

无线多跳空间网络间歇连通性问题研究

钱雁斌 陈性元 杜学绘 张传富

(解放军信息工程大学电子技术学院 郑州 450004)

摘要 链路的间歇连通性破坏了传统网络存在的基石——端到端原理,而具有长延时、低带宽、高噪音、路径不对称等特点的空间链路进一步为网络连通性提出了严峻挑战。分析了间歇连通性的本质和内涵,给出了多跳间歇连通性的定义,并回到以最基本的 end-to-end 和 hop-by-hop 原理探讨基本的存储-转发机制在间歇连通性环境下的性能,以期能够为克服链路的间歇、实现高效的无线多跳空间网络传输提供有益的帮助。分析与仿真结果表明,无线多跳空间网络中 hop-by-hop 方式的性能通常要高于 end-to-end,但也同样伴随着高协议复杂度和额外的内存、处理器需求等不利因素。

关键词 间歇连通性,空间网络,存储和转发

中图分类号 TP393 **文献标识码** A

Research on Intermittent Connectivity for Wireless Multiple-hop Space Network

QIAN Yan-bin CHEN Xing-yuan DU Xue-hui ZHANG Chuan-fu

(Institute of Electronic Technology, The PLA Information Engineering University, Zhengzhou 450004, China)

Abstract Intermittent connectivity destroyed the building blocks of the traditional network—end-to-end principle, and the space network further posed a great challenge because of the characteristics of long, variable latency, low band width, high noise and so on asymmetric links. This paper analyzed the essence and definition of intermittent connectivity, and revisited the fundamental trade-off between end-to-end and hop-by-hop to discuss the performance of basic store-and-forward mechanism under the intermittent connectivity environment. The analysis and the simulation result indicated that the performance of hop-by-hop way is usually higher than end-to-end in wireless multiple-hop space network. However, it's disadvantage is the higher protocol complexity and additional memory and processing requirements.

Keywords Intermittent connectivity, Space network, Store-and-forward

1 引言

近来对无线链路能力的研究^[1]表明,无线链路往往面临着频繁的链路中断,而且即使是固定节点也同样会出现类似的问题^[2],因此使得许多理论和仿真研究的基本前提假定失效。end-to-end 原理是现有网络存在的基石,端到端连通性的缺失将使得整个网络及其应用失去根基,使得网络的利用率和可用性大打折扣。

最早关注间歇连通性起始于 NASA 和 DARPA 建立的 Interplanetary Internet Project (IPN)^[3],其后,延迟容忍网络研究组 (Delay Tolerant Networking Research Group, DTNRG) 作为 IRTF 的一部分创建,目的是在无法满足持续连通性条件下实现网络互连接。到目前为止,为了适应各种不同网络中的间歇连通性问题,已经提出了许多的研究意见与建议。然而,间歇连通性还没有一个明确的定义,而无线多跳空间网络 (Wireless Multiple-hop Space Network, WMSN) 链路本质上是间歇/周期连接的,同时 WMSN 还具有长延时、低

带宽、高噪音、路径不对称等特点,在对网络连通性造成严重影响的同时,对节点自身的能量、缓存利用率,链路带宽等方面有着更高的要求。因而,本文回到基本的 end-to-end 和 hop-by-hop 原理,来分析间歇连通性的本质与内涵,探讨基本的存储-转发机制在间歇连通性环境下的性能,以期能够为克服链路的间歇,实现高效、可靠的 WMSN 信息传输提供有益的帮助。

2 间歇连通性分析

2.1 网络连通性

任何通信都需要一定形式的连通性。在通常的 TCP/IP 网络中,连通性是指源和目的节点之间存在的一条路径是连通的,并且路径上的所有链路同时具有连通性。文献[5]将这种连通性称为并发链路连通性 (Concurrent Link-based Connectivity, CLC)。CLC 连通性是双向的,其核心是各跳连接同时或者说并发存在。而 WMSN 由于其链路连接的间歇性或周期性,很多时候并不满足 CLC。同时,CLC 定义中并没

到稿日期:2008-12-16 返修日期:2009-03-15 本文受 863 国家重点基金项目(2006AA701416, 2007AA701309, 2008AA7010210)资助。

