



## 基于CoAP协议的泛在电力物联网拥塞控制算法

任智,吴本源,周舟,苏新

(重庆邮电大学通信与信息工程学院,重庆400065)

**摘要:** 泛在电力物联网具有接入节点数多、设备资源受限的特点,易发生网络拥塞而导致电力信息流丢失、时延过大等问题。基于受限应用协议(CoAP)的泛在电力物联网应用层通信架构,提出一种链路稳定性的CoAP拥塞控制(L-CoCC)算法。通过强、弱和失败消息往返时间来确定网络环境状态并平滑估计超时重传时间(RTO),根据超时重传次数和消息往返时间抖动值,引入下界RTO限制方法和更新老化概念,避免出现不必要的重传。仿真结果表明,相比CoCoA++和CoAP算法,L-CoCC算法能够提升吞吐量和请求成功率,降低平均端到端时延,有效缓解网络拥塞。

**关键词:** 泛在电力物联网;受限应用协议;拥塞控制;可靠传输;应用层通信协议

开放科学(资源服务)标志码(OSID):



中文引用格式:任智,吴本源,周舟,等.基于CoAP协议的泛在电力物联网拥塞控制算法[J].计算机工程,2021,47(10):166-173.

英文引用格式:REN Z, WU B Y, ZHOU Z, et al. Congestion control algorithm for ubiquitous electric IoT based on CoAP protocol[J]. Computer Engineering, 2021, 47(10): 166-173.

## Congestion Control Algorithm for Ubiquitous Electric IoT Based on CoAP Protocol

REN Zhi, WU Benyuan, ZHOU Zhou, SU Xin

(College of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

**[Abstract]** Ubiquitous electric IoT is characterized by massive access nodes and limited device resources. When providing reliable transmission service, ubiquitous electric IoT tend to suffer from congestion, resulting in loss of power information flow and excessive delay. To solve this problem, a link-stability-based congestion control algorithm, L-CoCC, is proposed based on the Constrained Application Protocol (CoAP) for the application layer in ubiquitous electric IoT. By using the Round-Trip Time (RTT) of strong, weak and failed messages, the algorithm determines the state of the network environment and smoothly estimates the Retransmission Timeout (RTO). Based on the number of retransmission times and the fluctuation value of message RTT, a method for limiting the lower bound RTO is introduced, and the aging concept is updated to avoid unnecessary retransmissions. The experimental results show that compared with the CoCoA++ algorithm and the CoAP algorithm, the proposed algorithm improves the throughput and the success rate, and reduces the average delay. It can effectively alleviate network congestion.

**[Key words]** ubiquitous electric IoT; Constrained Application Protocol (CoAP); congestion control; reliable transport; communication protocol in application layer

DOI: 10.19678/j.issn.1000-3428.0058910

### 0 概述

2019年,国家电网有限公司提出建设“三型两网”世界一流能源互联网的战略目标<sup>[1]</sup>,其核心是建设泛在电力物联网以实现电力业务全面感知、高效处理、便捷应用<sup>[2]</sup>,综合应用大数据、云计算、人工智能等新兴通信技术,使电力系统设备和通信技术相融合。目前,

电力系统业务发展具有信息采集业务爆发式增长、控制业务末端发展、覆盖范围广等特点<sup>[3-4]</sup>。由于泛在电力物联网节点数目众多且范围广,随着越来越多的资源受限影响节点间应用层数据传输效率,进而无法实现电力信息流的泛在互联和有效传输。

为解决物联网设备应用层通信的效率问题,国际互联网工程任务组(The Internet Engineering Task Force,

基金项目:国家自然科学基金(61379159);长江学者和创新团队发展计划(IR T1299)。

作者简介:任智(1971—),男,教授、博士,主研方向为移动自组织网络;吴本源、周舟、苏新,硕士研究生。

收稿日期:2020-07-13 修回日期:2020-10-14 E-mail:awkese\_96@163.com

IETF)于2014年提出RFC7252标准协议规范,设计一种基于表征状态转移(Representational State Transfer, REST)架构的合理通用应用层通信协议,即受限应用协议(Constrained Application Protocol, CoAP)<sup>[5]</sup>。该协议类似于HTTP协议,常用于低功耗、资源受限网络节点间的通信,主要目标是满足(如物联网)等节点资源受限环境下的特殊需求,实现物联网数据共享<sup>[6]</sup>。

合理的拥塞控制是通信可靠传输的必要保证。CoAP协议在RFC7252标准中提供一种基于Stop-and-wait ARQ重传机制<sup>[6]</sup>的拥塞控制机制,在默认情况下,重传时间初始值随机从2 s、3 s内选取,此后每次重传时用指数退避算法对重传超时时间(Retransmission TimeOut, RTO)进行指数级更新。但物联网的网络环境往往彼此差别很大,因此这种原始的拥塞控制在物联网环境中无法保证具有良好的数据传输效率。IETF提出一种新的RFC拥塞控制草案(CoAP Simple Congestion Control/Advanced, CoCoA)<sup>[6]</sup>,对CoAP原始拥塞控制机制进行改进。CoAP协议获取能够反映当前链路状况的链路往返时间(Round-Trip Time, RTT)并根据RTT自适应地调整RTO,为CoAP协议的拥塞控制机制增加了动态和学习特性。文献[9]提出CoAP物联网应用层通信方案,介绍物联网场景的数据传输,探讨物联网中应用层流服务的必要性。文献[10]提出动态模式选择机制,在CoAP低功耗网络场景下应用层通信有2种操作模式,以降低功耗并最小化延迟。文献[11]对CoAP原始拥塞控制算法与CoCoA算法进行性能和可靠性对比,原始CoAP拥塞控制算法过于保守导致在网络拥塞状态产生高时延和低成功率。文献[12]在CoAP头部增加重传ID字段来解决弱估计器不确定性和在低带宽、高丢包率的链路下存在估计误差的问题,从而显著提高拥塞网络中的吞吐量。文献[13]提出CoCoA协议在网络流量激增和超时重传选择不当时,弱估计器无法准确对RTT进行计算,进而导致RTO出现估计误差,拥塞控制性能明显低于原始CoAP拥塞控制机制。文献[14]采用CoCoA+算法对弱估计器计算进行修改,减少了弱变化对网络传输的影响,并增加了可变后退因子。文献[15]利用CoAP传输带宽延迟进行网络拥塞控制,对发送端的传输速率进行调整,并与估计带宽相匹配实现拥塞控制。

