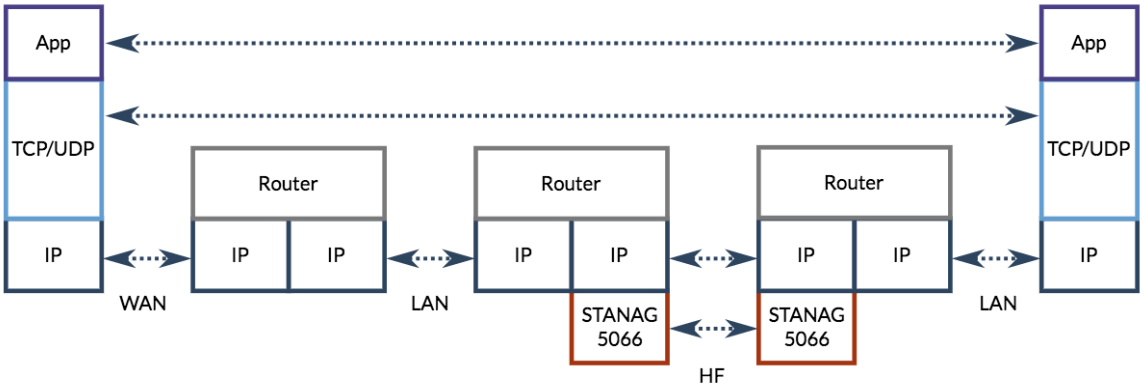

Figure 1: STANAG 5066 IP Client Architecture

STANAG 5066 is an open standard link layer that operates over HF Radio. It is the preferred NATO standard and the focus of this paper. It is anticipated that very similar considerations would apply to any other well designed HF link layer. STANAG 5066 defines an "IP Client" protocol that supports an IP subnet using the standard IP model, illustrated in Figure 1.

IP Client is built on the STANAG 5066 Unidata service which transfers blocks of data over an HF network. IP Client encapsulates each IP datagram in a single Unidata service request, which provides transfer over the HF network. Delay is highly variable, with longer transfer delays measured in minutes. Delays play a key role in overall performance. Factors include:

- Other nodes transmitting on the channel. There can be multiple nodes sharing a channel with transmissions lasting up to 127.5 seconds.
- Other locally generated traffic, which may be IP traffic from the same or other applications or non-IP applications running over STANAG 5066. The amount of buffering/delay is a STANAG 5066 implementation choice. Commonly this is around 2 minutes (to allow for maximum transmission length), although some servers buffer more.
- When local buffering is full, STANAG 5066 will flow control the IP client. The IP Client needs to choose how much (if any) traffic to buffer. Implications of the choice:
    - If IP packets are discarded, data will generally need to be retransmitted later.
    - If IP packets are not discarded, latency will increase.

There are two modes of STANAG 5066 data transfer used, and IP Client may choose either with different service implications:

- Non-ARQ. Unreliable transfer, so data may be lost; or
- ARQ. Reliable transfer, which can lead to extended delays if retransmission is needed. If there are CAS-1 soft link outages, transfers will be rejected. IP client may choose to re-queue rejected IP datagrams.

The delays and errors described will impact STANAG 5066 IP Client and lead to IP transfer with long delays and complex loss characteristics, with performance dependent on the detailed options chosen. This is very different to most link level services, and has significant implications on the layers above.

Use of IP Client with UDP and TCP services is now considered; UDP and TCP are the dominant protocols used over IP and most IP applications suitable for HF Radio use them.

2.1 UDP SERVICES OVER IP CLIENT

User Datagram Protocol (UDP) provides an unreliable datagram service over IP. Overheads for a small datagram are 20-60 bytes for IPv4 or 40 bytes for IPv6 plus 8 bytes for UDP. This is an acceptable overhead, although quite large in comparison to protocols designed for HF. Three types of UDP application are considered.

The first type of applications is custom military command and control applications that run over UDP. These can work well, provided that an ARQ mapping for IP Client is chosen to minimize message loss.

