

WSN 应用层协议 MQTT-SN 与 CoAP 的剖析与改进^①

陈 旖, 张美平, 许 力

(福建师范大学 福建省网络安全与密码技术重点实验室, 福州 350007)

摘 要: 针对 WSN 网络通讯中急需适合其特点的应用层通讯协议的问题, 剖析了两种主流的 WSN 应用层通讯协议并做出改进. 通过剖析应用层通讯协议 MQTT-SN 与 CoAP 的设计理念与工作机制, 展示了协议性能与适用的场景的差异. 借助 CoAP 的优点进一步提出了 MQTT-SN 的改进方案, 使之适合于点对点通讯与大块数据传输的情况.

关键词: WSN 网络; MQTT-SN 协议; CoAP 协议; 订阅/发布模型

Analysis and Improvement of WSN Application Layer Protocol MQTT-SN and CoAP

CHEN Yi, ZHANG Mei-Ping, XU Li

(Fujian Provincial Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fuzhou 350007, China)

Abstract: For the problem that WSN needs a suitable application layer communication protocol, this paper analyzes two kinds of mainstream WSN application layer communication protocols MQTT-SN and CoAP and gives the corresponding improvement schemes. By analyzing the design concept and working mechanism of these two application layer communication protocol, this paper shows the protocol performance and the differences among applicable scenarios. Using the CoAP advantages, we further improve the MQTT-SN scheme to make it is suitable for the P2P communication and the blockwise data transmission.

Key words: WSN; MQTT-SN; CoAP; subscribe/publish model

1 绪论

无线传感器网络(Wireless Sensor Networks, WSN)近年来已成为工业界、学术界共同的研究热点. WSN 网络可以广泛的应用在工业控制、智能家居、环境监测等诸多方面^[1]. 典型的 WSN 网络包含大量电池供电的无线节点, 这些节点通常使用内存与计算能力受限的低功耗处理器. 目前 WSN 网络已支持多种网络协议栈, 如 Zigbee、TCP/IP 等. WSN 节点间可通过这类协议栈以点对点、广播等方式通讯.

然而, 仅使用这类基础的通讯协议栈进行应用程序开发依然很困难. 因为 WSN 网络有许多有别于传统互联网的特点, 例如通讯实体的状态相较于传统网络是复杂多变的, 易出现节点移动、失效等异常情况.

开发者若仅使用基础的通讯协议栈, 则需要程序自己处理各种 WSN 网络的异常情况, 这对应用开发者提出了很高的要求.

所以, WSN 网络应用开发中还急需一种能封装其网络特点的应用层通讯协议. 由该协议来统一处理通讯中出现的特殊情况, 为上层的应用提供统一的交互逻辑, 以达到优化程序结构和提高网络利用率的目的.

WSN 自身的网络特点对应用层通讯协议的要求可归纳为以下几点:

- (1) 简单性: 节点运算能力受限, 协议应精简高效.
- (2) 流量控制: 节点通讯能力弱, 应减少数据流量.
- (3) 耗能控制: 节点能源受限, 应尽量使节点休眠以延长寿命.

^① 基金项目:福建省教育厅科技项目(JK20111010);福建省自然科学基金(2013J01222);福建省 2013 年战略性新兴产业技术开发项目(闽发改高技[2013]266 号)

收稿时间:2014-05-11;收到修改稿时间:2014-07-04

(4)网络封装: 协议应能封装 WSN 网络异常.

一个优良的 WSN 应用层通讯协议需要满足以上几点需求.

2 WSN网络主流的两种应用层协议CoAP与MQTT-SN

典型的 WSN 网络中存在边界网关和大量电池供电节点. 节点一方面将采集的环境数据通过网关传入传统网络, 另一方面可执行接收到的控制命令. 边界网关负责将 WSN 网络数据与传统网络进行转换.

2.1 MQTT-SN 协议介绍

MQTT-SN 协议^[2]的设计来源于 MQTT 协议^[3]. MQTT 协议常常部属在移动互联网中, 是一种基于主题(topic)的发布/订阅通讯协议^[3]. 所谓发布/订阅模型中通常有三种角色: 发布者(publisher)、订阅者(subscriber)与代理者(broker). 如图 1 所示, MQTT 中的“订阅者”作为数据的消费者, 向“代理者”订阅所需的消息; “发布者”作为数据的生产者, 向“代理者”发布数据; “代理者”接收数据后开始向已订阅消息的“订阅者”推送数据. 协议使用“主题”标示消息. “发布者”和“订阅者”是对节点功能抽象的概念, 节点可兼具两者的功能.