钱雁斌(1981—),男,博士研究生,主要研究方向为网络安全、空间网络等,E-mail:qianyanbin@sina.com;陈性元(1963—),男,教授,博士生导师,主要研究方向为信息安全、空间网络等;杜学绘(1967—),女,博士,副教授,主要研究方向为高效信息传输;张传富(1975—),男,博士,讲师,主要研究方向为网络仿真。

有加入延时或等待时间等因素,这是因为满足 CLC 连通性的网络如地面有线网络等,正常情况下其延时通常不会大到影响连通性的程度,而对于具有非规则变化长延时、非对称带宽的 WMSN 而言,这一点是不能忽略的,因此本节将探讨间歇连通性的本质,并给出其具体的定义。

实际上广义的连通性概念是包括 CLC 连通性和间歇连通性的,根据其连通性持续程度可以分为持续连通性、间歇中断、间歇连通、极弱性连通、完全自治性操作等,形成一个连通可能性频谱^[6],如图 1 所示。其中,持续连通性对应于我们给出的 CLC 连通性,而完全自治性操作极端情况下是连接完全断开状态,网络节点只能依靠初始配置或者该节点操作者(非远程)自己控制。广义的连通性可以包含除完全自治性操作之外的所有情况。

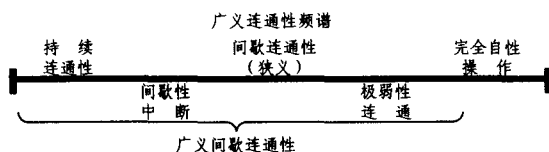


图 1 广义连通性频谱

2.2 间歇连通性定义

持续连通性可以看作狭义的连通性概念,从通信的角度看,连通性更一般的概念是,网络中的节点 A 到节点 B 的通信延迟小于节点 A 通信数据的最大生存周期,即数据若能够在有效时间内从 A 传递到 B,则可认为节点 A 到节点 B 是连通的。这种连通性的定义是单向的,即 A 到 B 是连通的并不意味着 B 到 A 的连通性。影响节点间链路连通性的因素有很多,如链路故障、节点移动性等,而本文主要关注链路间歇中断对连通性的影响,下面给出单跳间歇连通性的定义。

定义 1(单跳间歇连通性, One-hop Link Intermittent Connectivity, OLIC) 对于网络中相邻节点 A, B, 若 A 在 T_0 时刻想要发送数据包给 B, 若存在一个时间 T_A ($T_A > T_0 + D_{AB}$), 使得 A 到 B 存在连通性, 且 $T_A - T_0 + D_{AB}$ 小于数据的最大生存周期, 则称 A 到 B 存在 OLIC 连接。其中 D_{AB} 表示从 A 到 B 的传输延迟。

OLIC 是单向的并且随着时间而发生变化, 在此基础上很容易扩展到多跳间歇连通性, 其定义如下。

定义 2(多跳间歇连通性, Mul-hop Link Intermittent Connectivity, MLIC) 网络中的节点 M_0 在 T 时刻想要通过 n 个中继节点 (M_1, M_2, \dots, M_n) 发送数据包给 M_{n+1} , 若 $M_0, M_1, M_2, \dots, M_n, M_{n+1}$ 相邻节点之间均存在 OLIC 连接, 且 $(\sum_{i=1}^n (W_i + D_{i,i+1})) + T_0 - T < TTL \cap \sum_{i=1}^n W_i < TTL_i$ 为真, 则称 M_0 到 M_{n+1} 存在 MLIC 连接。其中 W_i 是在节点 M_i 处的存储等待时间, $D_{i-1,i}$ 是节点 M_{i-1} 到 M_i 之间的传输延迟。

MLIC 连接除起始节点 M_0 对传输时延的限制外, 中继节点 M_i ($0 < i < n+1$) 都可能因为下一跳链路的间歇失效而需要存储数据, 等待链路恢复后发送。这个等待时间 W_i 同样需要一定限制, 其值如下式所示, 其中 T_i 表示在收到 M_{i-1} 中继的数据后, 节点 M_i 到 M_{i+1} 存在连通性的瞬时刻。

由定义 1 和定义 2 可以看出, CLC 连接类型是 MLIC 连接类型的子集。在有限时间内, WMSN 中实现完全的 CLC 连接不仅代价十分昂贵, 而且就目前的研究现状而言也是不

现实的。

$$W_i = \begin{cases} T_0 - T & i=0 \\ T_i - T_{i-1} - D_{i-1,i} & 0 < i < n+1 \\ 0 & i=n+1 \end{cases}$$