A number of IP services such as Domain Name Service (DNS) use UDP, to avoid the overheads of TCP. DNS is considered as an example of this second class of application. To give reliability, queries are retried if no answer is received. The DNS standard specified a minimum retry delay of 5 seconds, which is often chosen. For operation over HF, the longer delays will mean that queries and responses will generally be repeated several times, which is sub-optimal.

ACP 142[4] is a bulk application, chosen as an example of a third class of UDP application. ACP 142 a multicast standard used by modern military messaging, which can operate over UDP. To transmit over UDP, a fixed transmission speed needs to be chosen. This can lead to poor performance over HF, due to outages and highly variable speeds. Fortunately, ACP 142 can be operated directly over STANAG 5066. This removes the UDP overheads and enables use of STANAG 5066 flow control to match transmission rate to current HF channel capacity.

Only the first group of UDP applications has reasonable performance over IP Client.

2.2 TCP SERVICES OVER IP CLIENT

Transmission Control Protocol (TCP) is a reliable stream protocol used for the majority of Internet traffic and by most of the protocols that are anticipated to be operated over HF Radio. Measurements[5] suggest that TCP over IP Client can be very inefficient. Reasons for this are considered here. TCP headers are 20-60 bytes, and for each data PDU there will be an ack. For a typical 512 byte data packet, this gives minimum overhead of 15% for IPv4 and 24% for IPv6. This base overhead is quite high, but may be considered acceptable for some deployments. It is anticipated that operation with this overhead is achievable in good HF conditions where the applied TCP load is significantly less than modem speed.

As load increases or there are data losses due to HF errors, two aspects of TCP operation come into play. TCP uses a simple window mechanism to detect what data has been received. When the sender believes that data has not been received, it will retransmit data from above the lower window edge. In busy HF networks traffic delays occur, and TCP will retransmit data that is delayed, leading to overhead due to data duplication.

The second factor is that TCP deals with data loss by reducing the window size. This is an excellent mechanism to deal with traffic congestion in a fast network. For a high latency network, this is disastrous as it leads to short transmissions and collapse of throughput. In order to get good throughput at high HF speeds, Isode has proposed an extension to STANAG 5066 ARQ, which also uses a windowing mechanism, to increase the maximum window size[6].

2.2.1 Comparison of TCP and STANAG 5066 ARQ Windowing

STANAG 5066 ARQ uses a windowing mechanism that is optimized for slow networks with potentially high data loss. On each ARQ transmission, the receiver will report back on its view of Lower Window Edge and the received/not received status of PDUs between Lower and Upper Window Edges. A good STANAG 5066 implementation will repeat transmission of this information to minimize risk of it not getting through. This approach enables the sender to rapidly retransmit missing PDUs and not transmit duplicates.

TCP windowing acknowledges the Lower Window Edge only. Choice to retransmit is driven by timers, which with default TCP settings leads to significant PDU duplication once HF delays build up. There does not appear to be a satisfactory solution to using the TCP window mechanism over HF, and the solution described later is to use the STANAG 5066 ARQ service.

3 WHY A DIFFERENT ARCHITECTURE IS NEEDED

Applications running over STANAG 5066 IP Client can work reasonably well if the applied load is significantly less than link capacity and link quality is high. This will rarely be the case for narrow band HF, particularly when link quality is poor or when there are transmission gaps. A different approach is needed that will support common use cases such as Web browsing.

4 DEPLOYING XMPP OVER HF

XMPP is the open standard for instant messaging and presence standardized by the XMPP Standards Foundation[7]. XMPP is increasingly being used for military deployments, such as the NATO JCHAT service. It is a natural service to run over HF Radio, because of wide military adoption and small messages making it suitable for constrained networking.