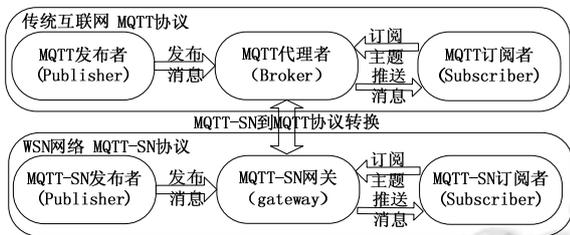


图 1 MQTT-SN 订阅/发布交互模型

MQTT-SN 协议是 MQTT 针对 WSN 网络的拓展. MQTT-SN 的运行需要依赖于已部署 MQTT 协议的网络. MQTT-SN 协议中有两个角色: 节点(node)和网关(GW). 如图 1 所示, WSN 节点通过 MQTT-SN 协议与网关交互, 网关进行协议转换之后使节点接入 MQTT 服务器. 由于 MQTT-SN 协议运行需要依赖于 MQTT 协议, 下文使用 MQTT-SN 协议泛指 MQTT\MQTT-SN 的组合.

MQTT-SN 协议结构如图 2 所示, 其上层是应用程序, 对下层仅要求无连接通讯服务与单跳广播服务. 这种设计模式使其移植性好, 支持多种 WSN 网络协

议栈, 包括 Zigbee 协议栈、IP 协议栈等.



图 2 MQTT-SN 协议结构

MQTT-SN 协议部属结构如图 3 所示. 典型的通讯流程中, 第一步由传统互联网应用程序作为“订阅者”向 MQTT“代理者”服务器订阅所需的数据; 第二步, WSN 节点通过 MQTT-SN 协议作为“发布者”将数据发向 WSN 网关, WSN 网关完成协议转换后发送给“代理者”服务器. “代理者”服务器最终将最新数据推送给已订阅的应用程序. 由于“代理者”服务器位于网络中心, 下文简称为中心服务器.

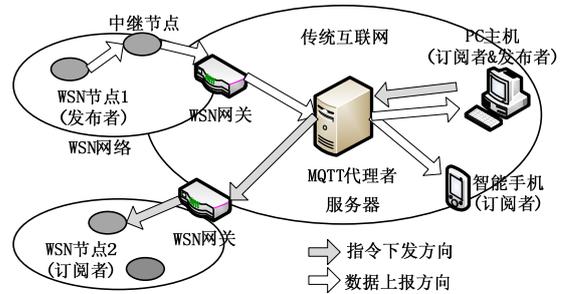


图 3 MQTT-SN 协议部署结构

2.2 CoAP 协议介绍

CoAP 协议在设计上参考了 HTTP 协议, CoAP 在节点间提供一种请求/响应的交互模型^[4]. 其秉承了 WEB 中重要的设计概念, 例如: URIs、因特网媒体类型等, 并使用了与 HTTP 类似的资源操作方法^[4]. 请求/响应交互模型可表示为:

- (1)客户端→服务器: 服务请求
- (2)客户端←服务器: 响应

模型中有两个角色: 客户端(client)和服务器(server). 当客户端需访问服务器资源时, 由客户端发起请求(request), 服务器接收请求后做出响应(response). 模型中的客户端和服务器是功能逻辑上的划分, 节点可兼具客户端和服务器的功能. CoAP 除基础的请求/响应交互模型外, 还支持组播模式^[5]、观察者模式^[6]、块数据传输^[7]的交互方式.

CoAP 协议结构如图 4 所示, 主要分为两层: 请求/响应层(request/response); 消息层(Message). 消息层下层面面向 UDP 层, 它负责在终端间交换数据. 消息层

可提供可靠传输服务;请求/响应层的数据向上面向应用程序,负责实现请求/响应的交付模型。



图 4 CoAP 协议结构

CoAP 协议部署结构如图 5 所示。CoAP 网络典型的通讯过程中,由传统互联网中的应用程序做为“客户端”发出 HTTP“请求”,HTTP 数据包由跨协议转换节点处理为 CoAP 数据包并发送到 CoAP“服务器”。最后节点通过网关回复“响应”给访问者。

