Escape the tyranny of TCP

The MIT Faculty has made this article openly available. Please share how this access benefits you. Your story matters.

<table>
<thead>
<tr>
<th>Citation</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>As Published</td>
<td><a href="http://dx.doi.org/10.1109/MILCOM.2009.5379959">http://dx.doi.org/10.1109/MILCOM.2009.5379959</a></td>
</tr>
<tr>
<td>Publisher</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>Version</td>
<td>Final published version</td>
</tr>
<tr>
<td>Accessed</td>
<td>Mon Apr 01 10:43:58 EDT 2019</td>
</tr>
<tr>
<td>Citable Link</td>
<td><a href="http://hdl.handle.net/1721.1/58925">http://hdl.handle.net/1721.1/58925</a></td>
</tr>
<tr>
<td>Terms of Use</td>
<td>Article is made available in accordance with the publisher’s policy and may be subject to US copyright law. Please refer to the publisher’s site for terms of use.</td>
</tr>
<tr>
<td>Detailed Terms</td>
<td></td>
</tr>
</tbody>
</table>
ESCAPE THE TYRANNY OF TCP

John Chapin, Member IEEE, and Vincent Chan, Fellow IEEE and OSA
Claude E. Shannon Communication and Network Group
Research Laboratory of Electronics
Massachusetts Institute of Technology

ABSTRACT
The Transmission Control Protocol (TCP) is ubiquitous, sophisticated, and effective. It also prevents the innovation needed to improve delivery of Internet services to the wireless tactical edge of DOD operations. We argue in this paper that TCP should be used as a short-range local access protocol for COTS compatibility rather than as the primary end-to-end transport protocol for the tactical GIG. We describe a straightforward way to implement this architecture without changing COTS endpoints. The implementation includes a TCP spoofing proxy, an open-standard HAIPE-compatible short-haul protocol, and a modularized core transport protocol endpoint.

INTRODUCTION
The Transmission Control Protocol (TCP) solves a challenging set of problems central to the Internet. It provides reliable end-to-end data transport over a lossy network. It enables multiple independent flows to collaborate for high link utilization and congestion control without coordination among users.

While TCP has worked well for wireline networks, its widely deployed variants (here called “COTS TCP”) are not designed to handle the conditions inherent in wireless networks [1][2][3][4]. Unlike wireline networks, wireless links have intrinsic packet loss and rapidly varying capacity and latency, and in some cases have high delay. Intruders can potentially modify packets in transit.

These challenges are particularly prominent in the DOD’s forward-deployed wireless networks, known as the tactical GIG (Global Information Grid). When COTS TCP is used in the tactical GIG, the result is poor end-to-end performance, low utilization of data links, and security weaknesses. The DOD overcomes these problems in part by deploying support components called middleboxes, for example PEP and HAIPE units (Performance Enhancement Proxy and High-Assurance Internet Protocol Encryptor). The current approach is a band-aid solution that will not meet the longer-term needs of the GIG.

TCP spoofing is a mature technology that is widely used in middleboxes. However COTS middleboxes incorporate TCP spoofing into individual point solutions that interact poorly with each other and are not evolvable. We propose in this paper a general architecture and supporting components that will enable affordable evolution of the tactical GIG transport and network layers. Solving the performance problems of the tactical GIG will then depend on development of appropriate transport protocols, which is outside the scope of this paper.

THE TYRANNY OF TCP
The tyranny of TCP refers to the effects of the limited feature set that can be expected from COTS implementations of TCP. While TCP is highly flexible at a technical level, it is tightly constrained as a COTS solution. Commercial market forces will limit the growth of TCP capabilities that handle the specialized requirements of challenged tactical networks.

For example, it is technically feasible for routers to identify which IP packets come from the same TCP connection (“flow tracking”). This supports a range of
useful transport layer optimizations that could be added to TCP. However, the giant scale of the commercial Internet makes flow tracking too expensive to implement in core network routers. Because it will not be implemented in the Internet, optimizations based on flow tracking will never be widely exploited by the TCP stacks of COTS endpoints, despite the potential technical benefits.

If the DOD relies on COTS TCP implementations as its end-to-end transport protocol, the capability and evolution of the tactical GIG will be tightly constrained, its efficiency greatly impaired and its cost prohibitive due to the amount of over-provisioning needed to guarantee quality of service.