针对泛在电力物联网资源受限特性,本文基于CoAP协议的泛在电力物联网应用层通信协议,提出一种链路稳定性的CoAP拥塞控制(A Link-Stability-Based CoAP Congestion Control, L-CoCC)算法。将泛在电力物联网架构结合CoAP协议进行应用层通信,完成电力业务与通信技术相融合。在此基础上,采用3种自适应RTT估计器来区分链路状态进而估计RTO。

## 1 泛在电力物联网与CoAP协议

泛在电力物联网的目标是将电力系统和新兴通信产业跨平台相结合,因此,泛在电力物联网场景下的应用层通信协议尤为重要。

### 1.1 CoAP协议

CoAP协议的设计是为了满足资源受限网络和物联网的应用层通信需求,其协议栈如图1所示。为满足CoAP协议的轻量级要求,以UDP为传输层协议,数据通过请求/响应模型进行交互。

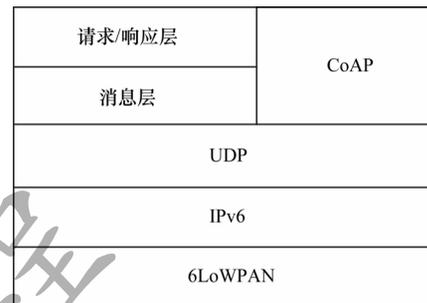


图1 CoAP协议栈

Fig.1 Protocol stack of CoAP

CoAP协议逻辑上采用双层结构:1)消息层,节点间数据进行交互和拥塞控制;2)请求/响应层,与HTTP类似的资源操作,包括获取资源的GET方法、创建资源的POST方法、更新资源的PUT方法、删除资源的DELETE方法。

CoAP协议共定义了4种消息类型,分别为需要被确认的消息(Confirmable Message, CON)、不需要被确认的消息(Non-Confirmable Message, NON)、应答消息(Acknowledge Message, ACK)、复位消息(Reset Message, RST)。通过这4种消息,CoAP协议同时实现了物联网场景下的不可靠传输和轻量级的可靠传输。

### 1.2 泛在电力物联网体系架构

泛在电力物联网将电网中的基础设施、员工、用户以及其所在环境进行互联互通的网络环境,是一种适用于电力业务系统的工业级物联网。其体系架构如图2所示,在物联网感知延伸层、网络传输层和平台应用层3层架构的基础上增加了边缘计算层。

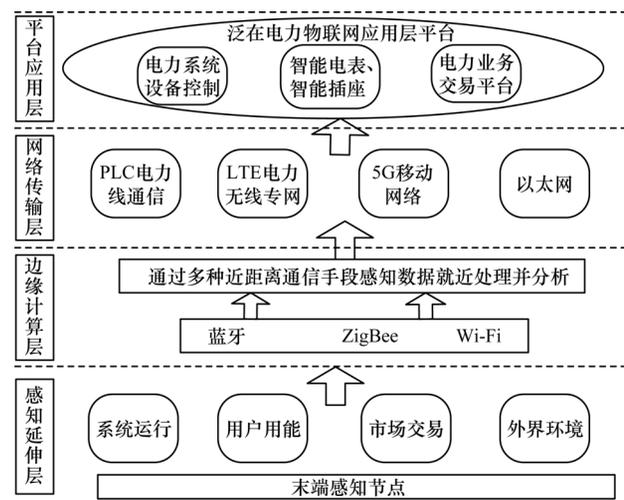


图2 泛在电力物联网架构

Fig.2 Architecture of ubiquitous electric IoT

泛在电力物联网正处在飞速发展阶段,通信网络核心正在搭建,但加入急速增长的终端业务节点后仍面临着通信覆盖力度和可靠性的问题。目前迫切需求一种能够满足电网业务精准负荷控制和智能化管理的通信协议[16],泛在电力物联网的网络特点对其通信协议的要求归纳为以下3个方面:1)轻量级,节点数目多且运算能力受限,协议应当尽量简单高效;2)流量控制,节点资源受限通信能力弱,应减少业务负荷量;3)兼容和可拓展性,考虑到将结合大数据、云计算、深度学习等新兴通信技术,协议应当具备兼容、可拓展性以及负载共享功能[17]。

CoAP协议适用于工业物联网[18],最大的优势是其兼容性高,其与已有现成网络架构和传输协议的系(如智能电表)[19]完美匹配。

轻量级的CoAP协议适合泛在电力物联网应用场景,但考虑到泛在电力物联网接入节点增多,网络流量负载增大、链路不稳定性等问题,很容易造成网络拥塞,还需进一步研究CoAP协议从应用层额外提供保障可靠性传输的拥塞控制机制[20]。

## 2 L-CoCC算法

本文根据泛在电力物联网的链路特点提出一种基于链路稳定性的L-CoCC算法。默认超时重传值 $R^{TO}$ 的合理估计是CoAP协议拥塞控制的关键因素[21]。在 $R^{TO}$ 时间内接收方未收到数据包,发送方将该数据包视为丢失,并启动包重传。当 $R^{TO}$ 小于实际网络往返时延 $R^{RT}$ 时,会造成不必要的虚假重传。但当 $R^{TO}$ 过大时,又会造成重传等待时间过长,形成网络拥塞影响传输效率。

