

Constrained Application Protocol

[Originalartikel](#)

[Backup](#)

<html> <div id=„siteSub“>From Wikipedia, the free encyclopedia</div>

```
<div id="mw-content-text" lang="en" dir="ltr" class="mw-content-ltr"><div class="mw-parser-output"><p><b>Constrained Application Protocol</b> (<b>CoAP</b>) is an Internet Application Protocol for constrained devices (defined in <a rel="nofollow" class="external text" href="https://tools.ietf.org/html/rfc7228">RFC 7228</a>). It enables those constrained devices to communicate with the wider Internet using similar protocols. CoAP is designed for use between devices on the same constrained network, between devices and general nodes on the Internet, and between devices on different constrained networks both joined by an internet. CoAP is also being used via other mechanisms, such as SMS on mobile communication networks.</p>
```

CoAP is an [application layer](https://en.wikipedia.org/wiki/Application_layer "Application layer") protocol that is intended for use in resource-constrained internet devices, such as [wireless sensor network](https://en.wikipedia.org/wiki/Wireless_sensor_network "Wireless sensor network") nodes. CoAP is designed to easily translate to [HTTP](https://en.wikipedia.org/wiki/HTTP "HTTP") for simplified integration with the web, while also meeting specialized requirements such as [multicast](https://en.wikipedia.org/wiki/Multicast "Multicast") support, very low overhead, and simplicity.^[1] Multicast, low overhead, and simplicity are extremely important for [Internet of Things](https://en.wikipedia.org/wiki/Internet_of_Things "Internet of Things") (IoT) and [Machine-to-Machine](https://en.wikipedia.org/wiki/Machine-to-Machine "Machine-to-Machine") (M2M) devices, which tend to be deeply [embedded](https://en.wikipedia.org/wiki/Embedded_system "Embedded system") and have much less memory and power supply than traditional internet devices have. Therefore, efficiency is very important. CoAP can run on most devices that support [UDP](https://en.wikipedia.org/wiki/User_Datagram_Protocol "User Datagram Protocol") or a UDP analogue.

The Internet Engineering Task Force ([IETF](https://en.wikipedia.org/wiki/IETF "IETF")) Constrained [RESTful](https://en.wikipedia.org/wiki/RESTful "RESTful") Environments Working Group ([CoRE](https://datatracker.ietf.org/doc/charter-ietf-core/)) has done the major standardization work for this protocol. In order to make the protocol suitable to IoT and M2M applications, various new functionalities have been added. The core of the protocol is specified in [RFC 7252](https://tools.ietf.org/html/rfc7252); important extensions are in various stages of the standardization process.