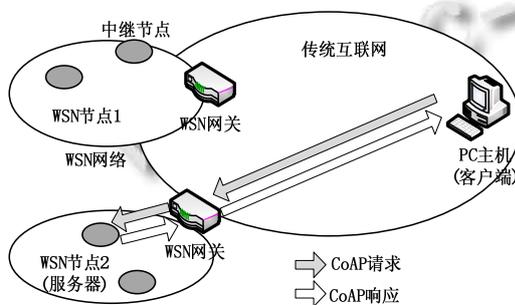


图 5 CoAP 协议部署结构

3 MQTT-SN与CoAP的主要功能与机制比较

3.1 协议设计思路分析与资源的集中方式的思考

从整体审视 MQTT-SN 与 CoAP 协议可发现两者以不同的设计思路来处理 WSN 网络。CoAP 协议将 WSN 网络视为互联网的一部分,是一种资源受限的互联网,所以协议中大量使用类似于传统互联网的设计原则。MQTT-SN 协议将 WSN 网络抽象成为传统互联网中的“资源”,这种“资源”浓缩在传统网络的服务器中,所有对 WSN 网络的操作都可通过操作该“资源”服务器实现。另外,协议的设计思路对资源的集中方式产生的影响。

CoAP 协议中,WSN 节点自身能够提供传感器数据与服务,其相当于传统网络中的服务器。客户端通过协议访问该节点以获取数据。CoAP 协议的这种交互方式源于 WEB 中著名的 REST(Representational State Transfer)模型^[8]。该模型下,网络的基本单位是节点,WSN 网络的资源分布在每个节点之中。操作资源的过程可简单刻画为:节点向另一节点的资源发出请求;

后者做出响应。

MQTT-SN 协议将 WSN 网络整体抽象成为互联网的“资源”。协议先将每个 WSN 节点的资源汇总为整个 WSN 网络的资源总体,再对资源总体以内容进行划分,最后协议可清晰的描绘出 WSN 网络可提供的各项资源集合。该模式下,网络的基本单位是资源类型,相似的节点资源合并为一类面向访问者的资源。协议工作过程可简单刻画为:所有节点将资源信息汇总到中心服务器;客户端通过访问服务器间接操作 WSN 网络。

CoAP 与 MQTT-SN 由于不同的设计思想,在资源的集中方式上形成了“集中式”与“分布式”两种方式;网络的基本单位划分成了“资源类型”和“节点实体”。

3.2 资源集中方式对协议机制设计的影响

3.2.1 流量控制

由第 1 节的需求分析可知应用层协议应尽量减少网络流量。CoAP 和 MQTT-SN 协议的诸多机制都为优化流量而设计,本节从流量优化的角度进行比较分析。

(1)消息重用:CoAP 协议使用缓存机制(cache)和代理服务器来实现消息重用^[4]。缓存机制会缓存部分“请求”消息的“响应”,当节点收到相似“请求”时,就直接使用缓存中的“响应”回复。该机制使用“新鲜(Freshness)模型”^[4]和“有效(Validation)模型”^[4]保证缓存的正确性。缓存机制常常和代理服务器一起使用,客户端向支持缓存机制的代理服务器发送“请求”,如果代理发现缓存中有合适的“响应”就直接回复客户端。MQTT-SN 中,服务器负责向所有已“订阅”某主题的客户推送。节点仅需上报一条新消息,由服务器完成重复消息的下发。

(2)消息群发:CoAP 协议使用组播的方式进行消息群发:组播涉及到节点间一对多的传输关系,是通过 IP 多播的方式实现的^[5]。MQTT-SN 的客户端都向服务器订阅数据,由服务器集中向所有客户端发送新数据,通过这种方式完成了群发。

(3)消息推送:客户端要求实时获取资源的变化情况时,使用 REST 模型就比较复杂。传统的 HTTP 协议采用轮询服务器或者 HTTP long polling^[9],但效率都较低。CoAP 协议为此提供观察者交互模式^[5],其工作流程可表述为:

第一步:观察者→服务器:注册要监视的资源

第二步:观察者←服务器:资源变化时回复通知

客户端作为“观察者”(observers)向服务器注册要监视资源;当服务器发现资源变化时,主动通知客户端. MQTT-SN 协议服务器主动推送新数据,能保证新数据的实时推送.

在消息推送中,两种协议中的服务器是不同的,CoAP 的服务器是普通的节点,它只负责推送自身的新消息;MQTT-SN 为中心服务器,它负责全网的数据推送.

3.2.2 资源定位与服务发现

CoAP 协议中网络的基本单位是实际的节点实体,而 MQTT-SN 网络的基本单位是抽象的资源类型. 这种差别影响了资源定位方式和节点的标示方法.