2.3 应对链路间歇的一般方法

实际上, 许多网络都有着类似的 MLIC 问题^[4], 如有着适度延迟和周期连通性的卫星网络、MANETs、无线传感器网络、由于环境因素而导致适度延迟与频繁中断的水下声波网络、未开发领域的 Internet 接入网络、不能总是保证端到端连通性的陆地无线网络等。为了应对 MLIC, 提到网络的可靠性, 目前已经提出了许多方案^[4,7-10,15], store-and-forward 是最基本的方法, 也就是说节点不是简单的转发而是需要将接收到的数据缓存, 并在链路可用时进行转发。现有的方案大致可以分为 3 类: (1) 针对 end-to-end 传输协议的改进, 典型例子是针对 TCP^[8,11,12] 的改进, 但无法从根本上实现解决端到端连通性及源节点存储瓶颈问题; (2) 利用 hop-by-hop 原理将可靠传递的责任从单一的源节点扩展到源节点、中继节点, 典型例子是托管传递方案^[4,13], 但其实现过于复杂, 同时可能会引入新的安全隐患^[14], 并不适合资源受限的 WMSN 网络。 (3) 利用特殊节点的往返运动来传递数据, 如消息摆渡方案^[10], 这种情况较为特殊, 不具有通用性。

3 间歇连通性建模与分析

针对当前应对 MLIC 环境的各种研究, 文献^[9]提出回到最基本的 end-to-end 和 hop-by-hop 来探讨间歇连通性的本质及影响, 但重点考虑的是失败概率及链路中断周期对失败概率的影响。本文进一步完善间歇连通链路模型, 并从成功递送率、发送次数、缓存使用量等方面探讨了 MLIC 对网络可靠性的影响, 比较了 end-to-end 和 hop-by-hop 两种方式。

3.1 简单间歇连通链路模型

为了简化问题的复杂度, 我们考虑一个有着 H 跳链式拓扑, 它仅有唯一源、目的对。在每一跳 i , 链路传输要么成功要么失败, 服从参数为 q_i 的 (0-1) 分布模型。如果出现发送失败, 数据最多可以被重传 $L-1$ 次。End-to-end 重传在源节点处实施, 而 hop-by-hop 在传送失败的中继节点处重传。

H 跳链式拓扑中的每一跳的链路简单的分为两个状态: (1)“on”状态, 数据可以顺利通过该跳链路; (2)“off”状态, 链路中断, 数据无法通过。因此每一跳的链路状态都可以模型化为一个两个状态的离散时间马尔科夫链, 在“on”和“off”两个状态之间交替变换。状态转换以一个离散时间步骤发生。假设转换时间、停留在某个状态的时间、每跳数据发送时间均保持一致, 设从“on”到“off”以及从“off”到“on”的转换概率分别为 p_1 和 p_2 , 对应地从“on”到“on”和从“off”到“off”状态的概率分别为 $1-p_1$ 和 $1-p_2$, 则其平稳分布概率分别为:

$$P_{on} = \frac{p_2}{p_1 + p_2}, \quad P_{off} = \frac{p_1}{p_1 + p_2}$$

定义其平均链路连通率为 $q_i = P_{on}$, $0 \leq i \leq H-1$, 同时链路的一步、 n 步状态转移矩阵分别为:

$$P = \begin{pmatrix} 1-p_1 & p_1 \\ p_2 & 1-p_2 \end{pmatrix}, \quad P(n) = P^n$$

如果 $p_1 + p_2 < 1$, 则 $P_{on} < P_2 = 1 - P_1$, 说明转移矩阵表示的过程是相关的, 例如若当前为“on”状态, 则下一时刻为

“on”的概率要高于“on”状态的平稳分布概率,说明下一时刻很有可能是“off”状态。相反地,若 $p_1 + p_2 > 1$, 则 $P_{ON} > P_2 = 1 - P_1$, 说明转移矩阵表示的过程也是相关的, 例如若当前为“on”状态, 则下一刻为“on”的概率要低于“on”状态的平稳分布概率, 说明下一时刻很有可能仍然是“on”状态。而当 $p_1 + p_2 = 1$ 时“on”和“off”状态之间完全独立, 即下一刻链路状态与当前链路状态无关。本文暂时仅考虑二者独立的情况, 另外两种情形留在将来进一步研究。