4.1 UK MOD TRIAL RESULTS

UK MoD funded a trial of XMPP over HF, reported on in a presentation "UK MOD XMPP OVER HF PILOT" to the HF Industry Association in February 2019[2]. This demonstrated 1:1 chat and group chat. Extended tests were run at 600 bps and 4800 bps, showing typical application latency of 10 seconds and 3 seconds respectively. Operation was demonstrated at 300 bps, where it worked nicely. The system dealt with transmission failures, short outages (a few minutes) and extended outages. The trial was viewed as a success and it was commented that “XMPP worked, even when HF didn’t”.

4.2 WHY THE RESULTS WERE GOOD

There were three key reasons why the XMPP trial results were good:

1. Direct operation over STANAG 5066. XMPP communication over HF works directly over STANAG 5066 following XEP-0365[8].
2. Elimination of hand shaking. Standard XMPP protocols are asynchronous in operation, but have significant handshaking on startup, which has a major impact on performance

over high latency networks where connections are not stable. This is eliminated by following XEP-0361[9].

3. Application optimization. Various optimization made to reduce message size and to eliminate sending some optional messages. In particular, group chat communication was optimized using Federated MUC specified in XEP-0289[10].

Operation of standard XMPP protocols over TCP and STANAG 5066 IP Client would not have given acceptable results. There was a need to both eliminate an inefficient protocol layer and to optimize the application. Although special protocols were used over HF, the rest of the system was able to be operated using standard XMPP over TCP.

5 PERFORMANCE ENHANCING PROXY (PEP)

The XMPP trial used a Performance Enhancing Proxy (PEP) model. The PEP concept is widely used and set out in RFC 3135[11], which applies it particularly to TCP. PEP can be applied at many levels. PEP used with TCP has been important for satellite and other high latency networks. RFC 3135 explains both the benefits and downsides of a PEP approach. The essence of PEP is to use an optimized protocol over a constrained link, and to break end to end communication. This improves performance but does introduce some problems not present in an end to end system, in particular new failure modes. For HF, a PEP approach is essential, as the end to end approach using STANAG 5066 IP Client is not operationally viable for many target deployments.

6 APPLICATION LEVEL PEP

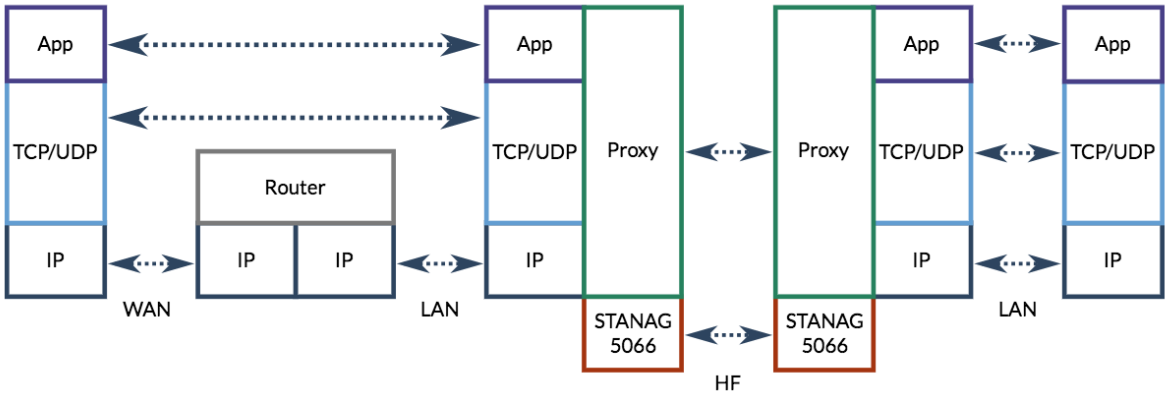

Figure 2: Application Level PEP Architecture

The XMPP system described operates as an application level PEP, illustrated in Figure 2. For some protocols, this has significant advantages over the TCP level PEP, which is discussed later. Use of an application proxy is highly desirable where it can eliminate hand-shaking and other overhead in the general purpose protocol. It is of particular benefit for XMPP and messaging, where significant optimizations can be made over standard XMPP protocols and messaging protocols. There are number of messaging protocols for providing messaging application level PEP[12].