### 2.1 3种 $R^{RT}$ 估计

L-CoCC算法利用强 $R^{RT}$ 、弱 $R^{RT}$ 和失败 $R^{RT}$ 这3种估计器确定准确的网络链路状态进而估计 $R^{TO}$ ,并提供自适应的拥塞控制。L-CoCC算法在请求-响应交互模型中加入重传计数(Retransmission Count, RC)变量,以减小由于物联网环境波动而引起的 $R^{RT}$ 负变化。因此使用下界 $R^{TO}$ 约束方法,避免了不必要的虚假重传,且老化更新机制限制了 $R^{TO}$ 值的有效性,从而提高了所提方案的效率。3种 $R^{RT}$ 估计器分别将成功、延迟和失败的包视为强、弱和失败的 $R^{RT}$ 。

在CoAP协议中实现的大多数拥塞控制机制的假设只由网络拥塞引起包丢失。但从大规模物联网场景分析,当节点在争夺网络资源时,会出现丢包和延迟,导致节点和通信链路都出现网络拥塞。节点级的网络拥塞大多为数据包延迟,而链路级的拥塞会导致数据冲突。因此,在 $R^{TO}$ 测量中,L-CoCC算法同时考虑了链路级拥塞碰撞和节点级拥塞延迟的影响。

L-CoCC算法运行3种 $R^{RT}$ 估计器,分别利用不同ACK消息进行 $R^{RT}$ 估计,强 $R^{RT}$ 表示数据包从客户端节点到服务器节点的成功传输,即时ACK用于计算数据包已成功传输的强 $R^{RT}$ ;弱 $R^{RT}$ 表示数据包从客户端节点到服务器节点的延迟传输,延迟ACK用于计算在重传中传输数据包的弱 $R^{RT}$ ;失败 $R^{RT}$ 表示从客户端节点到服务器节点的数据包传输失败,NOACK用于计算失败的 $R^{RT}$ ,对于该 $R^{RT}$ ,相同的包已被重新传输,并且最多4次都未能交付,失败 $R^{RT}$ 表明存在着链路级的冲突,即有些数据包丢失。根据上述方法,实现了 $R^{RT}$ 的自适应计算。

### 2.2 动态 $S^R$ 和 $R^{TO}$ 估计器

L-CoCC算法利用 $R^{TO}$ 来确保传输可靠性和最小的重传。CoAP协议默认拥塞机制使用平滑 $R^{RT}$ ( $S^R$ )和 $R^{RT}$ 抖动值( $R^{RTVAR}$ )。CoAP协议根据最大延迟和处理延迟来决定数据包传输的 $R^{RT}$ 值,在收到第一个 $R^{RT}$ 样本后,L-CoCC算法计算 $R^{TO}$ 如式(1)~式(3)所示:

$$S_k^R = \frac{\sum_{i=1}^K R_i^{RT}}{K} \quad (1)$$

其中:平滑时延 $S_k^R$ 为先前 $R^{RT}$ 的平均值; $K$ 为4。

$$R_k^{RTVAR} = (1 - \delta) \times R_{k-1}^{RTVAR} + \delta \times |R_{new}^{RT} - R_{k-1}^{RT}| \quad (2)$$

其中: $R_k^{RTVAR}$ 为连续 $R^{RT}$ 值的抖动大小,即两个时刻 $R^{RT}$ 的差值; $R_{k-1}^{RTVAR}$ 为上一时刻往返抖动值; $R_{k-1}^{RT}$ 为上一时刻 $R^{RT}$ 值; $R_{new}^{RT}$ 为当前 $R^{RT}$ 值; $\delta$ 为1/8。

$$R_k^{TO} = S_k^R + \gamma \times R_k^{RTVAR} \quad (3)$$

其中: $R_k^{TO}$ 为对第 $k$ 次重传 $R^{TO}$ 预估; $\gamma$ 为往返抖动值的权重比,默认为4。

对于后续传输,每次重传估计都是必须的。超时重传时间总体估计值( $R_{overall}^{TO}$ )是当前计算的 $S^R$ 和上一步获得 $R_{overall}^{TO}$ 值的平均值。为了避免测量当前 $R^{TO}$ 值后 $R^{TO}$ 陡增或陡减, $R_{overall}^{TO}$ 保持 $R^{TO}$ 值一致增/减。 $R_{overall}^{TO}$ 估算如式(4)所示:

$$R_{overall}^{TO} = \frac{1}{2} \times S_{new}^R + \frac{1}{2} \times R_{overall}^{TO} (p) \quad (4)$$

强 $R^{RT}$ 、弱 $R^{RT}$ 和失败 $R^{RT}$ 分别表示正常网络中、节点级阻塞延迟和链路级冲突的 $R^{RT}$ 。运用L-CoCC算法对 $S^R$ 测量进行了改进,以区分网络拥塞和链路冲突情况。在实际应用场景中,3种 $R^{RT}$ 的大小关系是:强 $R^{RT}$ <弱 $R^{RT}$ <失败 $R^{RT}$ 。用FR、SR和WR分别表示最后4次传输中失败 $R^{RT}$ 、强 $R^{RT}$ 和弱 $R^{RT}$ 的频率。其中, $\alpha$ 值总是大于(1- $\alpha$ )。

在链路冲突情况下FR>SR>0,即失败 $R^{RT}$ 的贡献要高于强 $R^{RT}$ 。由于链路级别冲突,有更多的数据包被丢弃,只能及时接收到少部分数据包。在这种情况下,拥塞窗口将处于最小状态。链路冲突中的 $S_k^R$ 如式(5)所示:

$$S_x^R = \left( (1-\alpha) \frac{R_{strong}^{TT}}{S_{time}} \right) + \left( \alpha \frac{R_{fail}^{TT}}{F_{time}} \right) \quad (5)$$