3.2 成功递送率与发送次数

无论是 end-to-end 和 hop-by-hop 传输, 其重传次数都是有限的, 设最大重传次数为 $L-1$ ($L \geq 2$, 即至少重传一次), 即最大发送次数为 L ; 对于 end-to-end 情况, 源节点最多发送 L 次; 对于 hop-by-hop 情况, 每个节点最多发送 L 次。两种方式最大发送次数都等于 $H * L$, end-to-end 机制下, 如果每次前 $H-1$ 跳都成功传递, 而最后一跳传输失败, 则可以达到最大发送次数。hop-by-hop 机制中, 若每一跳的前 $L-1$ 次传递都失败, 最后一次成功可以达到最大发送次数。

假定相邻两跳链路状态是独立的, 但是成功递送率的计算, 还是必须考虑上一跳成功与否, 在 hop-by-hop 机制中只有上一跳成功发送, 考虑下一跳是否能够成功才有意义, 即经过 i 跳成功递送的概率为:

$$P_{s,hop}^i = P(S_i^{(L)} | S_{i-1}^{(L)}), 2 \leq i \leq L \quad (1)$$

其中, $S_i^{(L)}$ 表示在第 i 跳经过 k 次 ($1 \leq k \leq L$) 尝试后成功发送, 而 $S_{i-1}^{(L)}$ 表示第 $i-1$ 跳成功发送。而在 end-to-end 机制下, 第 i 次成功发送的概率为:

$$P_{s,end}^i = P(S_i^{(H)} | F_{i-1}^{(H)}), 2 \leq i \leq L \quad (2)$$

其中, $S_i^{(H)}$ 表示第 i 次尝试成功通过 H 跳, $F_{i-1}^{(H)}$ 表示第 $i-1$ 次尝试未能成功。因此, 结合式(1)、式(2)并考虑各跳之间的相对独立性及转移矩阵表示的两个状态之间的独立性, 可以分别得到两种机制的成功递送率如下:

$$P_{s,hop} = (1 - P_{off}^L)^H \quad (3)$$

$$P_{s,end} = 1 - (1 - P_{on}^H)^L \quad (4)$$

根据成功递送率可以计算发送次数的数学期望。对于 hop-by-hop 机制, 第一跳发送次数的数学期望如下:

$$E_1(S_{hop}) = \sum_{i=1}^L i * P_{on} * P_{off}^{i-1} + L * P_{off}^L + P_{off} \quad (5)$$

需要说明的是, 即使报文递送失败, 这种失败的发送仍然有其发送次数, 也同样要消耗空间节点能量。因此, 结合式(1)、式(3)、式(5) hop-by-hop 机制总发送次数的数学期望如下:

$$E(S_{hop}) = \left(\sum_{i=1}^L i * P_{on} * P_{off}^{i-1} + L * P_{off}^L + P_{off} \right) \left(\sum_{i=0}^{H-1} (1 - P_{off}^L)^i \right) \quad (6)$$

对于 end-to-end 机制, 源节点 L 次发送重试之间也并不是完全独立的, 出现 k 次重试的前提是 $k-1$ 次都失败了, 但是在失败的 $k-1$ 次尝试中具体失败在哪一跳, 相互之间是不相关的。End-to-end 机制中一次发送重试发送次数的数学期望如下:

$$E_1(S_{end}) = \sum_{i=1}^H P_{off} * P_{on}^{i-1} * i + P_{on}^H * H \quad (7)$$

在此基础上结合式(2)、式(4)可以得到 end-to-end 机制下总发送次数的数学期望:

$$E(S_{end}) = \left(\sum_{i=1}^H P_{off} * P_{on}^{i-1} * i + P_{on}^H * H \right) * \sum_{i=1}^L (1 - P_{on}^H)^{i-1} \quad (8)$$

3.3 缓存使用量

在前面的讨论中并没有考虑节点缓存的限制, 实际上缓存是重传、可靠性传递的前提, 在实际应用中缓存数量总是有限的, 尤其是空间节点往往缺乏必要的缓存数量。本节重点讨论缓存数量对 MLIC 环境下 end-to-end 和 hop-by-hop 传输的影响(优化的缓存队列管理模型不属本文研究范围, 假设使用简单的 FIFO 队列模型)。