Developing application-specific proxy protocols is significant effort, and this needs to be justified in terms of the benefit that can be achieved and the value of operating the application over HF.

6.1 TRANSPARENCY & OTHER BEARERS

Unlike with TCP PEP, use of an application level PEP generally needs to be visible to the end server application, because the application will authenticate its peer. This can be done easily for messaging, where relay is standard, using MX DNS Records for SMTP or message routing configuration for STANAG 4406 messaging. For XMPP, the XMPP Trunking[13] approach is used. For clients, the use of PEP is transparent.

Using Application Level PEP means that switching to other bearers (e.g., switching between HF and satellite) needs to be done at the application level. Generally, the best way for this to happen is for the application proxy to switch bearers, which can be automatic and transparent to the end application. For example, a message switch could shift between ACP 142 over HF and SMTP over satellite for a link, dependent on conditions.

7 SLEP: A BUILDING BLOCK FOR RELIABLE APPLICATIONS OVER HF

There is a need to provide a generic approach to addressing the IP Client performance issues noted, without having to develop a special protocol for every application. Modern application protocols are developed in a layered architecture. What is needed is a layer service that is efficient for HF and can be used directly by standard IP applications. TCP applications are the most important to address.

7.1 WHY STANAG 5066 ALONE IS INSUFFICIENT

STANAG 5066 ARQ provides much of the service provided by TCP and it seems attractive to use this capability and avoid duplicating it by running TCP over ARQ.

STANAG 5066 provides the Unidata service which is too low level for direct use by standard applications. The RCOP (Reliable Connection Oriented Protocol) and UDOP (Unreliable Datagram Oriented Protocol) are attempts to extend the STANAG 5066 framework. UDOP is a viable replacement for the Internet UDP.

RCOP attempts to provide a reliable datagram service. One problem is that reliable datagram is not used by many applications; most use the TCP streaming service. A more basic problem is that RCOP does not provide a fully reliable service. This means that applications building on RCOP need to allow for datagram loss, and must retransmit in the event of failure. This means that RCOP is really only suitable for quite small datagrams. The most significant failure conditions not handled are when:

1. The receiving application is not listening, causing data to be acknowledged, but not delivered.
2. The CAS-1 soft link fails during transfer, leading to ARQ state reset and Unidata rejections.
3. The STANAG 50666 server restarts during transfer, leading to data loss.

7.2 DATAGRAM AND STREAM SERVICES USING SLEP

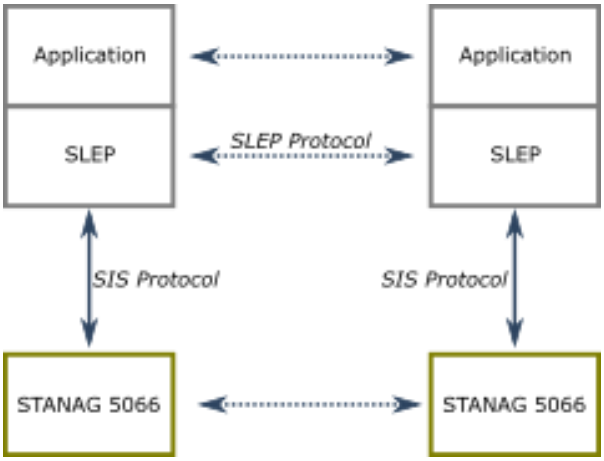

Figure 3: SLEP Architecture

SIS Layer Extension Protocol (SLEP)[3] defines three services:

- Unreliable Datagram, which offers some minor improvements over UDOP
- Reliable Datagram.
- Stream. A service equivalent to TCP.

The SLEP services are used by an application, which communicates with a peer as shown in Figure 3, and uses the STANAG 5066 SIS protocol to communicate with a STANAG 5066 server.

These services use a common set of compact PDUs and operates directly over STANAG 5066. The datagram services use compression (if it makes the PDU smaller) and compression is optional for the stream service.