**Network design constraints**

To enable COTS TCP to achieve high performance, wireless networks have to be engineered to operate as close to wireline networks as possible, in terms of low link loss, low capacity and latency variation, and continuous end-to-end connectivity. This tightly constrains the design space of possible network architectures and optimizations. Techniques that are hard to deploy include the following.

Distributing packets from a single flow across multiple routes can improve load balancing, energy management, and reliability. TCP will have low performance due to out-of-order packets that trigger its fast retransmission mechanism.

Use of link-layer retransmission can reduce the required amount of interleaving over long-distance links that experience burst errors, leading to substantial improvements in normal-case delays. TCP will have low performance due to out-of-order packets.

Use of a low transmit duty cycle can save energy, reduce the probability of detection and increase spectral efficiency. Moreover, short transmit bursts may be all that is possible in situations where a relay or endpoint is located on a fast aircraft. TCP will offer poor or no connectivity due to the 1.5 round-trip times required for a connection setup handshake.

Allowing occasional short disconnections can significantly simplify the routing problem in MANETs and the capacity allocation problem for networks with multiple traffic priority levels sharing variable capacity links. TCP will have low performance due to retransmission timeouts and connection restarts.

Some network designs result in sharing a single link among users with significantly different round trip times. For example, some SATCOM flows may have only a single satellite hop, while others may traverse two hops, resulting in double the RTT. TCP fails to share the link effectively. The low RTT traffic gets an unfairly high share, irrespective of traffic priority levels, since it increases its transmit rate much more rapidly after a congestion loss.

Some wireless links offer high capacity but also intermittent outages. An attractive system design is to combine an intermittent high capacity link with a more robust but lower capacity backup link between the same pair of nodes. A free-space optical link that is periodically interrupted by atmospheric turbulence can be paired with an RF microwave link. A directional RF antenna that occasionally loses lock during acceleration or turns can be paired with an omni-directional RF antenna. TCP will have low performance over such a network due to its inability to respond appropriately and quickly to the sudden changes in capacity that occur when the primary link activates or drops out.

**Protocol design constraints**

COTS TCP only manages congestion effectively if most other traffic sharing the congested links or subnetworks uses a TCP-compatible rate control algorithm. This is called being “TCP-friendly.” A TCP-friendly rate control algorithm is one that achieves the same long-term throughput as TCP under similar congestion conditions. There are a range of algorithms with this property [5]. However, it is a small portion of the overall design space. Many flows behave differently [6].

One of the worst forms of TCP-unfriendliness is to increase transmission rate when packet loss occurs. However, this is exactly what is required when variable rate FEC (forward error correction) is used to support fixed-bit-rate communication to users with low-quality connections. In the tactical GIG where some of the most challenged links are the connections to war-fighters on the battlefield, such TCP-unfriendly FEC protocols may be highly valuable.

**Heterogeneity constraints**

The GIG is a heterogeneous network, where the connection between sender and receiver traverses multiple network modes (wireless, SATCOM, fiber). TCP is fundamentally an end-to-end transport management mechanism. A single choice of TCP parameters must function acceptably across all network modalities. This constrains the design space of each of them. Examples where end-to-end transport management works poorly include the following.

Faster rate growth in congestion avoidance mode would enable exploiting short high-capacity spikes in a capacity-varying wireless network where there is low congestion, such as a link to a moving vehicle where congestion is managed to first order by admission control. This tuning will lead to congestion problems if the flow
Figure 1. Use of middleboxes in the current tactical GIG. APP = application, PPP = PEP-to-PEP protocol (proprietary).

also traverses a fiber network that is not significantly overprovisioned.

Faster rate growth in slow-start mode would lead to much higher performance over high-delay links, but can cause significant congestion problems in a tactical wireless network connected to the high-delay link. This occurs if one of the links in the tactical network happens to be the most constrained link for a particular flow.

Fast-retransmit could be adjusted to tolerate higher levels of packet reordering in a wireless network before declaring a packet has been lost. However this reduces performance when the same flow encounters congestion loss in a fiber network.

**Security constraints**

TCP offers no security features. Therefore achieving secure communications requires encrypting the entire packet, both headers and data, and sending it through a tunnel.\(^2\) Examples of this approach include commercial VPNs and military HAIPES (Figure 1a).