在节点级网络拥塞下  $WR > FR$ , 即弱  $R^{TT}$  的贡献要高于失败  $R^{TT}$ 。因此, 发送方在最大程度上增加  $R^{TO}$ , 降低了网络拥塞影响, 最终提高物联网场景下的通信性能。节点级网络拥塞下的  $S_x^R$  如式(6)所示:

$$S_x^R = \left( (1-\alpha) \frac{R_{fail}^{TT}}{F_{time}} \right) + \left( \alpha \frac{R_{weak}^{TT}}{W_{time}} \right) \quad (6)$$

在正常情况下,  $SR$  比  $WT$ 、 $FR$  大。因此, 无需增加拥塞窗口的  $S_x^R$  如式(7)所示:

$$S_x^R = \left( (1-\alpha) \frac{R_{strong}^{TT}}{S_{time}} \right) + \left( \alpha \frac{R_{weak}^{TT}}{W_{time}} \right) \quad (7)$$

将式(5)~式(7)代入式(4), 可估计出总体  $R^{TO}$  值。利用3个  $R^{TT}$  估计器来动态自适应调整  $R^{TO}$  值, 但是在发端接收 ACK 时不知道是第几次重传, 导致计算出  $S_x^R$  不准确, 所以拟加入重传次数 RC 字段以确定当前是第几次重传并引入  $R^{TO}$  下限机制和老化更新机制。

### 2.3 RC 值和 $R^{TT}$ 波动估计 $R^{TO}$

在标准 CoAP 协议中, 节点使用消息 ID 识别所有 CoAP 确认消息, 并帮助系统检测重复的数据包。对于拥塞情况, 无法判定发端收到的响应是重传的延迟 ACK 还是即时 ACK。当发端计划向接收节点发出多个连续请求时, 对于拥塞场景仅使用消息 ID 来确定数据包是不够的, 在估计  $R^{TO}$  时导致2个问题: 1) 确定太大的  $R^{TO}$ , 比  $R^{TT}$  易导致包延迟, 影响网络吞吐量; 2)  $R^{TO}$  选择不当小于  $R^{TT}$ , 导致不必要虚假重传和网络拥塞造成的数据包丢失。

因此, L-CoCC 机制提出利用 RC 字段来解决  $R^{TT}$  的不公平性<sup>[22]</sup>, 避免  $R^{TO}$  的陡增和过度收缩。为了附加 RC 新的字段, 使用 CoAP 协议中头部控制消息的 Option 选项。利用此字段信息, 发端在收到 ACK 包时提取 RC 值以明确重传次数。当不发生重传时, RC 值被赋值为零。对于每个重新传输的包, RC 值递增1。该机制利用 RC 计数信息, 避免了由于  $S_x^R$  和  $R^{TO}$  测量不当而造成意外延迟和损失。

CoAP 和 L-CoCC 机制传输对比如图3所示。CoAP 的原始拥塞控制机制如图3(a)所示, 可以看出节点 A 收到的 ACK 包中没有重传信息, 无法得知 ACK 包经历了几次重传, 导致无法准确计算  $R^{TT}$ 。L-CoCC 机制表明节点 A 能够得到正确的  $R^{TT}$ , 因 ACK 包携带了 RC 信息。链路冲突情况下 L-CoCC 机制如图3(b)所示, 发送 CON RC=2 后, 发端收到第一个携带 RC 的 ACK。因此它被认为是弱  $R^{TT}$ , 第一个 CON 请求是一个失败  $R^{TT}$ , 因此失败和弱  $R^{TT}$  的频率都为1。当网络畅通时 L-CoCC 机制如图3(c)所示, 第1个和第2个 CON 请求都失败, 第3次重传后 ACK 包被即时接收。

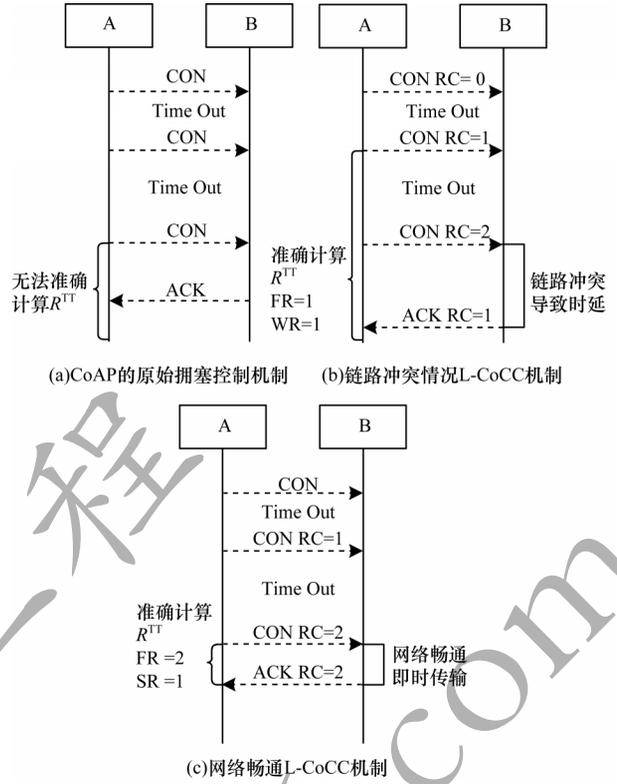


图3 CoAP和L-CoCC机制传输对比

Fig.3 Transmission comparison between CoAP and L-CoCC mechanisms

与其他拥塞控制机制简单的收包时间减去发包时间计算  $R^{TT}$  不同, L-CoCC 算法通过比较具有相同 RC 值的 CON 传输时间和 ACK 的接收时间, 使发送方节点测量数据包的实际  $R^{TT}$ 。

此外, 利用  $R^{TT}$  的波动  $R^{TTVAR}$  对  $R_x^{TO}$  进行加权预估, 最大限度地减少  $R^{TO}$  在后续传输中不必要的扩展和收缩。  $R_x^{TTVAR}$  使  $R_x^{TO}$  体现出  $R^{TT}$  的变化历史,  $\beta$  的值设为 0.125, 返回  $R_x^{TTVAR}$  值如式(8)所示:

$$R_x^{TTVAR} = (1-\beta) \times R_x^{TTVAR} (p) + \beta \times (R^{TT} - S_x^{RTT}) \quad (8)$$