MLIC 环境下由于端到端连通性的缺乏, 可能需要长时间存储数据。end-to-end 机制中, 源节点缓存数据直到数据被确认成功传递到目的节点(或者经 $L-1$ 次重传失败后将数据删除), 此时源节点的缓存数量可能成为一个瓶颈。而 hop-by-hop 机制将缓存数据的责任扩展到所有的中继节点, 实行逐跳的缓存、确认及重传, 但其使用也仍然受到源节点、中继节点缓存量的限制。

图 2 以简单的示例说明了二者之间的明显区别, 图中黑色部分对应于各个节点需要存储等待的时间, D 表示传输延迟, T_0 表示节点 A 开始发送数据的时刻。如图 2(a) 所示, end-to-end 机制下源节点 A 存储数据直到数据被成功发送到节点 D 并得到确认。而在 hop-by-hop 机制下, 数据由 A 发送到 B 后(此时 BC 之间链路为断开状态), B 缓存数据直到 B 到 C 的链路有效, 并通知节点 A(A 可以从缓存中删除该数据)。同样, 数据可以在节点 C 缓存直到可以发送到节点 D。

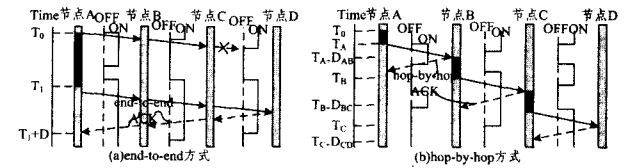


图 2 MLIC 环境下缓存使用示意图

无论是 end-to-end 还是 hop-by-hop 都要受到节点缓存量的限制, 只是前者仅限于源节点缓存, 而后者可以利用网络中继节点缓存。具体的缓存使用量与网络流量、发送频率、重传机制、间歇链路模型等有关, 下一节中将通过仿真试验来探讨 MLIC 条件下缓存数量对 end-to-end 和 hop-by-hop 机制的影响。

3.4 延迟时间

在成功递送数据的前提下, 延迟时间的大小标志着网络性能的优劣。延迟时间可以简单地看作是发送次数和重传时间间隔(RTO)的函数, 而发送次数的数学期望已经在 3.2 节中给出。间歇连通性模型应该支持任意的重传策略, 令 $rto(k)$ 表示第 k 次重传到第 $k+1$ 次重传的时间间隔, 即在第 k 次重传失败后, 等待 $rto(k)$ 时间后开始第 $k+1$ 次重传。常数 RTO 是最简单的重传策略, 例如 $rto(k) = 2k, 1 \leq k \leq H$ 。而指数 RTO 是最常用的重传策略, 通常开始于一个恒定值, 但是每一次重传都将重传超时设定翻倍(有一个上限), 即 $rto(1) = m$ (常数), $rto(k) = m * 2^{k-1}, 2 \leq k \leq L$ 。延迟时间可以看作是发送次数和 $rto(k)$ 的函数, 假定 end-to-end 和 hop-by-hop 采用相同的重传策略, 此时延时与发送次数成正比。

4 仿真与结果分析

在不考虑缓存限制时, 成功递送率、发送次数、缓存使用量、延迟时间等指标都可以分别看作是 $q_i, H, L, rto(k)$ 的函数, 如果考虑具体的数据流量和各空间节点的缓存限制情况

将更为复杂。在重传策略相同的情况下,延时可以看作是发送次数的函数,因而我们下一步将通过仿真试验,考察不同的 q_i, H, L 对 WMSN 网络的成功递送率、发送次数的影响以及两种机制在缓存使用量方面的差别。

本文使用 OPNET14.0, 在一个 2000m * 2000m 的范围来进行仿真,并假定节点间通过 802.11 链路、2Mbps 带宽无线链路实现通信,采用 MANET 方式组网(路由协议为 ADOV 协议)。仿真中采用相同的源节点及数据发送速率,每次源节点间隔发送 100 个大小为 128k 的数据包给目的节点,并进行 10 次这样的实验、记录 10 次平均值,而目的节点则随着 H 的增加而相应调整。仿真中采用 0.8, 0.5, 0.2 三种不同的平均链路连通率,以构建较好、中等、较差三种不同层次的 MLIC 环境。