All three services share common protocol. Data PDUs can be numbered and associated with identified transfers. There is a set of control PDUs. All three services provide flow control to the application. The application sets priority for data, and the SLEP services transfers data strictly in priority order. This enables higher priority data to overtake lower priority.

Unreliable Datagram will send small datagrams as a single non-ARQ Unidata. Transmissions may be repeated by STANAG 5066. Larger datagrams get Transfer IDs and are split over a set of numbered blocks.

Reliable datagram splits the datagram into a set of numbered blocks with a transfer ID, which are transferred with ARQ Unidata. The datagram is acknowledged, to guarantee reliability. The receiver may request resend of missing blocks. The sender can probe the receiver, when data is not acknowledged. This is all controlled by a number of timeouts. In normal operation,

operation the ARQ service does everything needed, without use of the timers. The timers are needed to deal with possible failures, that will usually be infrequent.

The stream service starts with a handshake to establish either unidirectional or bidirectional flow, with a Transfer ID for each direction. Data blocks are numbered with a three byte integer, which wraps and so does not limit transfer volume. Data blocks are transferred with ARQ Unidata and un-ordered delivery. Missing data blocks can be requested by data receiver. The close mechanism can be a clean close with all data transferred or an abort. As with reliable datagram, timers are used to deal with possible (but infrequent) errors.

8 DEPLOYING TCP APPLICATIONS

The SLEP Stream Service is functionally equivalent to TCP, but operates efficiently over the STANAG 5066 ARQ service. It can be used in two ways to support TCP applications and avoid the performance issues with IP Client.

8.1 DIRECT USE OF SLEP STREAM SERVICE

The SLEP services can be embedded directly in an application. The SLEP specification defines how to do this for XMPP and messaging applications. This is a general approach and the SLEP Stream service can be embedded in any TCP application, with a SLEP Stream module replacing the TCP module. This would enable the application to run efficiently between two HF nodes, connected by STANAG 5066. A peer node would be addressed using STANAG 5066 Address rather than domain name/IP address. This is a good way to operate in a system where all nodes communicate using HF only.

8.2 TCP PEP

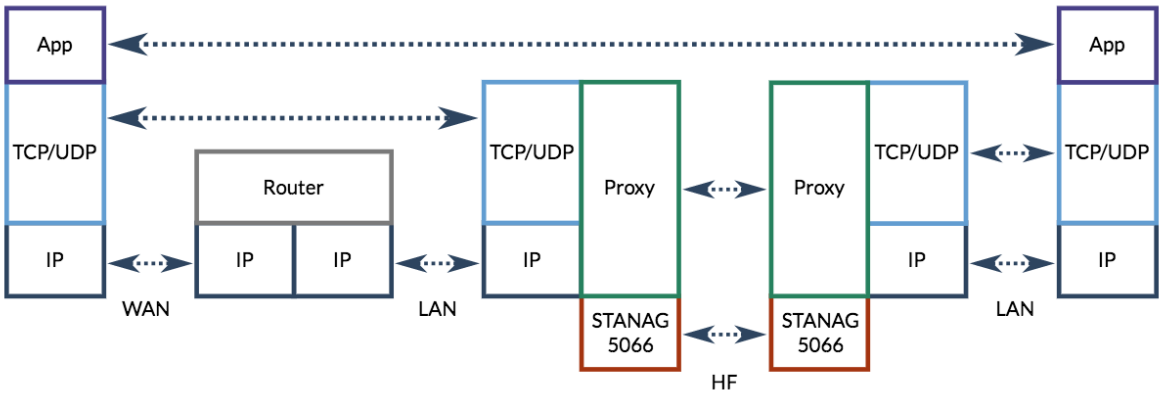

Figure 4: TCP (or UDP) PEP Architecture

The diagram in Figure 4 shows a middle layer proxy architecture, where proxying happens using middle layer protocols such as TCP or UDP, but applications still communicate end to end. This architecture is particularly useful for TCP, and this TCP PEP approach enables a wide range of TCP applications to work efficiently over HF. (There are benefits to this approach for UDP and other protocols running over IP, but this paper focusses on TCP.)