从图3(b)可以看出,  $R^{TT}$  大于  $S^R$ , 因此  $R_x^{TTVAR}$  的当前值主要依赖于过去的  $R^{TTVAR}$ 。相比图3(c)的  $R^{TT}$  值小于  $S^R$ 。因此,  $R_x^{TTVAR}$  值降低。 L-CoCC 机制采用  $R_x^{TTVAR}$  和  $S^R$  的估计算法区分了拥塞和冲突场景, 利用  $R^{TT}$  波动限制了  $R^{TO}$  的上限波动, 并有效动态调整  $R^{TO}$  值。与  $R^{TO}$  上限值一样,  $R^{TO}$  的下限也决定了拥塞控制机制的效率。因此, 利用  $R^{TO}$  收缩方案成功地限制了  $R^{TO}$  下限值。

### 2.4 $R^{TO}$ 限制与更新

$R^{TT}$  的负变化使  $R^{TO}$  比  $R^{TT}$  小, 这种情况会导致虚假重传, 再次出现高碰撞和拥堵。因此, 本文附加了限制  $R^{TO}$  ( $R_{lim}^{TO}$ ) 测量中的  $\max(-L, K \times R^{TTVAR})$  条件, 如式(9)所示:

$$R_{lim}^{TO}(x) = S^R + \max(-L, K \times R_x^{TTVAR}) \quad (9)$$

其中： $L$ 为 $R^{TT}$ 的下限，建议初始 $L$ 值不大于10 s。在随后的 $R^{TO}$ 测量中，如果 $R^{TT}$ 的负变化增加，则 $L$ 值增加10；当 $R^{TT}$ 出现正变化时， $L$ 的值重置为10 s。因此，L-CoCC算法决定了 $R^{TO}$ 收缩的最小程度以及CON报文的虚假重发。下限 $R^{TO}$ 对拥塞控制机制的影响如图4所示。

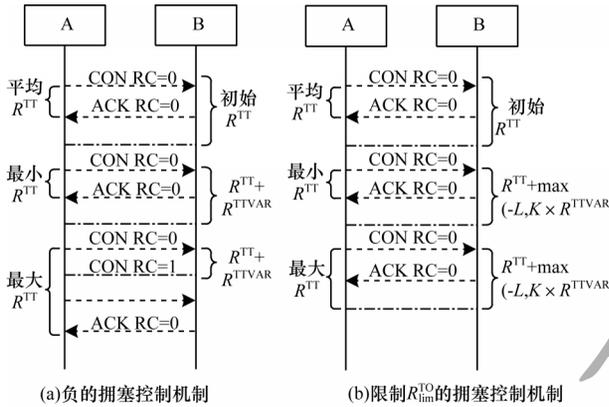


图4 L-CoCC 下限机制

Fig.4 L-CoCC lower bound mechanism

$K$ 的值设置为4， $R_x^{TO}$ 利用 $S_x^R$ 值进行估计，降低了 $R^{TTVAR}$ 对弱 $R^{TO}$ 估计的影响。从图4(a)可以看出，由于 $R^{TT}$ 值极小， $R^{TTVAR}$ 为负的拥塞控制机制。 $R^{TTVAR}$ 大幅减小，导致 $R^{TO}$ 急剧减少，并虚假重传。但L-CoCC算法降低了 $R^{TT}$ 的 $L$ 值，因此发送方节点可以在超时之前收到ACK消息，避免了不必要的数据包重传和延迟。

考虑到泛在电力物联网的链路不稳定因素， $R^{TT}$ 变化很快。当估计的 $R^{TO}$ 值在一段时间内没有更新时，就执行更新老化。L-CoCC平衡了基于 $S_x^R$ 的 $R_x^{TO}$ 和基于 $R_{overall}^{TTVAR}$ 的 $R_{lim}^{TO}$ 之间的 $R^{TO}$ 值， $R^{TO}$ 值如式(10)所示：

$$R_{new}^{TO}(x) = \frac{1}{2} \times R_{overall}^{TO} + \frac{1}{2} \times R_{lim}^{TO}(x) \quad (10)$$

在CoAP中L-CoCC机制使用3种 $R^{TO}$ 估计器、 $R^{TO}$ 值的下界限限制、 $R^{TT}$ 波动处理方案和老化概念，提高了在泛在电力物联网场景下的CoAP性能。

### 3 仿真分析

以网络链路估计为基础从理论上对L-CoCC算法进行定量分析。本文模拟泛在电力物联网场景，选取原始CoAP协议中指数退避机制算法，L-CoCC算法、文献[23]中CoCoA++算法，重复实验分别对3种算法的吞吐量、端到端时延、请求成功率等指标进行比较分析。

#### 3.1 网络拓扑特征

由于泛在电力物联网部署环境的特殊性和复杂性，泛在电力物联网的拓扑结构如图5所示。树状拓扑将多个子节点连接到中央服务器，中央服务器

承担子节点入网、资源调度、数据收集、数据处理等功能。但由于泛在电力物联网资源受限性和传输链路不稳定性，因此节点间通信很容易发生冲突。

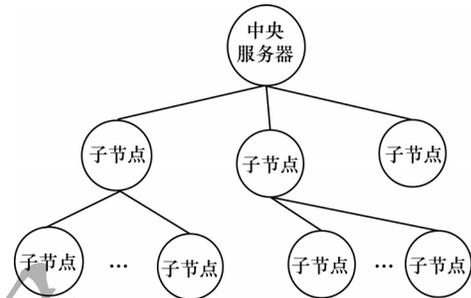


图5 泛在电力物联网拓扑结构

Fig.5 Topology structure of ubiquitous electric IoT

#### 3.2 仿真参数设置

本文使用Windows平台上OPENT Modeler 14.5仿真软件，构建了如图5所示的拓扑仿真测试环境。CoAP消息由每个子节点产生，并定期向拓扑中的中央服务器传输，为了生成不同级别的网络流量，设置不同节点个数 $M$ ，并将数据传输间隔 $T$ 设为2~30 s，模拟网络拥塞情况和链路冲突情况。每个节点的发射、接收功率以及通信范围均相同；其中链路带宽设为50 kb/s，模拟资源受限网络场景；消息重传时间初值为2 s，最大重传次数 $K$ 值设为默认值4。在资源受限环境下网络节点数 $M$ 不同和数据传输间隔 $T$ 不同对各性能指标的影响，其具体数值如表1所示。