4.1 H, L 对成功递送率的影响

首先考虑没有缓存容量限制的情况,图 3 分别对应于 $q_i = 0.2, q_i = 0.5, q_i = 0.8$ 时, P_s 随着 H, L 的改变而变化的趋势,有三个共同趋势:首先, P_s 均随着 H 的增加而逐渐下降,随着 L 的增加而上升。end-to-end 机制对 MLIC 跳数的增加更加敏感,呈快速下降趋势,比如 $q_i = 0.8, L = 3$ 时,随着 H 的增加, P_{s_end} 迅速下降,而对应的 hop-by-hop 机制仍然可以保持较高的成功递送率($H = 10$ 时, $P_{s_end} \approx 28\%, P_{s_hop} \approx 81\%$)。而 hop-by-hop 机制对 L 的增加更加敏感,迅速上升,比如 $q_i = 0.5$, 随着 L 增加到 4, 即使 $H = 10$, hop-by-hop 机制仍然可以实现超过 50% 的递送率,而 end-to-end 机制仍然接近 0。这种结果正是因为 end-to-end 机制在源节点处实施重传,而 hop-by-hop 机制在数据丢失处“就地”重传。其次,当 $L = 1$ (没有重传)或 $H = 1$ 时,两种机制的成功递送率基本一致,此时两种机制没有本质区别;再次,在相同条件下 hop-by-hop 机制均要好于或等于 end-to-end 机制,这与我们的分析是一致的。实际上,MLIC 条件下, hop-by-hop 吞吐量肯定要大于 end-to-end 吞吐量,链路的吞吐量 T 受到链路容量 C 的限制,对于 end-to-end, $T = q^H * C$, 而对于 hop-by-hop, $T \leq q * C$, 因而,对于所有的 $0 \leq q \leq 1$, 后者均要好于前者。

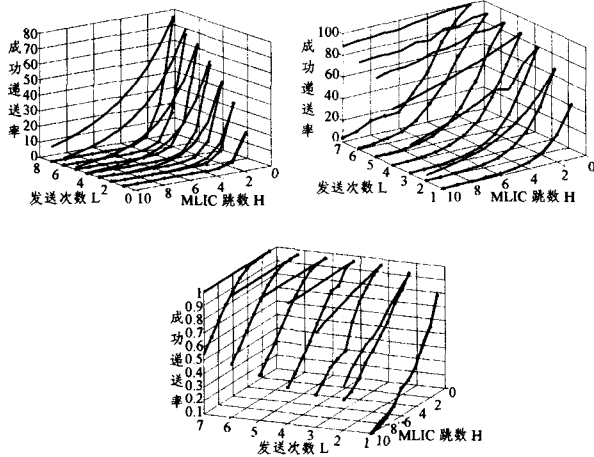


图 3 P_s 随着 H, L 的变化趋势

4.2 间歇持续时间对成功递送率、发送次数的影响

空间节点往往具有有限的资源且相互间距离较远,因此空间节点重传次数应当有效控制,在相同成功率下其发送次数的高低标志着性能的优劣。同时空间网络与地面网络重要的不同之处在于其链路间歇时间可能很长,从几分钟、几小时

甚至到几天(如 IPN),平均链路连通率并不能反应这一状况。因此,引入一个新的参数称之为平均间歇持续时间 AID(Average Intermittent Duration)来研究特定条件下($q_i = 0.8, H = L = 5$), end-to-end 和 hop-by-hop 的成功递送率以及对应的发送次数,以进一步评估空间网络在 MLIC 条件下的执行性能。

仿真中我们采用常数 RTO,对于 end-to-end 而言 $rto(k) = 2H$,对于 hop-by-hop 而言每一跳设置相同: $rto(k) = 1$ 。因此,两种机制在发送数据包后分别经过 $2H * L = 50, 1 * L = 5$ 后数据传递失败。当 $1 \leq AID \leq 100$ 时, end-to-end 和 hop-by-hop 两种机制的成功递送率和发送次数如图 4 所示。可以看出较低的 AID 对于两种机制而言影响较低,而随着 AID 的增加两种机制均明显下降,且二者之间的差距逐渐缩小,说明 hop-by-hop 机制对于 AID 的增加更加敏感,这是因为 hop-by-hop 在发送错误这一跳就地重传,且每次重传间隔过小(相对于 end-to-end),这一点在 AID 较小时是它的优势,但是当这个数值明显小于 AID 时,很容易快速导致发送数据失败,而此时 end-to-end 较长的等待时间可能会“覆盖”一个较长的中断周期。我们的实验中曾尝试着将 AID 增大到 1000 以上,此时两种机制的成功递送率基本趋于一致,甚至有时 end-to-end 机制表现要更好些。