Adding security in this fashion to an insecure transport protocol eliminates many forms of feedback from the network to the transport layer. For example, the Explicit Congestion Notification mechanism [7] relies on a bit in the IP header that is set when the packet traverses a congested router. With tunnelled TCP, the black IP header where that bit was set is stripped before the packet is delivered to the communicating node. The red IP header, which is all the TCP stack sees, cannot be modified by the routers because it is encrypted.

More generally, most desirable interactions between the transport layer and the routing layer are prevented by full-packet tunneling for security. In wireless networks, many proposals for increasing performance and robustness depend on cross-layer interaction and optimization. The tunnel security approach mandated by the lack of security in TCP is thus a significant barrier to exploiting these advanced networking techniques.

---

\(^2\) Encrypting data connections over TCP (e.g. SSH) provides confidentiality but lacks other required security properties.
TCP+middleboxes transport architecture faces significant challenges in a black core.

The use of end-to-end encryption prevents PEPs, co-located with the SATCOM ground terminal in the network core, from viewing or modifying TCP headers (Figure 2a). PEPs at this location cannot transparently break the end-to-end TCP connection to substitute their own protocol over the satellite link. An alternate approach would be to install PEPs in the red enclave (Figure 2b). This also fails, since the dynamics of the shared network between the PEP and the SATCOM ground terminal interfere with the PEP’s ability to manage satellite link effects. There is also the need to adjust the PEP protocol so it manages congestion properly in the shared network, which reduces the performance it achieves over the satellite link.

Two solutions have been proposed for these problems, both undesirable. The architecture in Figure 2a could work if the PEP at the satellite terminal is given the necessary session keys to decrypt HAIPE tunneled packets. In larger tactical networks with many different flows, this creates a significant key management problem that impairs security and interferes with over-the-network rekeying. The architecture in Figure 2b could work if the PEP in the red enclave has a guaranteed-bandwidth channel through the joint tactical wireless network to the SATCOM ground terminal. This prevents statistical multiplexing of the tactical network’s resources and thus wastes valuable wireless capacity.

While we have used SATCOM as an example in this discussion, the problems are common to the PEPs needed for other advanced networks such as free-space optical systems and power-optimized MANETs. These problems mean that today’s TCP+middleboxes approach needs to be replaced by a new transport layer architecture that enables evolution to a black core.

A new transport layer

One objection to developing a new transport layer is that moving away from TCP as the transport protocol would increase cost, due to reduced ability to use COTS derived applications and to rely on COTS products for tasks such as deep packet inspection, network management, and intruder detection. However, this objection is moot. The widespread use of HAIPE and PEPs means that TCP-based COTS products cannot be used in critical parts of the network anyway.

If the current TCP+middleboxes approach continues to be used as shared tactical wireless networks are deployed in a black core architecture, new forms of PEPs will have to be installed to handle the issues of those networks. The proprietary PEP-PEP protocols will then need to take on all the congestion management features of TCP. They will also need to be supported and managed by other network components just as TCP is in the Internet. The result will be an extremely high cost patchwork.

Managers of the GIG should recognize that the tactical GIG has de facto already moved to a new transport layer, one including a number of different vendor-proprietary PEP protocols. The low cost path forwards is to simplify down to a small set of open standard transport protocols that meet the needs of the tactical GIG, while preserving TCP compatibility for COTS endpoints.

TCP-ACCESS ARCHITECTURE

We propose a transport layer architecture called “TCP-access” in which TCP is used as a short-range access protocol rather than an end-to-end transport protocol. COTS endpoints still use TCP at layer 4 to access the network, but TCP is not responsible for end-to-end reliability, performance, or congestion management. TCP-access enables specialized transport protocols to be used for different modalities (wired, wireless, SATCOM, free-space optical, MANETs, others).

The TCP-access architecture has three key components (Figure 3): a TCP spoofing proxy, an open-standard HAIPE-compatible short-haul protocol, and a modularized core transport protocol endpoint.
two key differences from the TCP+middleboxes architecture.

TCP+middleboxes uses TCP as the end-to-end transport protocol, with individual links using specialized PEP-PEP protocols. TCP-access uses TCP as a red-side access protocol, and a new GIG-oriented core transport protocol as the end-to-end transport.

TCP+middleboxes combines TCP spoofing and specialized transport endpoint functions into a single unit. TCP-access separates these roles. TCP spoofing is performed on the red side where key management is not an issue. Core transport is performed on the black side, where it is possible to perform advanced cross-layer optimizations, collaborate with routers, and collaborate with remote gateways such as a satellite ground terminal.