(1)资源定位与节点标示

CoAP 使用 WEB 的通用资源表示符号(URI 即 Uniform Resource Identifier)来对网络资源定位. 其格式为“coap-URI = "coap:" "://" host [":" port] path-abempty ["?" query]”. 其首先使用“coap”或“coaps”标示协议类型. 第二部分使用 host: port 的方式指明访问的主机地址与端口,端口缺省为 5683. 第三部分为资源在服务器上具体的路径地址.

MQTT-SN 使用主题与主题通配符来定位资源. 主题用来标示一类资源,并使用“/”分层. 典型的主题如“fjnu/mcs/lab507/light-sensor”,意为福建师范大学(fjnu)的数学计算机科学学院(mcs)的“507 实验室”的光敏传感器(light-sensor). 通配符可用以通配多条资源,例如使用“fjnu/mcs//lab507/#”来匹配“507 实验室”层下的任意资源.

CoAP 协议以节点实体作为基本单位,所使用的 URI 需要使用主机名的方式来标示实体,进而描述其资源. MQTT-SN 协议以资源类型作为基本单位,使用主题标示资源的集合. 主题下的资源集合来源于多个节点. 协议使用客户端 ID 标示单个节点.

(2)服务发现

CoAP 使用多播来发现服务器^[4],并通过发送 URI: “/.well-known/core”向服务器获取可用资源列表. MQTT-SN 客户端使用通配符从中心服务器获取用主题标示的资源列表. CoAP 与 MQTT-SN 进行服务发现的区别与其资源集中方式有关: CoAP 获取资源列表的方式是对每一个具体的节点发起请求,而 MQTT-SN 中是对网络的中心服务器发起请求.

3.2.3 连接互联网与安全协议

(1)连接互联网

CoAP 通过跨协议代理(Cross-Protocol Proxy)的方式连接传统互联网^[4]. 由于 CoAP 参考了 HTTP 协议并实现了 HTTP 的子集^[4],在 CoAP 与 HTTP 之间建立跨协议代理是简单而高效的. 火狐浏览器现已集成该功能. MQTT-SN 协议需与传统网络中的 MQTT 协议配合工作, MQTT-SN 节点通过 MQTT-SN 网关转换后接入 MQTT 协议工作. 由于 MQTT-SN 和 MQTT 原本就是相似的设计,网关的转换过程简单而高效.

(2)网络安全

CoAP 使用 DTLS 协议^[10]来保证通讯安全^[4]. DTLS 相当于是在 UDP 的基础上实现了 TLS 协议的功能. 由于 WSN 节点资源受限,只有部分功能较强的节点可支持 DTLS 协议. DTLS 协议通过公钥密码系统和证书来提供包括用户和服务器的合法性认证、加密数据以及隐藏被传送的数据、维护数据的完整性的服务.

在现有的 MQTT 版本中(如 Apache 实现的 MQTT 服务器 ActivityMQ)已经开始使用 SSL 协议来提供网络安全性保证. 但 MQTT-SN 协议还没有设计安全机制.

3.3 其他一些重要机制

(1)MQTT-SN 的睡眠模式

由于 MQTT-SN 中的集中化处理可以将节点的大部分工作交给后端服务器处理,使得 MQTT-SN 协议可提供节点休眠机制. 而 CoAP 协议中节点需处理突发的请求,协议没有定义睡眠机制.

(2)WILL 消息

MQTT-SN 协议中提供“WILL 消息”以通知节点的异常情况. 在节点注册时,可设定 PING 周期和 WILL 消息. 节点需在 PING 周期内向服务器发送 PING 消息,以通报其处于活动状态. 若服务器超时未收到其 PING 消息,则通报所有订阅了 WILL 消息的客户端,该节点处于异常的离线状态.

(3)数据块传输

CoAP 支持块数据(block)传输^[7]. 由于 CoAP 使用 UDP 通讯,过大的数据(如图片数据)将导致底层协议对数据包分片. 一片分片丢失将引起整个数据包传输失败. 块数据传输模式在应用层对数据进行分割和重组以避免底层分片,并使用 ACK 机制保证数据可靠到达,从而提升块数据传输的效率. MQTT-SN 协议中不支持数据分片,协议中规定了数据的最大上限.

