



XMPP

XEP-0286: Mobile Considerations

Dave Cridland

<mailto:dave.cridland@isode.com>

<xmpp:dave.cridland@isode.com>

Sam Whited

<mailto:sam@samwhited.com>

<xmpp:sam@samwhited.com>

<https://blog.samwhited.com/>

2017-01-17

Version 0.4.0

Status	Type	Short Name
Experimental	Informational	NOT_YET_ASSIGNED

This document provides background information for XMPP implementors concerned with mobile devices operating on an LTE cellular network.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2017 by the [XMPP Standards Foundation](#) (XSF).

Permissions

Permission is hereby granted, free of charge, to any person obtaining a copy of this specification (the "Specification"), to make use of the Specification without restriction, including without limitation the rights to implement the Specification in a software program, deploy the Specification in a network service, and copy, modify, merge, publish, translate, distribute, sublicense, or sell copies of the Specification, and to permit persons to whom the Specification is furnished to do so, subject to the condition that the foregoing copyright notice and this permission notice shall be included in all copies or substantial portions of the Specification. Unless separate permission is granted, modified works that are redistributed shall not contain misleading information regarding the authors, title, number, or publisher of the Specification, and shall not claim endorsement of the modified works by the authors, any organization or project to which the authors belong, or the XMPP Standards Foundation.

Warranty

NOTE WELL: This Specification is provided on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.

Liability

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall the XMPP Standards Foundation or any author of this Specification be liable for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising from, out of, or in connection with the Specification or the implementation, deployment, or other use of the Specification (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if the XMPP Standards Foundation or such author has been advised of the possibility of such damages.

Conformance

This XMPP Extension Protocol has been contributed in full conformance with the XSF's Intellectual Property Rights Policy (a copy of which can be found at <https://xmpp.org/about/xsf/ipr-policy>) or obtained by writing to XMPP Standards Foundation, P.O. Box 787, Parker, CO 80134 USA).

Contents

1	Introduction	1
2	Overview	1
3	Compression	1
4	Power Consumption	2
4.1	Transmit no data	2
4.2	Transmit as much data as you can at once	3
5	Notable Extensions	3
6	Acknowledgements	4
7	Security Considerations	4
8	IANA Considerations	4
9	XMPP Registrar Considerations	4

1 Introduction

XMPP as a protocol was designed before the wide spread adoption of mobile devices, and is often cited as not being very mobile friendly as a result. However, this mostly stems from undocumented lore and outdated notions of how XMPP works. As the Internet and protocol design have changed to be more accommodating for mobile, so has XMPP.

This XEP aims to provide useful background knowledge of mobile handset behavior, and those considerations that client and server designers can take to ensure that bandwidth and battery are used efficiently.

2 Overview

The two major constraints on mobile devices are power and bandwidth. With the wide spread proliferation of 3G and LTE technologies, mobile bandwidth and speeds have become broadly comparable to broadband. However, they are still relatively expensive compared to traditional wired networks, and therefore conserving them is still desirable. This XEP mostly focuses on LTE as it already has a very wide deployment and will only continue to further replace 3G technologies.

3 Compression

XML, and by extension XMPP, is known to be highly compressible. Compression of XMPP data can be achieved with the DEFLATE algorithm ([RFC 1951](http://tools.ietf.org/html/rfc1951)¹) via TLS compression ([RFC 3749](http://tools.ietf.org/html/rfc3749)²) or [Stream Compression \(XEP-0138\)](https://xmpp.org/extensions/xep-0138.html)³ (which also supports other compression algorithms). While the security implications of stream compression are beyond the scope of this document (See the aforementioned RFC or XEP for more info), the author does not recommend using TLS compression with XMPP (or in general). If compression must be used, stream level compression should be implemented instead, and the compressed stream should have a full flush performed on stanza boundaries to help prevent a class of chosen plaintext attacks which can cause data leakage in compressed streams. While this may mitigate some of the benefits of compression by raising compression ratios, in a large, real world deployment at HipChat, network traffic was still observed to decrease by a factor of 0.58 when enabling [Stream Compression \(XEP-0138\)](https://xmpp.org/extensions/xep-0138.html)⁴ with ZLIB compression!

While the CPU cost of compression may directly translate to higher power usage, it is vastly outweighed by the benefits of reduced network utilization, especially on modern LTE networks which use a great deal more power per bit than 3G networks as will be seen later

¹RFC 1951: DEFLATE Compressed Data Format Specification version 1.3 <<http://tools.ietf.org/html/rfc1951>>.

²RFC 3749: Transport Layer Security Protocol Compression Methods <<http://tools.ietf.org/html/rfc3749>>.

³XEP-0138: Stream Compression <<https://xmpp.org/extensions/xep-0138.html>>.

⁴XEP-0138: Stream Compression <<https://xmpp.org/extensions/xep-0138.html>>.

in this document. However, CPU usage is also not guaranteed to rise due to compression. In the aforementioned deployment of stream compression, a *decrease* in CPU utilization by a factor of 0.60 was observed due to the fact that there were fewer packets that needed to be handled by the OS (which also takes CPU time), and, potentially more importantly, less data that needed to be TLS-encrypted (which is a much more CPU-expensive operation than compression). Therefore CPU time spent on compression (for ZLIB, at least; other algorithms were not tested) should be considered negligible.

Supporting compression and performing a full flush on stanza boundaries is recommended for mobile devices.