</p> <h2>Features[edit]</h2> <p>The CoRE group has designed CoAP with the following features in mind:</p> Overhead and parsing complexity. <a href=„<https://en.wikipedia.org/wiki/URI>“ class=„mw-redirect“ title=„URI“>URI and content-type support. Support for the discovery of resources provided by known CoAP services. Simple subscription for a resource, and resulting push notifications. Simple caching based on max-age. <p>The mapping of CoAP with <a href=„<https://en.wikipedia.org/wiki/HTTP>“ class=„mw-redirect“ title=„HTTP“>HTTP is also defined, allowing proxies to be built providing access to CoAP resources via HTTP in a uniform way.^{[3]}</p> <p>With the introduction of CoAP, a complete networking stack of open-standard protocols that are suitable for constrained devices and environments becomes available.^{[4]}</p> <h2>Message formats<sup id=„cite_ref-5“ class=„reference“>edit]</h2> <p>CoAP makes use of two message types, requests and responses, using a simple, binary, base header format. The base header may be followed by options in an optimized Type-Length-Value format. CoAP is by default bound to UDP and optionally to <a href=„<https://en.wikipedia.org/wiki/DTLS>“ class=„mw-redirect“ title=„DTLS“>DTLS, providing a high level of communications security.</p> <p>Any bytes after the headers in the packet are considered the message body if any. The length of the message body is implied by the datagram length. When bound to UDP the entire message MUST fit within a single datagram. When used with <a href=„<https://en.wikipedia.org/wiki/6LoWPAN>“ title=„6LoWPAN“>6LoWPAN as defined in <a rel=„nofollow“ class=„external text“ href=„<http://tools.ietf.org/html/rfc4944>“>RFC 4944, messages SHOULD fit into a single IEEE 802.15.4 frame to minimize fragmentation.</p> <h2>Implementations<sup id=„cite_ref-6“ class=„reference“>edit]</h2> <table class=„wikitable sortable“><tr><th>Name</th> <th>Programming Language</th> <th>Implemented CoAP version</th> <th>Client/Server</th> <th>Implemented CoAP features</th> <th>License</th> <th>Link</th> </tr><tr><td>aiocoap</td> <td>Python 3</td> <td><a class=„external mw-magiclink-rfc“ rel=„nofollow“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC 7252</td> <td>Client + Server</td> <td>Blockwise Transfers, Observe (partial)</td> <td>MIT</td> <td><a rel=„nofollow“ class=„external free“ href=„<https://pypi.python.org/pypi/aiocoap>“>https://pypi.python.org/pypi/aiocoap</td> </tr><tr><td>Californium</td> <td>Java</td> <td><a class=„external mw-magiclink-rfc“ rel=„nofollow“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC 7252</td> <td>Client + Server</td> <td>Observe, Blockwise Transfers, DTLS</td> <td>EPL+EDL</td> <td><a rel=„nofollow“ class=„external free“ href=„<https://www.eclipse.org/californium>“>https://www.eclipse.org/californium</td> </tr><tr><td>cantcoap</td> <td>C++/C</td> <td><a class=„external mw-magiclink-rfc“ rel=„nofollow“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC 7252</td> <td>Client + Server</td> <td></td> <td>BSD</td> <td><a rel=„nofollow“ class=„external free“ href=„<https://github.com/staropram/cantcoap>“>https://github.com/staropram/cantcoap</td> </tr><tr><td>Canopus</td> <td>Go</td> <td><a class=„external mw-magiclink-rfc“ rel=„nofollow“