4 协议评价

本文选取物联网中两个基础属性指标: 水平性和可扩展性^[11], 以及第 1 节中对 WSN 应用层协议的四个要求: 简单性, 流量控制, 耗能控制, 封装性这六个角度评价 CoAP 与 MQTT-SN 协议。

(1)水平性: 指系统可以集成不同的通信、传输和信息处理技术, 应用于不同的领域^[11]. CoAP 协议作为传统互联网向 WSN 网络的拓展, 其要求底层提供 IP 网络协议栈的支持. 而 MQTT-SN 对底层协议要求宽松, 使之可兼容多种协议包括 Zigbee、IP 等.

(2)可扩展性: 指系统的性能不会随着设备数量的增加而快速的降低^[11]. CoAP 协议中, 节点承担主要工作, 节点增加对网络整体性能影响不大. MQTT-SN 由中心服务器承担主要工作, 节点增加会加重中心服务器的负担.

(3)简单性: MQTT-SN 将节点的大部分重复工作交由中心服务器处理, MQTT-SN 中节点的工作量少. 而 CoAP 节点独立处理对自身资源的请求, 工作量较大.

(4)流量控制: MQTT-SN 协议中消息重用、群发、推送等流量控制机制都借助传统网络中的服务器实现; CoAP 协议中使用代理与缓存、组播、观察者模式等方式进行流量控制.

(5)耗能控制: MQTT-SN 节点工作量小且协议中规定睡眠机制, 有益于降低耗能. CoAP 节点自身工作量较大, 如果节点休眠会导致访问资源失败.

(6)封装性: CoAP 协议着重点在 WSN 网络中的个体, 而 MQTT-SN 协议着重点在对 WSN 网络整体进行抽象, 其拥有较好的封装性.

通过性能的比较, 可分析两种协议的适用场景. MQTT-SN 协议的网络重心集中于传统网络服务器, 使之适用于数据采集型的网络, 其数据流量主要从 WSN 网络流向传统互联网. CoAP 的运行并不依赖于传统网络或网络中心节点, 故其比较适合于 M2M 的控制场景.

5 MQTT-SN的优化设计改进方案

5.1 基于 MQTT-SN 协议的点对点传输机制设计

MQTT-SN 协议由于其集中式处理的特点, 在点对点通讯方面存在缺陷.

如图 6 所示, 若 A 与 B 进行点对点传输, 数据需经网关发送至 MQTT 服务器, 再回传至 WSN 网络内的 B. 假设底层路由算法能找到点对点传输的最优跳

数, A 到 GW 路径为 n 跳; B 到 GW 路径为 m 跳; A 到 B 的跳数为 k 跳; 需要点对点通讯需传输数据包为 M , 共有 N 条. 则以上传输过程与消耗的 WSN 网内传输跳数如表 1 所示.

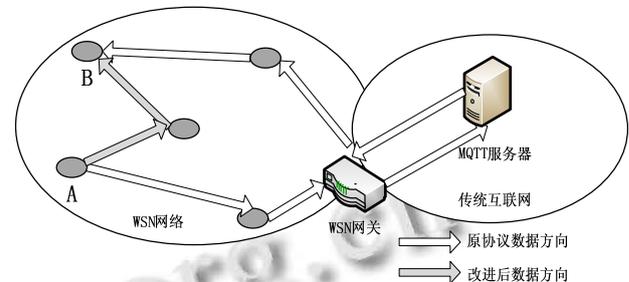


图 6 MQTT-SN 协议点对点通讯路径

表 1 原协议中点对点传输过程

传输过程	传输跳数消耗
1、A→GW: M	n
2、GW→服务器	非 WSN 内流量
3、GW←服务器	非 WSN 内流量
4、GW→B: M	m
5、重复 1-4 步骤 N 次	$N*(n+m)$

本节利用 CoAP 的优点为 MQTT-SN 设计了一种新机制以改进其点对点传输性能. CoAP 协议中, 节点 A 与 B 进行点对点传输, A 可直接向 B 的地址发送数据. 而 MQTT-SN 以节点 ID 标示节点, 源节点若要以类似 CoAP 的方式进行点对点通讯, 需先知道目的节点 ID 对应的地址. 而 GW 恰恰存有所有节点 ID 与地址的转换表.

所以新机制中, 协议需增加一条节点向 GW 寻问地址的命令. 发起点对点通讯时, 源节点首先使用该命令向 GW 发送目的节点的 ID 号; 之后, GW 将回复目的节点的地址; 最后, 源节点向 B 进行点对点传输. 其数据包传输路径如图 6 所示. 其传输过程与消耗的网内传输跳数如表 2 所示.