4 Power Consumption

While the wide spread adoption of LTE has dramatically increased available bandwidth on mobile devices, it has also increased power consumption. According to one study, early LTE devices consumed 5–20% more power than their 3G counterparts ⁵. On some networks that support the legacy SVLTE (Simultaneous Voice and LTE) instead of the more modern VoLTE (Voice Over LTE) standard, or even CSFB (Circuit-switched fallback) this number would (presumably) be even higher.

XMPP server and client implementers, bearing this increased power usage in mind, and knowing a bit about how LTE radios work, can optimize their traffic to minimize network usage. For the downlink, LTE user equipment (UE) utilizes Orthogonal Frequency Division Multiplexing (OFDM), which is somewhat inefficient ⁶. On the uplink side a different technology, Single-carrier frequency division multiple access (SC-FDMA) is used, which is slightly more efficient than traditional (non linearly-precoded) OFDM, slightly offsetting the fact that broadcasting requires more power than receiving. LTE UE also implements a Discontinuous reception (DRX) mode in which the hardware can sleep until it is woken by a paging message or is needed to perform some task. LTE radios have two power modes: RRC_CONNECTED and RRC_IDLE. DRX is supported in both of these power modes. By attempting to minimize the time which the LTE UE state machine spends in the RRC_CONNECTED state, and maximize the time it stays in the DRX state (for RRC_CONNECTED and RRC_IDLE), we can increase battery life without degrading the XMPP experience. To do so, the following rules should be observed:

4.1 Transmit no data

Whenever possible, data that is not strictly needed should not be transmitted (by the server or client). Supporting [Client State Indication \(XEP-0352\)](#) ⁷ is recommended. Most importantly, XMPP pings should be kept as far apart as possible and only used when necessary. Server

⁵LTE Smartphone measurements <http://networks.nokia.com/system/files/document/lte_measurement_s_final.pdf>

⁶A Close Examination of Performance and Power Characteristics of 4G LTE Networks <http://www.cs.columbia.edu/~lierranli/coms6998-7Spring2014/papers/rrclte_mobisys2012.pdf>

⁷XEP-0352: Client State Indication <<https://xmpp.org/extensions/xep-0352.html>>.

operators are encouraged to set high ping timeouts, and client implementors are advised to only send pings when absolutely necessary to prevent the server from closing the socket.

4.2 Transmit as much data as you can at once

If one is on 3G, transmitting a small amount of data will cause the radio to enter FACH mode which is significantly cheaper than its high power mode. On LTE radios, however, transmitting small amounts of data is vastly more expensive per bit due to the higher tail-times (the time it takes for the radio to change state). On LTE radios, one should transmit as much data from the client as possible when the radio is already on (eg. by placing messages in a send queue and executing the queue as a batch when the radio is on). Similarly, when data is being received from the server, the mobile devices radio is already in a high power state and therefore any data that needs to be sent to the server should be transmitted.

These rules also apply to server operators: If the server receives data, the phones radio is already on therefore you should send any pending data. Batching data to be sent and sending it all at once will help reduce power consumption.

5 Notable Extensions

This section provides pointers to other documents which may be of interest to those developing mobile clients, or considering implementing optimizations for them in servers.

[Stream Compression \(XEP-0138\)](#)⁸ provides stream level compression.

[Efficient XML Interchange \(EXI\) Format \(XEP-0322\)](#)⁹ allows XMPP streams to use the EXI XML format.

[Entity Capabilities \(XEP-0115\)](#)¹⁰ provides a mechanism for caching, and hence eliding, the disco#info requests needed to negotiate optional features.

[Roster Versioning \(XEP-0237\)](#)¹¹ provides a relatively widely deployed extension for reducing roster fetch sizes.

[Stream Management \(XEP-0198\)](#)¹² allows the client to send and receive smaller keep-alive messages, and resume existing sessions without the full handshake. This is useful on unstable connections.

[Client State Indication \(XEP-0352\)](#)¹³ allows clients to indicate to the server that they are inactive, allowing the server to optimize and reduce unnecessary traffic.

[Push Notifications \(XEP-0357\)](#)¹⁴ implements push notifications (third party message delivery), which are often used on mobile devices and highly optimized to conserve battery. Push

⁸XEP-0138: Stream Compression <<https://xmpp.org/extensions/xep-0138.html>>.

⁹XEP-0322: Efficient XML Interchange (EXI) Format <<https://xmpp.org/extensions/xep-0322.html>>.

¹⁰XEP-0115: Entity Capabilities <<https://xmpp.org/extensions/xep-0115.html>>.

¹¹XEP-0237: Roster Versioning <<https://xmpp.org/extensions/xep-0237.html>>.

¹²XEP-0198: Stream Management <<https://xmpp.org/extensions/xep-0198.html>>.

¹³XEP-0352: Client State Indication <<https://xmpp.org/extensions/xep-0352.html>>.

¹⁴XEP-0357: Push Notifications <<https://xmpp.org/extensions/xep-0357.html>>.

notifications also allow delivery of notifications to mobile clients that are currently offline (eg. in an XEP-0198 "zombie" state).

[Message Archive Management \(XEP-0313\)](#)¹⁵ lets clients fetch messages which they missed (eg. due to poor mobile coverage and a flaky network connection).

6 Acknowledgements

This XEP was originally written by Dave Cridland, and parts of his original work were used in this rewrite. Thanks to Atlassian for allowing me to release hard numbers from their XMPP compression deployment.

7 Security Considerations

This document introduces no new security considerations.

8 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA).

9 XMPP Registrar Considerations

No namespaces or parameters need to be registered with the XMPP Registrar as a result of this document.

¹⁵XEP-0313: Message Archive Management <<https://xmpp.org/extensions/xep-0313.html>>.