This TCP PEP architecture can remove the protocol overheads of using IP Client, and more importantly can prevent the significant inefficiencies caused by TCP retransmission and window mechanisms operating directly over HF. However, this approach does not make the link faster and does not remove the basic HF latency. As a consequence, application handshakes will still lead to performance problems and short application timers may lead to failures. TCP PEP over HF will always give better performance that TCP over IP Client, and in most situations very much better performance. Not every TCP application will work well over TCP PEP over HF, although most common applications are expected to work well.

TCP PEP needs a protocol for transfer of data between the two proxies. SLEP Stream provides exactly the right functionality to do this efficiently, by providing an equivalent service directly over STANAG 5066 ARQ. There also needs to be some simple protocol to establish new connections correctly. SLEP provides the key capability to make the TCP PEP work.

8.2.1 Transparency & Other Bearers

For basic use, a TCP PEP can be made transparent to the end applications. This can be achieved using standard IP routing. From the perspective of routers and end systems, the system with a proxy is simply another router. The TCP PEP will just get connected when needed.

The SLEP-based HF TCP PEP will only connect pairs of systems over HF. Where there are multiple network links (e.g., HF and satellite), alternate IP routing can be used to select the link. This enables either link to be used. What would not work is if some IP packets used one link and some used the other. The TCP PEP will need to handle all traffic for a period. This means that although routing choice can be transparently handled at IP level, it needs to be done in a way that means traffic is routed one way or the other, with only occasional switches of direction.

9 CONCLUSIONS

This paper has described the STANAG 5066 IP Client approach. While this can work acceptably in some situations, when there is high load or poor HF conditions, performance becomes poor. For some applications, such as messaging and XMPP, an application PEP approach is ideal. This can be integrated with standard IP end applications and optimize traffic over HF.

Use of a TCP PEP provides a flexible approach to integrate a wide range of applications such as Web browsing over HF. A TCP PEP can use the SLEP protocol described in this paper to efficiently transfer data over HF. This can operate in a manner that is transparent to the application and enables switching between bearer services.

REFERENCES

[1] *STANAG 5066 - Edition 3 "Profile for HF Radio Data Communications"*, NATO Standard, December 2010

[2] *UK MOD XMPP OVER HF PILOT*, Presentation to the HF Industry Association, February 2019

[3] *SIS Layer Extension Protocol (SLEP)*, published by Isode as S5066-APP3, https://www.isode.com/whitepapers/S5066-APP3.html (May 2019).

[4] *P_MUL- A PROTOCOL FOR RELIABLE MULTICAST IN BANDWIDTH CONSTRAINED AND DELAYED ACKNOWLEDGEMENT (EMCON) ENVIRONMENTS*, ACP 142(A), Combined Communications-Electronics Board, October 2008.

[5] *Performance Measurements of Applications using IP over HF Radio*, Isode white paper, https://www.isode.com/whitepapers/performance-of-ip-applications-over-hf-radio.html (May 2019).

[6] *STANAG 5066 Large Windows Support*, published by Isode as S5066-EP5, https://www.isode.com/whitepapers/S5066-EP5.html (May 2019).

[7] *XMPP*, XMPP Standards Foundation (XSF), https://xmpp.org/ (May 2019).

[8] *Server to Server communication over STANAG 5066 ARQ*, XEP-0365 published by XSF, July 2018.

[9] *Zero Handshake Server to Server Protocol*, XEP-0361 published by XSF, September 2017.

[10] *Federated MUC for Constrained Environments*, XEP-0289, published by XSF, May 2012.

[11] *Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations*, RFC 3135, published by IETF, June 2001.

[12] *Messaging Protocols for HF Radio*, Isode white paper, https://www.isode.com/whitepapers/messaging-protocols-hf.html (May 2019)

[13] *Providing XMPP Trunking with M-Link Peer Controls*, Isode white paper, https://www.isode.com/whitepapers/xmpp-trunking.html (May 2019).