表 2 改进后点对点传输过程

传输过程	传输跳数消耗
1、A→GW: B 节点 ID	n
2、A←GW: B 地址	n
3、A→B: M	k
4、重复 1-3 步骤 N 次	$2n+N*k$

由于假设底层路由算法能找到最佳路径, 则绝大

多数情况下 $k < n+m$, 相应的, 随着 N 的增长, $2n+N*k$ 将小于 $N*(n+m)$. 例如图 6 中 k 为 2; n 与 m 都为 2. 当 $N > 2$ 时, $2n+N*k < N*(n+m)$. 较少的传输跳数意味着更低的能量消耗与更少的传输时间. 故新的点对点传输机制性能更好.

5.2 基于 MQTT-SN 协议的块数据传输机制设计

CoAP 协议提供了块数据传输机制, 其核心思想是在应用层对数据进行分割和重组, 以避免由于部分分片丢失引起整个包发送失败. 而 MQTT-SN 协议不支持超过其负载上限的数据传输.

本节借鉴 CoAP 的设计原理, 为 MQTT-SN 设计一种块数据传输机制. 该机制中需新增加一条块传输协商命令和块数据发布命令. 该机制流程可描述为:

- (1) A → GW: 发送块传输协商请求
- (2) A ← GW: 返回传输参数
- (3) A → GW: 发送分片数据
- (4) A ← GW: 接收确认消息
- (5) 重复 3-4 步发送所有分片, GW 重组消息

该机制通避免了底层网络协议对大数据包的切割, 并借由确认机制保证了大块数据的可靠到达, 填补了 MQTT-SN 在块数据传输方面的空白.

6 总结与展望

WSN 网络由于其资源受限、应用环境特殊等特点, 使得 WSN 网络中的应用开发急需适合其网络特点的应用层通讯框架. 本文首先介绍了两种针对 WSN 网络设计的应用层通讯协议 CoAP 和 MQTT-SN, 并从协议的设计思路与资源的集中处理方式剖析两个协议的设计细节. 接下来文章从流量控制、资源定位等方面剖析了不同的资源集中方式对协议设计产生的影响, 并通过多项指标比较这两种协议的优缺点. 最后对 MQTT-SN 协议提出了优化方案.

由于 WSN 网络中可能存在可控制电气设备的节点, 如果网络瘫痪或者被攻击者控制将导致重大的损失. 如何在 WSN 应用层协议中提供安全保障也是一个值得研究的问题.

参考文献

- 1 Jean-Philippe V, Dunkels A. Interconnecting Smart Objects with IP: The Next Internet. Elsevier, 2010: 342-343.
- 2 Stanford-Clark A, Truong HL. MQTT For sensor networks (MQTT-SN) protocol specification version 1.2 (work in progress), mqtt protocol specifications. http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf.
- 3 IBM Developer Works Technical Library. MQ telemetry transport (MQTT) V3. 1 protocol specification. <http://www.ibm.com/developerworks/webservices/library/ws-mqtt/2010>.
- 4 Shelby Z, Sensinode, Hartke K, Bormann C. Constrained application protocol (CoAP), draft-ietf-core-coap-13(work in progress). The Internet Engineering Task Force-IETF, Dec 2013.
- 5 Rahman A, Dijk EO. Group communication for CoAP, draft-ietf-core-groupcomm-06(work in progress). The Internet Engineering Task Force-IETF, April 2013.
- 6 Hartke K. Observing resources in CoAP.draft-ietf-core-observe-08 (work in progress). The Internet Engineering Task Force-IETF, February 2013.
- 7 Bormann C, Shelby Z. Blockwise transfers in CoAP, draft-ietf-core-block-10(work in progress). The Internet Engineering Task Force-IETF, October 2012.
- 8 Fielding RT. Architectural styles and the design of network-based software architectures[Ph.D. Thesis]. University of California, 2000.
- 9 Loreto S, Saint-Andre P, Salsano S, Wilkins G. Known issues and best practices for the use of long polling and streaming in bidirectional HTTP, RFC 6202. The Internet Engineering Task Force-IETF, April 2011.
- 10 Rescorla E, Modadugu N. Datagram transport layer security version 1.2, RFC 6347. The Internet Engineering Task Force-IETF, January 2012.
- 11 陈海明, 崔莉, 谢开斌. 物联网体系结构与实现方法的比较研究. 计算机学报, 2013, 36(1): 168-188.