<https://tools.ietf.org/html/rfc7252>>RFC 7252</td> <td>Client + Server</td>
 <td>Core</td> <td>Apache License 2.0</td> <td><a rel=„nofollow“ class=„external free“
<https://github.com/zubairhamed/canopus>><https://github.com/zubairhamed/canopus></td>
 > </tr><tr><td>CoAP implementation for Go</td> <td><a
[https://en.wikipedia.org/wiki/Go_\(programming_language\)](https://en.wikipedia.org/wiki/Go_(programming_language))>Go</td> <td><a class=„external mw-magiclink-rfc“ rel=„nofollow“
<https://tools.ietf.org/html/rfc7252>>RFC 7252</td> <td>Client + Server</td> <td>Core
 + Draft Subscribe</td> <td>MIT</td> <td><a rel=„nofollow“ class=„external free“
<https://github.com/dustin/go-coap>><https://github.com/dustin/go-coap></td>
 </tr><tr><td>CoAP.NET</td> <td>C#</td> <td><a class=„external mw-magiclink-rfc“
 rel=„nofollow“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC 7252, coap-13, coap-08,
 coap-03</td> <td>Client + Server</td> <td>Core, Observe, Blockwise Transfers</td> <td>3-
 clause BSD</td> <td><a rel=„nofollow“ class=„external free“
<https://github.com/smeshlink/CoAP.NET>><https://github.com/smeshlink/CoAP.NET></td>
 </tr><tr><td>CoAPSharp</td> <td>C#, .NET</td> <td><a class=„external mw-magiclink-rfc“
 rel=„nofollow“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC 7252</td> <td>Client +
 Server</td> <td>Core, Observe, Block, RD</td> <td>LGPL</td> <td><a rel=„nofollow“
 class=„external free“ href=„<http://www.coapsharp.com>“><http://www.coapsharp.com></td>
 </tr><tr><td>CoAPthon</td> <td>Python</td> <td><a class=„external mw-magiclink-rfc“
 rel=„nofollow“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC 7252</td> <td>Client + Server +
 Forward Proxy + Reverse Proxy</td> <td>Observe, Multicast server discovery, CoRE Link Format
 parsing, Block-wise</td> <td>MIT</td> <td><a rel=„nofollow“ class=„external free“
<https://github.com/Tanganelli/CoAPthon>><https://github.com/Tanganelli/CoAPthon></td>
 </tr><tr><td>Copper</td> <td>JavaScript (Browser Plugin)</td> <td><a class=„external mw-
 magiclink-rfc“ rel=„nofollow“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC 7252</td>
 <td>Client</td> <td>Observe, Blockwise Transfers</td> <td>3-clause BSD</td> <td><a
 rel=„nofollow“ class=„external free“
<https://github.com/mkovatsc/Copper>><https://github.com/mkovatsc/Copper> <a
 rel=„nofollow“ class=„external free“
<https://addons.mozilla.org/firefox/addon/copper-270430/>>[https://addons.mozilla.org/firefox/ad
 don/copper-270430/](https://addons.mozilla.org/firefox/addon/copper-270430/)</td> </tr><tr><td>eCoAP</td> <td>C</td> <td><a class=„external
 mw-magiclink-rfc“ rel=„nofollow“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC 7252</td>
 <td>Client + Server</td> <td>Core</td> <td>MIT</td> <td><a rel=„nofollow“ class=„external
 free“ href=„<https://gitlab.com/jobol/ecoap>“><https://gitlab.com/jobol/ecoap></td>
 </tr><tr><td>Erbium for Contiki</td> <td>C</td> <td><a class=„external mw-magiclink-rfc“
 rel=„nofollow“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC 7252</td> <td>Client +
 Server</td> <td>Observe, Blockwise Transfers</td> <td>3-clause BSD</td> <td><a
 rel=„nofollow“ class=„external free“ href=„[http://www.contiki-
 os.org/](http://www.contiki-os.org/)>(er-rest-example)</td> </tr><tr><td>iCoAP</td> <td>Objective-C</td> <td><a
 class=„external mw-magiclink-rfc“ rel=„nofollow“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC
 7252</td> <td>Client</td> <td>Core, Observe, Blockwise Transfers</td> <td>MIT</td>
 <td><a rel=„nofollow“ class=„external free“
<https://github.com/stuffrabbitt/iCoAP>><https://github.com/stuffrabbitt/iCoAP></td>
 </tr><tr><td>jCoAP</td> <td>Java</td> <td><a class=„external mw-magiclink-rfc“
 rel=„nofollow“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC 7252</td> <td>Client +
 Server</td> <td>Observe, Blockwise Transfers</td> <td>Apache License 2.0</td> <td><a
 rel=„nofollow“ class=„external free“
<https://code.google.com/p/jcoap/>><https://code.google.com/p/jcoap/></td>
 </tr><tr><td>libcoap</td> <td>C</td> <td><a class=„external mw-magiclink-rfc“ rel=„nofollow“
 href=„<https://tools.ietf.org/html/rfc7252>“>RFC 7252</td> <td>Client + Server</td>
 <td>Observe, Blockwise Transfers, DTLS</td> <td>BSD/GPL</td> <td><a rel=„nofollow“