Proxy
TCP-access uses the same proxy-based spoofing technique as PEPs to transparently break the end-to-end TCP connection. We recommend this be implemented as an open-source software module (e.g. [10]) that can be incorporated into many different platforms. One useful place to put it would be as a network loopback process in COTS endpoints, just as VPNs are implemented today on laptops. This avoids the need for a proxy hardware component when the endpoint is sufficiently capable.

Short-haul protocol
The proxy translates TCP to a UDP-based open standard short-haul transport protocol. The protocol is short-haul in the sense that it is designed to operate over a local uncongested lossless connection. The only nontrivial transport-layer feature of the protocol is bidirectional flow control (pausing the sender if the receiver’s buffer is full).

The short-haul protocol is designed so that it functions correctly with one end on the red side of a HAIPE and the other on the black side. This may require appropriating bits already approved for HAIPE bypass, such as the ECN bits, for protocol-internal purposes such as flow control and flow identification. Appropriating header bits is acceptable because the assumption of a local connection implies there is no router between sender and receiver. However, it would be better to avoid appropriating bits if possible in order to improve deployment flexibility and standards compliance.

HAIPE is not the ultimately desirable security solution for a new transport layer architecture. However, rapid development and incremental deployment both mandate that HAIPE be used as the initial security mechanism. Hence, until a better security solution is developed, we require the short-haul protocol to support HAIPE.

Core Transport Endpoint
On the core network end of the HAIPE is the core transport protocol endpoint. This open source software component includes a stack implementing the black side of the short-haul protocol. It has an internal modular architecture permitting easy replacement of the network-side core transport protocol stack.

The CTPE architecture significantly reduces the barriers to evolution of the core network transport protocol and to specializing it for the different network modalities. For connections that cross modalities, gateways at the border between the networks can terminate and translate between the specialized transport protocols, collaborating with remote core transport endpoints. The cost of implementing protocol translation in gateways is one of the factors to be considered in judging whether to specialize transport protocols to the modalities.

Core Transport Protocol
Although this may be counter-intuitive, COTS TCP is the best protocol to initially deploy in the core transport endpoint for end-to-end transport functions. This is separate from the use of TCP as an access protocol between the application and the red-side proxy. Using TCP across the black core permits immediate deployment of the TCP-access architecture in today’s GIG. For example, PEPs already installed at SATCOM ground terminals will intercept it and substitute their own SATCOM tuned PEP-PEP protocol over the satellite link giving good performance.
(a) **Deployment with no changes to endpoints or routers.**

Compared to current GIG deployments, there are two additional boxes: proxy and CTPE.

(b) **Deployment without additional boxes.**

Capable user endpoints (laptops or servers) can run the proxy function as an internal software process. The CTPE function can run as software on a colocated router or gateway.

(c) **Deployment in a standalone device.**

Even more optimized implementations are possible (e.g. no proxy) for constrained devices with non-COTS applications.

Figure 4. Potential deployments of the TCP-access architecture.

Once the TCP-access architecture is in place using TCP as the core protocol, the core transport protocol can be easily evolved and specialized to meet the evolving needs of the heterogeneous tactical GIG. In particular, it becomes possible to apply advanced networking techniques that today are precluded by the attempt to rely on COTS TCP implementations for end-to-end transport. These techniques include co-design of routers and transport protocol using explicit control signaling between layers 3 and 4, and codesign of gateways and transport protocol with signaling between gateways and layer 4.

EVALUATION OF TCP-ACCESS

The TCP-access architecture does not directly solve the performance problems of TCP over wireless and other advanced networks. Instead, the goal of the TCP-access architecture is to isolate the core of the tactical GIG from the constraints of COTS TCP. Once transport protocol evolution becomes possible, a new family of protocols can be deployed that address the needs of the tactical GIG.

To determine whether TCP-access is useful, we should ask whether it enables affordable protocol evolution at a reasonable development, deployment, and latency cost. In our view, TCP-access provides significant benefits in this area at low cost.

Protocol evolution becomes possible because the core transport protocol endpoint has a modular design permitting easy replacement of the transport protocol stack. Indeed, multiple transport protocols can be preloaded and activated as required by the situation. For example, innovative routers, gateways and other components can be co-designed with advanced transport protocols to solve the problems of specific network modalities and/or technologies. At flow startup time, the appropriate core transport protocol can be selected based on the situation and the available capabilities in that part of the network. In this context, development and deployment of new transport protocols is much more affordable than in the current tactical GIG, because neither end-user systems nor applications need to change. New protocols can be deployed incrementally rather than requiring a big-bang change.