图 5 表示了两种机制发送次数的变化情况, hop-by-hop 机制发送次数较为平稳且在特定的中断周期下有一个最大值,而 end-to-end 机制基本上处于不断上升的过程中,这说明尽管对较大的 AID 可能会迅速降低 hop-by-hop 方式成功递送率,但是在降低发送次数、减少空间节点资源消耗方面始终是要好于 end-to-end 机制的。

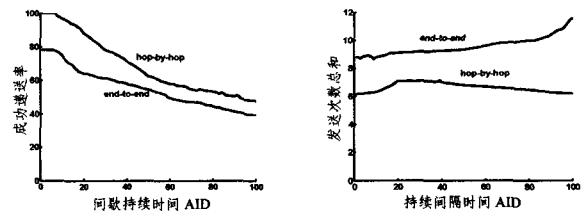


图 4 成功递送率随 AID 变化 图 5 发送次数随 AID 变化趋势

4.3 两种机制的缓存使用量比较

所有先前的仿真过程都没有考虑每一个节点所消耗的存储量,实际上对于具有 MLIC 的空间节点,存储量也是十分珍贵的资源。在 3.3 节中分析了两种机制在网络中使用存储资源的区别,本节将探讨 end-to-end 和 hop-by-hop 机制消耗的总存储空间。仿真中记录的存储量是与时间相关的,存储量 $S = \sum_{i=1}^n t_i$, 其中 t_i 为网络中传递的第 i 个数据在网络中存储的时间片总和。仿真中仍然使用 10 个节点,每个节点有着自己独立的 4M 缓存,适度的设置 $L = 5, q_i = 0.8$ 。

end-to-end 和 hop-by-hop 机制的缓存使用量分别如图 6 所示,结果清晰地表明 hop-by-hop 可以有效降低整个网络的存储量,这是因为 hop-by-hop 机制使得网络可以使用从源到目的之间所有节点的有效缓存。而 end-to-end 机制需要大量的源节点存储,并且由于链路的间歇导致数据长时间在源节点处缓存。需要指出的是,因为 hop-by-hop 机制下存储需求

(下转第 74 页)

pdf

[2] Bucci M, Luzzi R. Design of Testable Random Bit Generators in Cryptographic Hardware and Embedded Systems[C]// CHES 2005. Springer-Verlag, 2005; 147-156

[3] Yin Yifeng, Li Xinshe, Hu Yupu. Fast S-box security mechanism research based on the polymorphic cipher[J]. Information Sciences, 2008, 178(6): 603-1610

[4] Ripeanu M, Foster I. Mapping the Gnutella Network: Macroscopic Properties of Large-scale Peer-to-Peer Systems[C]// IPTPS '02. MIT Faculty Club, Cambridge, MA, USA, March 2002

[5] Viega J. Practical Random Number Generation in Software[C]// Proc. 19th Annual Computer Security Applications Conference, Dec. 2003

[6] Goldreich O. Foundations of Cryptography Basic Tools [M]. Press Syndicate of the University of Cambridge, 2001; 75-89

[7] Menezes A, Oorschot P, Vanstone S. Handbook of Applied Cryptography [M]. CRC, 1996; 169-190

[8] 胡子濮, 张玉清, 肖国镇. 对称密码学[M]. 北京: 机械工业出版社, 2002; 56-57

[9] Molina M, Niccolini S, Duffield N G. A Comparative Experimental Study of Hash Functions Applied to Packet Sampling[C]// Proceedings of ITC-19. Beijing, China, 2005

[10] Stallings W. Network Security Essentials: Applications and Standards[M]. New Jersey: Pearson Education, 2004; 164-166

[11] 张三峰, 陈刚, 于坤, 等. P2P网络可用性关键技术研究进展[J]. 计算机科学, 2008(6)

(上接第 55 页)