class=„external free“
href=„<https://github.com/obgm/libcoap>“><https://github.com/obgm/libcoap></td>
</tr><tr><td>LibNyoci</td> <td>C</td> <td><a class=„external mw-magiclink-rfc“
rel=„nofollow“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC 7252</td> <td>Client +
Server</td> <td>Core, Observe, Block, DTLS</td> <td>MIT</td> <td><a rel=„nofollow“
class=„external free“
href=„<https://github.com/darconeous/libnyoci>“><https://github.com/darconeous/libnyoci></td>
</tr><tr><td>lobaro-coap</td> <td>C</td> <td><a class=„external mw-magiclink-rfc“
rel=„nofollow“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC 7252</td> <td>Client +
Server</td> <td>Observe, Blockwise Transfers</td> <td>MIT</td> <td><a rel=„nofollow“
class=„external free“ href=„<http://www.lobaro.com/lobaro-coap>“><http://www.lobaro.com/lobaro-coap></td> </tr><tr><td>microcoap</td> <td>C</td> <td><a class=„external mw-magiclink-rfc“
rel=„nofollow“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC 7252</td> <td>Client + Server</td> <td></td> <td>MIT</td> <td><a rel=„nofollow“ class=„external free“
href=„<https://github.com/1248/microcoap>“><https://github.com/1248/microcoap></td>
</tr><tr><td>nCoap</td> <td>Java</td> <td><a class=„external mw-magiclink-rfc“
rel=„nofollow“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC 7252</td> <td>Client +
Server</td> <td>Observe, Blockwise Transfers, CoRE Link Format, <a rel=„nofollow“
class=„external text“ href=„<https://tools.ietf.org/html/draft-kleine-core-coap-endpoint-id-01>“>Endpoint-ID-Draft</td> <td>BSD</td> <td><a rel=„nofollow“ class=„external free“
href=„<https://github.com/okleine/nCoAP>“><https://github.com/okleine/nCoAP></td>
</tr><tr><td>node-coap</td> <td>Javascript</td> <td><a class=„external mw-magiclink-rfc“
rel=„nofollow“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC 7252</td> <td>Client +
Server</td> <td>Core, Observe, Block</td> <td>MIT</td> <td><a rel=„nofollow“ class=„external
free“ href=„<https://github.com/mcollina/node-coap>“><https://github.com/mcollina/node-coap></td> </tr><tr><td>Ruby coap</td> <td>Ruby</td> <td><a class=„external mw-magiclink-rfc“
rel=„nofollow“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC 7252</td> <td>Client + Server (david)</td> <td>Core, Observe, Block, RD</td> <td>MIT, GPL</td> <td><a
rel=„nofollow“ class=„external free“
href=„<https://github.com/nning/coap>“><https://github.com/nning/coap>
<a rel=„nofollow“
class=„external free“
href=„<https://github.com/nning/david>“><https://github.com/nning/david></td>
</tr><tr><td>Sensinode C Device Library</td> <td>C</td> <td><a class=„external mw-magiclink-rfc“
rel=„nofollow“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC 7252</td> <td>Client + Server</td> <td>Core, Observe, Block, RD</td> <td>Commercial</td> <td><a
rel=„nofollow“ class=„external free“
href=„<https://silver.arm.com/browse/SEN00>“><https://silver.arm.com/browse/SEN00></td>
</tr><tr><td>Sensinode Java Device Library</td> <td>Java SE</td> <td><a class=„external mw-magiclink-rfc“
rel=„nofollow“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC 7252</td> <td>Client + Server</td> <td>Core, Observe, Block, RD</td> <td>Commercial</td> <td><a
rel=„nofollow“ class=„external free“
href=„<https://silver.arm.com/browse/SEN00>“><https://silver.arm.com/browse/SEN00></td>
</tr><tr><td>Sensinode NanoService Platform</td> <td>Java SE</td> <td><a class=„external
mw-magiclink-rfc“ rel=„nofollow“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC 7252</td>
<td>Cloud Server</td> <td>Core, Observe, Block, RD</td> <td>Commercial</td> <td><a
rel=„nofollow“ class=„external free“
href=„<https://silver.arm.com/browse/SEN00>“><https://silver.arm.com/browse/SEN00></td>
</tr><tr><td>SwiftCoAP</td> <td>Swift</td> <td><a class=„external mw-magiclink-rfc“
rel=„nofollow“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC 7252</td> <td>Client +
Server</td> <td>Core, Observe, Blockwise Transfers</td> <td>MIT</td> <td><a rel=„nofollow“