表1 仿真参数设置

Table 1 Simulation parameters setting

参数	数值
仿真场景/m <sup>2</sup>	100×50
节点数	50,100,150,200,250
节点通信距离/m	100
链路带宽/(kb·s <sup>-1</sup> )	50
数据传输间隔	2,5,10,15,20,25,30
发包大小/Byte	64
仿真时间/s	100
仿真随机SEED值	64,128,256,512

#### 3.3 仿真结果分析

依据上述参数设定仿真场景。对4种随机SEED值下的的仿真各重复实验20次，随机种子数不同对应的仿真结果相近，并取整体平均值95%置信区间作为最终的仿真数据进行分析。

##### 3.3.1 吞吐量

选取传输时间间隔 $T=5$  s，不同节点个数的吞吐量对比如图6所示。仿真结果表明，相比CoCoA++算法，L-CoCC算法有更高的吞吐量，平均提升9.61%，比原始CoAP协议平均提升34.7%。3种算法在节点数目较少

时,吞吐量差别不是很明显,但随着节点数目增加,网络开始出现拥塞状态,因此CoCoA++和L-CoCC算法吞吐量均高于原始CoAP。

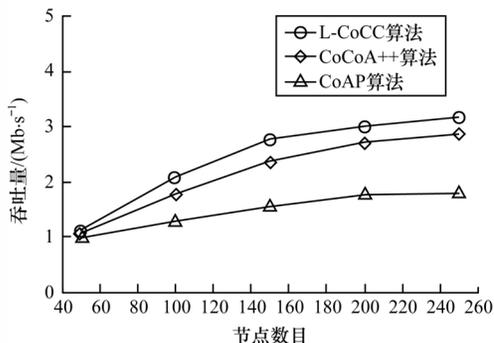


图6 在不同节点数L-CoCC、CoCoA++、CoAP算法吞吐量对比

Fig.6 Throughput comparison between L-CoCC, CoCoA++, CoAP algorithms on different nodes number

不同传输时间间隔吞吐量对比如图7所示。图7选取节点数目 $M=100$ ,仿真结果表明,相比CoCoA++算法,L-CoCC算法吞吐量平均提升7.61%,比原始CoAP算法平均提升14.7%。L-CoCC算法在网络中以最小重传数量交付大部分数据包,当发送周期间隔从2s增加到30s时,由于网络流量网络吞吐量从4.62 Mb/s减少到0.31 Mb/s,但性能仍优于CoCoA++算法和CoAP算法。

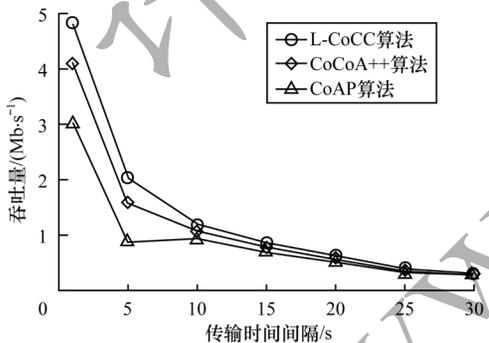


图7 在不同传输时间间隔L-CoCC、CoCoA++、CoAP算法吞吐量对比

Fig.7 Throughput comparison of L-CoCC, CoCoA++, CoAP algorithms on different transmission time intervals

随着网络流量增大,网络开始出现拥塞或冲突,L-CoCC算法使用3种基于整体的 $R^{TO}$ 估计器,从而更精确地估算出 $R^{TO}$ 值,减少重传等待时间和虚假重传次数,及时准确地对 $R^{TO}$ 进行估算,有效提高了数据包交付的成功率,能够维持较高的网络吞吐量。L-CoCC算法在网络出现严重拥塞或链路级冲突时提升了网络性能。

### 3.3.2 端到端时延

不同节点数目端到端时延对比如图8所示。选取传输时间间隔 $T=10$ s,L-CoCC算法有更低的平均端到端时延。相比CoCoA++算法,L-CoCC算法平均降低了9.18%。相比CoAP算法,平均降低了18.31%。在节点数目较少时,CoCoA++算法端到端时延略高于其他两种算法。

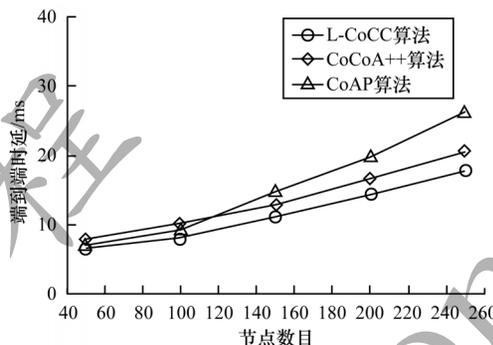


图8 在不同节点数L-CoCC、CoCoA++、CoAP算法端到端时延对比

Fig.8 End-to-end delay comparison between L-CoCC, CoCoA++, CoAP algorithms on different nodes number

不同传输时间间隔端到端时延对比如图9所示。选取节点数目 $M=100$ ,相比CoCoA++算法,L-CoCC算法的端到端时延平均降低了11.98%,相比CoAP算法,L-CoCC算法平均降低了15.47%。当传输时间间隔低时,网络流量较大L-CoCC算法端到端时延明显低于其他两种算法。