在网络节点中是分布式的, 因而非源节点可能需要增加额外的存储负担。

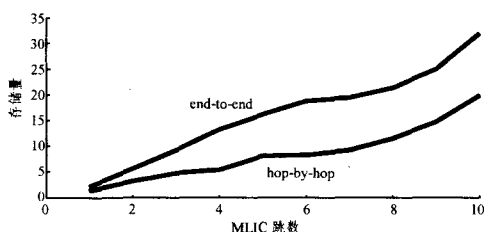


图 6 存储开销比较

结束语 在改善 MLIC 网络性能方面, hop-by-hop 方式的性能通常要高于 end-to-end, 但后者也同样伴随着高协议复杂度和额外的内存、处理器需求等不利因素。为了简化问题的复杂度, 本文在建模和分析中作出了许多的假设, 这使得文章具有一定局限性, 这些也正是我们下一步需要继续研究的方向。

• 模型没有考虑从源到目的节点之间的节点移动性及相应的路由变化, 而路由变化将导致包传递路径长度的改变, 进而对成功递送率、发送次数、缓存使用等产生影响。

• 本文假定每一跳的链路状态改变都是独立的, 然而在真实网络中, 相邻两跳链路上的相互干扰等因素将导致链路状态具有一定的相关性。

• 文章简单地使用 FIFO 机制来管理缓存, 将来的工作将研究 MLIC 环境下更合适的缓存管理机制, 以更好地利用分布于全网的缓存。

实际上, 减少 MLIC 影响可以分为预先式和响应式两种方法。前者在断开连接之前执行预先的行为, 这需要对何时可能发生间歇做出相对准确的预测, 如针对卫星等有着周期性可预测的链路, 但对于一般空间节点这种预测是很难实现的, 因为一般节点很少知道或者没有关于将来网络拓扑的变化知识, 节点移动也是随机的和未知的。后者在连接断开后执行矫正行为。本文主要关注响应方法, 针对卫星等可预测链路间歇连通性研究以及与本文所研究的响应式方法的结合等问题, 也将作为下一步的研究内容。

参 考 文 献

[1] Aguayo D, Bicket J, Biswas S, et al. Link-level measurements

from an 802.11b mesh network[C]// ACM SIGCOMM '04. New York, NY, USA, Aug. 2004

[2] Lenders V, Wagner J, May M. Analyzing the Impact of Mobility in Ad Hoc Networks[C]// ACM REALMAN. Florence, Italy, May 2006

[3] Burleigh S, et al. Delay-Tolerant Networking - An Approach to Interplanetary Internet[J]. IEEE Communications Magazine, July 2003

[4] Cerf V, Burleigh S, et al. Delay-Tolerant Networking Architecture[J]. RFC4838, April 2007

[5] Khan J I Dr. A Commentary on the Nature of Mobility and Connectivity in Space and Implications on the Design of Space Network Protocols[R]. An report to NASA Space Networking Initiative, August 2005

[6] Almeroth K C. Distinguishing between connectivity, intermittent connectivity, and intermittent disconnectivity [J]. Challenged networks, 2007; 1-2

[7] Marshall P. Disruption Tolerant Networking Industry Day BAA 04-13[OL]. <http://www.darpa.mil/ato/solicit/DTN/DTN-industryday.pdf>, Jan. 2004

[8] Akyildiz I F, Zhang X, Fiang J. TCP-Peach+: Enhancement of TCP-Peach for Satellite IP Networks[J]. IEEE Communications Letters, 2002, 6; 303-305

[9] Heimlicher S, Karaliopoulos M, Levy H. End-to-end vs. Hop-by-hop Transport under Intermittent Connectivity [C]// Martin May Proceedings of Autonomics 2007. ICST, Rome, Italy, October 2007

[10] Viswanathan R, Li J, Chuah M C. Message Ferrying for Constrained Scenarios[C]// poster. Proc. of WoWMoM. 2005

[11] Elaag H. Improving TCP Performance over Mobile Networks [J]. ACM Computer Surveys, 2002, 34(3)

[12] Akan O B, Fang J, Akyildiz I F. TP-Planet: A Reliable Transport Protocol for Interplanetary Internet[J]. IEEE Journal of Selected Areas in Communications (JSAC), Feb. 2004, 22; 384-361

[13] Seligman M. Storage usage of custody transfer in delay tolerant networks with intermittent connectivity[C]// Proceedings of ICWN06. June 2006

[14] Farrell S, Symington S, Weiss H, et al. Delay-Tolerant Networking Security. Overview. Internet-Draft[J]. February 2008

[15] 周晓波, 周健, 卢汉成, 等. DTN网络的延时模型分析[J]. 计算机研究与发展, 2008, 45(6): 960-966