class=„external free“
 href=„<https://github.com/stuffrabbitt/SwiftCoAP>“><https://github.com/stuffrabbitt/SwiftCoAP></td>
 </tr><tr><td>TinyOS CoapBlip</td> <td>nesC/C</td> <td>coap-13</td> <td>Client +
 Server</td> <td>Observe, Blockwise Transfers</td> <td>BSD</td> <td><a rel=„nofollow“
 class=„external free“
 href=„<http://docs.tinyos.net/tinywiki/index.php/CoAP>“><http://docs.tinyos.net/tinywiki/index.php/CoAP>
 </td> </tr><tr><td>txThings</td> <td>Python (Twisted)</td> <td><a class=„external mw-
 magiclink-rfc“ rel=„nofollow“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC 7252</td>
 <td>Client + Server</td> <td>Blockwise Transfers, Observe (partial)</td> <td>MIT</td> <td><a
 rel=„nofollow“ class=„external free“
 href=„<https://github.com/mwasilak/txThings/>“><https://github.com/mwasilak/txThings/></td>
 </tr><tr><td>FreeCoAP</td> <td>C</td> <td><a class=„external mw-magiclink-rfc“
 rel=„nofollow“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC 7252</td> <td>Client + Server +
 HTTP/CoAP Proxy</td> <td>Core, DTLS, Blockwise Transfers</td> <td>BSD</td> <td><a
 rel=„nofollow“ class=„external free“
 href=„<https://github.com/keith-cullen/FreeCoAP>“><https://github.com/keith-cullen/FreeCoAP></td>
 </tr><tr><td>coap-rs</td> <td>Rust</td> <td><a class=„external mw-magiclink-rfc“
 rel=„nofollow“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC 7252</td> <td>Client +
 Server</td> <td></td> <td>MIT</td> <td><a rel=„nofollow“ class=„external free“
 href=„<https://github.com/Covertness/coap-rs>“><https://github.com/Covertness/coap-rs></td>
 </tr></table><h2>Proxy implementations[edit]</h2> Squid 3.1.9
 with transparent HTTP-CoAP mapping module <a rel=„nofollow“ class=„external free“
 href=„<http://telecom.dei.unipd.it/pages/read/90/>“><http://telecom.dei.unipd.it/pages/read/90/>
 jcoap Proxy <a rel=„nofollow“ class=„external free“
 href=„<https://code.google.com/p/jcoap/>“><https://code.google.com/p/jcoap/> Californium
 cf-proxy <a rel=„nofollow“ class=„external free“
 href=„<https://github.com/mkovatsc/Californium>“><https://github.com/mkovatsc/Californium>
 CoAPthon <a rel=„nofollow“ class=„external free“
 href=„<https://github.com/Tanganelli/CoAPthon>“><https://github.com/Tanganelli/CoAPthon>
 FreeCoAP <a rel=„nofollow“ class=„external free“
 href=„<https://github.com/keith-cullen/FreeCoAP>“><https://github.com/keith-cullen/FreeCoAP>
 <h2>CoAP group communication[edit]</h2> <p>In many
 CoAP application domains it is essential to have the ability to address several CoAP resources as a
 group, instead of addressing each resource individually (e.g. to turn on all the CoAP-enabled lights in
 a room with a single CoAP request triggered by toggling the light switch). To address this need, the
 IETF has developed an optional extension for CoAP in the form of an experimental RFC: Group
 Communication for CoAP - <a class=„external mw-magiclink-rfc“ rel=„nofollow“
 href=„<https://tools.ietf.org/html/rfc7390>“>RFC 7390<sup id=„cite_ref-5“ class=„reference“>[5]</sup>
 This extension relies on IP multicast to deliver the CoAP request to all group members. The use of
 multicast has certain benefits such as reducing the number of packets needed to deliver the request
 to the members. However, multicast also has its limitations such as poor reliability and being cache-
 unfriendly. An alternative method for CoAP group communication that uses unicasts instead of
 multicasts relies on having an intermediary where the groups are created. Clients send their group
 requests to the intermediary, which in turn sends individual unicast requests to the group members,
 collects the replies from them, and sends back an aggregated reply to the client.<sup id=„cite_ref-6“
 class=„reference“>[6]</sup></p> <h2>See also[edit]</h2> Internet of Things OMA Lightweight M2M Web of Things<h2>References[edit]</h2> <div class=„reflist“ style=„list-style-type: decimal;“> <div class=„mw-references-wrap“> <ol class=„references“><li id=„cite_note-1“>^ <a rel=„nofollow“ class=„external text“ href=„<https://tools.ietf.org/html/rfc7252>“>RFC 252, Constrained Application Protocol (CoAP) <li id=„cite_note-2“>^ „Integrating Wireless Sensor Networks with the Web“ , Walter, Colitti 2011 <li id=„cite_note-3“>^ <a rel=„nofollow“ class=„external text“ href=„<https://tools.ietf.org/html/rfc8075>“>RFC 8075, Guidelines for Mapping Implementations: HTTP to the Constrained Application Protocol (CoAP) <li id=„cite_note-4“>^ <a rel=„nofollow“ class=„external text“ href=„<http://www.mdpi.com/2224-2708/2/2/235>“>IETF Standardization in the Field of the Internet of Things (IoT): A Survey <li id=„cite_note-5“>^ <a rel=„nofollow“ class=„external text“ href=„<https://tools.ietf.org/html/rfc7390>“>RFC 7390, Group Communication for CoAP <li id=„cite_note-6“>^ „<a href=„<http://www.mdpi.com/1424-8220/14/6/9833/htm>“>Flexible Unicast-Based Group Communication for CoAP-Enabled Devices“ , Ishaq, I.; Hoebeke, J.; Van den Abeele, F.; Rossey, J.; Moerman, I.; Demeester, P. Sensors 2014</div> </div>