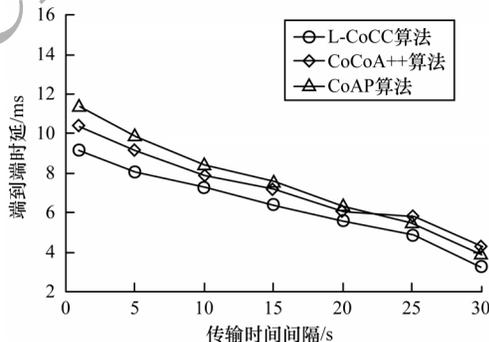


图9 在不同传输时间间隔L-CoCC、CoCoA++、CoAP算法端到端时延对比

Fig.9 End-to-end delay comparison between L-CoCC, CoCoA++, CoAP algorithms on different transmission time intervals

L-CoCC算法使用三种基于整体的 $R^{TO}$ 估计器,在拥塞和冲突环境下结合 $R^{TT}$ 和 $R^{TT}$ 抖动进行 $R^{TO}$ 更新,保证了 $R^{TO}$ 的准确性和实时性。在低网络流量中,由于CoCoA++算法加入选择字段且增加了少量

控制开销,所以其延迟略高;当网络拥塞严重时,L-CoCC算法的时延明显低于其他两种算法。

### 3.3.3 请求成功率

不同节点数目请求成功率对比如图10所示。选取传输时间间隔 $T=10$  s,相比CoCoA++算法,L-CoCC算法的请求成功率平均提升1.93%;相比CoAP算法,L-CoCC算法平均提升3.97%。当入网节点为50个时,3个算法的请求成功率都接近88%,随着入网节点数增加,请求成功率逐渐降低,但L-CoCC算法表现仍优于CoAP和CoCoA++算法。

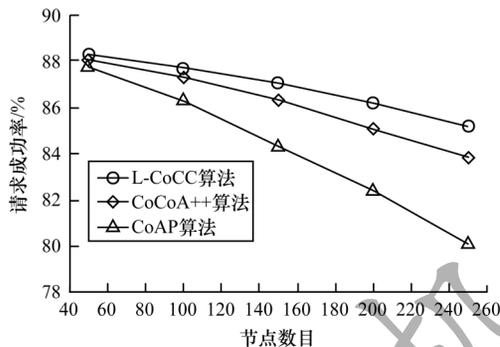


图10 在不同节点数L-CoCC、CoCoA++、CoAP算法请求成功率对比

Fig.10 Request success rate comparison between L-CoCC, CoCoA++, CoAP algorithms on different nodes number

不同传输时间间隔3种算法请求成功率对比如图11所示,选取节点数 $M$ 为100,相比CoCoA++算法,L-CoCC算法的请求成功率平均提升2.73%;相比CoAP算法,L-CoCC算法平均提升了4.34%。在传输时间间隔低时,L-CoCC和CoCoA++算法的请求成功率明显高于CoAP算法,在时间间隔高时三者都接近于94%。

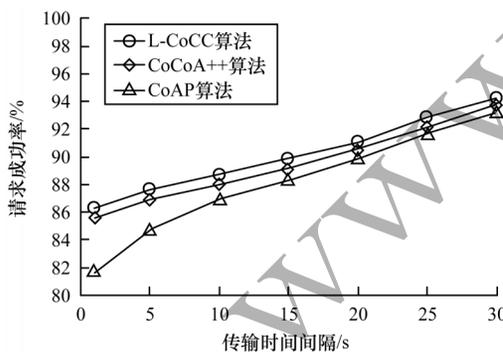


图11 L-CoCC、CoCoA++、CoAP算法不同传输时间间隔请求成功率对比

Fig.11 Request success rate comparison between L-CoCC, CoCoA++, CoAP algorithms on different transmission time intervals

泛在电力物联网网络具有结构复杂、稳定性不高特点,易造成信道冲突。L-CoCC算法可以自适应

地调整 $R^T$ ,避免 $R^T$ 出现陡增和过度收缩,并且利用RC字段明确重传状态和重传次数,避免网络中过度重传进而确保了高请求成功率。

## 4 结束语

针对泛在电力物联网的资源受限特性,以及应用层通信协议上的局限性,本文提出L-CoCC算法。在网络拥塞状态下,使用3种 $R^T$ 估计器对网络链路稳定性进行估计。实验结果表明,相比CoAP协议和CoCoA++算法,L-CoCC算法降低了时延并提高了网络吞吐量和请求成功率,有效缓解了电力物联网场景下的网络拥塞和链路冲突。下一步将结合泛在电力物联网场景,引入动态学习思路,对重传次数和更新时间设置进行研究。