We expect it will be affordable to develop initial implementations of TCP-access. Use of HAIPE as the initial security solution eliminates one major driver of development cost and time. The proxy component is a straightforward evolution of existing solutions, examples of which are available as open-source [10]. The only fully new components are the short-haul protocol and the core transport protocol endpoint, neither of which is complicated.

Incremental deployment costs will depend on the context (Figure 4). At the boundary between a local red enclave and the black core, one or two additional boxes may be needed. One box is required for the core transport protocol endpoint, unless this capability can be integrated into the existing router or gateway. Another box may be required for the proxy, except in the (common) situation where...
that end systems on the red side are capable enough to run the proxy component as an additional software process.

Run-time latency and throughput will depend to a large extent on the design of the proxy and the core transport protocol endpoint, as well as the short-haul protocol. A prototype implementation is needed for analysis of performance impacts.

**CHALLENGES & NEXT STEPS**

While the TCP-access architecture appears straightforward to implement, there are undoubtedly challenges hidden in the usual areas such as buffer management and flow demultiplexing. To identify and solve these problems, as well as measure end-to-end performance impacts, the next step is to develop a prototype implementation.

There are significant security certification risks inherent in the plan to perform flow control, flow identification and related tasks between red and black endpoints of the short-haul protocol. An unpublished systems analysis of this problem suggests that a solution is possible (see Acknowledgement below). HAsurname]E specialists and certification experts will be needed to assist the design and analysis of the protocol.

An important potential form of network optimization is to customize transport behavior to specific application requirements. While nothing in the TCP-access architecture prevents this, an initial implementation using HAIPE for security does not support it. The identity of the application associated with any given data flow is hidden behind the HAIPE unit. This is one of the reasons why TCP-access will benefit from an improved security mechanism that goes beyond HAIPE.

Another security challenge is raised by the direct connection of the core transport protocol to the black core. This software component will be relatively sophisticated and at the same time it is exposed to all packets arriving from the network. Some of the packets can be expected to be malicious. While attacks on the CTP and CTPE cannot compromise the security of red-side systems and data, denial of service and similar effects are possible. A high level of trust in the correctness of the CTP and CTPE software will be needed. This challenge is inescapable in any network design that seeks to achieve cross-layer optimization between the transport layer and lower network layers or black core network components.

If the TCP-access architecture is successfully deployed, it will create the opportunity to deploy new transport layer solutions. Research is urgently needed that looks at how routers, gateways, and the transport protocol can collaborate to solve the problems of the tactical GIG, once the tyranny of TCP is no longer a constraint.

**Acknowledgment**

The authors gratefully acknowledge the contribution of Robert C. Durst of MITRE Corporation. His unpublished presentation “A HAIPE-Friendly PEP Design,” October 2004, provided the initial stimulus for the ideas presented here. The presentation includes detailed systems analysis for a short-haul protocol that works between the red and black sides of a HAIPE box without violating security requirements.

**References**


**John Chapin** is a visiting scientist in the Claude E. Shannon Communication and Network Group at the Research Laboratory of Electronics of MIT. He spent 9 years in technical leadership roles at Vanu, Inc., a provider of SDR based cellular radio access networks. His work there on SDR and cognitive radio earned multiple awards including IEEE DYSPAN best paper, SDR Forum best paper, and the SDR Forum Industry Achievement Award. Prior to Vanu he was on the faculty of the EECS department of MIT. He earned his Ph.D. degree in Computer Science from Stanford University in 1997.

**Vincent W. S. Chan**, the Joan and Irwin Jacobs Professor of EECS, MIT, received his BS(71), MS(71), EE(72), and Ph.D.(74) degrees in EE from MIT. He joined MIT Lincoln Laboratory in 1977 and had been Division Head of the Communications and Information Technology Division until becoming the Director of the MIT Laboratory for Information and Decision Systems (1999-2007). In July 1983, he initiated the Laser Intersatellite Transmission Experiment Program and in 1997, the follow-on GeoLITE Program. He has led multiple academic-industry research consortia including the All-Optical-Network Consortium, the Next Generation Internet Consortium, ONRAMP and the Satellite Networking Research Consortium.