<!-- NewPP limit report
Parsed by mw1165
Cached time: 20170709035759
Cache expiry: 2592000
Dynamic content: false
CPU time usage: 0.044 seconds
Real time usage: 0.068 seconds
Preprocessor visited node count: 179/1000000
Preprocessor generated node count: 0/1500000
Post‐expand include size: 6264/2097152 bytes
Template argument size: 0/2097152 bytes
Highest expansion depth: 4/40
Expensive parser function count: 0/500
Lua time usage: 0.006/10.000 seconds
Lua memory usage: 619 KB/50 MB -> <!-- Transclusion expansion time report
(%,ms,calls,template) 100.00% 38.648 1 -total 75.87% 29.324 1
Template:IP_stack 14.28% 5.520 1
Template:Reflist -> </div> <!-- Saved in parser cache with key
enwiki:pcache:idehash:32830538-0:canonical and timestamp 20170709035759 and revision id 787541050 -> <noscript><img src=„<https://en.wikipedia.org/wiki/Special:CentralAutoLogin/start?type=1x1>“ alt=„“ title=„“ width=„1“ height=„1“ style=„border: none; position: absolute;“/></noscript></div> <div class=„printfooter“>

```
Retrieved from "<a dir="ltr"
href="https://en.wikipedia.org/w/index.php?title=Constrained_Application_Pro
tocol&oldid=787541050">https://en.wikipedia.org/w/index.php?title=Constr
ained_Application_Protocol&oldid=787541050</a>"
</div>
```

```
<div class="visualClear"/>
```

```
</html>
```

From:

<https://schnipsl.qgelm.de/> - Qgelm

Permanent link:

<https://schnipsl.qgelm.de/doku.php?id=wallabag:constrained-application-protocol>

Last update: **2021/12/06 15:24**