### 参考文献

- [1] 左新强. 国家电网“三型两网”新战略带来全新变革[N/OL]. 中国能源报,2019-04-22(21). [2020-06-02]. [http://paper.people.com.cn/zgnyb/html/2019-04/22/node\\_2222.htm](http://paper.people.com.cn/zgnyb/html/2019-04/22/node_2222.htm). ZUO X Q. State grid's new strategy of "three types and two networks" brings new changes[N/OL]. China Energy News, 2019-04-22(21). [2020-06-02]. [http://paper.people.com.cn/zgnyb/html/2019-04/22/node\\_2222.htm](http://paper.people.com.cn/zgnyb/html/2019-04/22/node_2222.htm). (in Chinese)
- [2] 王毅,陈启鑫,张宁,等. 5G通信与泛在电力物联网的融合:应用分析与研究展望[J]. 电网技术,2019,43(5):1575-1585. WANG Y, CHEN Q X, ZHANG N, et al. Convergence of 5G communication and ubiquitous power Internet of Things: application analysis and research perspectives[J]. Power Grid Technology, 2019, 43(5): 1575-1585. (in Chinese)
- [3] REKA S S, DRAGICEVIC T. Future effectual role of energy delivery: a comprehensive review of Internet of Things and smart grid[J]. Renewable and Sustainable Energy Reviews, 2018(91): 90-108.
- [4] 梁玉珠,沈雪微,邱磊,等. 物联网中基于扩展卡尔曼滤波的移动群体定位[J]. 北京邮电大学学报,2019,42(2): 95-100. LIANG Y Z, SHEN X W, QIU L, et al. Mobile group localization based on extended Kalman filtering in the Internet of Things[J]. Journal of Beijing University of Posts and Telecommunications, 2019, 42(2): 95-100. (in Chinese)
- [5] SHELBY Z, HARTKE K, BORMANN C. The Constrained Application Protocol (CoAP) [EB/OL]. [2020-06-02]. <https://www.potaroo.net/ietf/rfc-pdf/rfc7252.pdf>.
- [6] 宋衍,傅骞. 基于受限网络应用层协议的物联网应用代理研究与实现[J]. 计算机应用,2013,33(11):3010-3015. SONG Y, FU Q. Research and implementation of Internet of Things application agent based on limited network application layer protocol[J]. Computer Application, 2013, 33(11): 3010-3015. (in Chinese)
- [7] KHONGORZUL D, KIM Y, KIM M. Performance analysis of the stop-and-wait automatic repeat request protocol under Markovian interruptions[J]. Frontiers information Technol Electronic Engineering, 2019, 20: 1296-1306.

- [ 8 ] GOMEZ C, DEMIRKOL I, BORMANN C, et al. CoAP simple congestion control/advanced[EB/OL]. [2020-06-02]. <http://tools.ietf.org/id/draft-bormann-core-cocoa-03.txt>.
- [ 9 ] CHOI G, KIM D, YEOM I. Efficient streaming over CoAP[C]//Proceedings of 2016 International Conference on Information Networking. Washington D. C., USA: IEEE Press, 2016: 476-478.
- [ 10 ] HERRERO R. Dynamic CoAP mode control in real time wireless IoT networks[J]. IEEE Internet of Things Journal, 2019, 6(1): 801-807.
- [ 11 ] JARVINEN I, DANIEL L, KOJO M. Experimental evaluation of alternative congestion control algorithms for Constrained Application Protocol (CoAP)[C]//Proceedings of the 2nd World Forum on Internet of Things. Washington D. C., USA: IEEE Press, 2015: 453-458.
- [ 12 ] BHALERAO R, SUBRAMANIAN S S, PASQUALE J. An analysis and improvement of congestion control in the CoAP Internet-of-Things protocol[C]//Proceedings of the 13th IEEE Annual Consumer Communications & Networking Conference. Washington D. C., USA: IEEE Press, 2016: 889-894.
- [ 13 ] ANCILLOTTI E, BRUNO R. Comparison of CoAP and CoCoA+ congestion control mechanisms for different IoT application scenarios[C]//Proceedings of 2017 Symposium on Computers & Communications. Washington D. C., USA: IEEE Press, 2017: 1186-1192.
- [ 14 ] BETZLER A, GOMEZ C, DEMIRKOL I, et al. CoCoA+: an advanced congestion control mechanism for CoAP[J]. Ad Hoc Networks, 2015, 33(7007): 126-139.
- [ 15 ] ANCILLOTTI E, BRUNO R. BDP-CoAP: leveraging bandwidth-delay product for congestion control in CoAP[C]//Proceedings of the 5th World Forum on Internet of Things. Washington D. C., USA: IEEE Press, 2019: 656-661.
- [ 16 ] 姜凯文. 面向智能配电网数据通信的无线传感器网络拥塞控制研究[D]. 合肥: 合肥工业大学, 2017.
- JIANG K W. Research on wireless sensor network congestion control for intelligent distribution network data communication[D]. Hefei: Hefei University of Technology, 2017. (in Chinese)
- [ 17 ] 刘骥, 谭毓银, 符发, 等. MPTCP与CMT-SCTP拥塞控制机制研究[J]. 计算机工程, 2015, 41(4): 117-124.
- LIU J, TAN Y Y, FU F, et al. Study on MPTCP and CMT-SCTP congestion control mechanism[J]. Computer Engineering, 2015, 41(4): 117-124. (in Chinese)
- [ 18 ] IGLESIAS-URKIA M, ORIVE A, URBIETA A, et al. Analysis of CoAP implementations for industrial Internet of Things: a survey[J]. Journal of Ambient Intelligence & Human Computing, 2019, 10: 2505-2518.
- [ 19 ] 程瑛颖, 肖冀, 周峰, 等. 智能电表数据采集新型网络结构与传输协议[J]. 自动化与仪表, 2015, 30(7): 9-12.
- CHENG Y Y, XIAO Y, ZHOU F, et al. New network structure and transmission protocol for smart meter data collection [J]. Automation and Instrumentation, 2015, 30(7): 9-12. (in Chinese)
- [ 20 ] BETZLER A, GOMEZ C, DEMIRKOL I, et al. CoAP congestion control for the Internet of Things [J]. IEEE Communications Magazine, 2016, 54(7): 154-160.
- [ 21 ] 何孟. CoAP自适应拥塞控制机制的研究[D]. 广州: 暨南大学, 2018.
- HE M. Research on CoAP adaptive congestion control mechanism [D]. Guangzhou: Jinan University, 2018. (in Chinese)
- [ 22 ] 李振涛, 任勇毛, 周旭, 等. BBR-TCP协议实验性能评价[J]. 计算机系统应用, 2018, 27(9): 229-235.
- LI Z T, REN Y M, ZHOU X, et al. Performance evaluation of BBR-TCP based on experimental[J]. Computer System Applications, 2018, 27(9): 229-235. (in Chinese)
- [ 23 ] VISHAL R, NATASHA J, SAMANVITA S, et al. CoCoA++: delay gradient based congestion control for Internet of Things [J]. Future Generation Computer Systems, 2019, 100: 1053-1072.