

一种提高 TCP 性能的移动自组网 E-OAODV 协议的实现

邬春学 杨朝梁

(上海理工大学计算机工程学院 上海 200093)

摘要 为了改善移动 Ad Hoc 网络环境下 TCP 的性能,对 AODV 的改进协议 OAODV 协议做了一定的修订,得到 E-OAODV(Enhanced Optimal Ad Hoc On-demand Distant Vector)协议。该协议通过在路由失效时使用来自目的节点的 ACK 应答信号所包含的路由信息来尝试继续传输数据,同时通过对 OAODV 和 TCP 两者的修订来实现在路由重建后尽可能地恢复因路由失效所丢失的数据包。NS-2 下的仿真结果表明,E-OAODV 对断线点离源节点较近的情况下的 TCP 性能有较显著的改善,尤其在路由总跳数较大时。

关键词 Ad Hoc 网络,按需驱动距离向量路由协议,传输控制协议,NS-2 仿真

Implementation of an Ad Hoc Network Protocol E-OAODV Improving the Performance of TCP

WU Chun-xue YANG Chao-liang

(Department of Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract In order to improve the performance of TCP used in Ad Hoc networks, an enhanced routing protocol based on AODV was presented, which is called E-OAODV (Enhanced Optimal Ad Hoc On-demand Distant Vector) in this paper. By using routing information included in ACK returned from destination nodes, E-OAODV tries to transfer data to destination node while routing failures occurred. And at the same time, with modifications in both TCP and OAODV, E-OAODV tries its best to recover the lost data caused by routing failures. By NS-2, the simulation result shows that, E-OAODV improves the TCP performance notably when break nodes are closed to source nodes, especially in the routing of multiple hops.

Keywords Ad hoc network, AODV routing protocol, TCP, NS-2 simulation

1 引言

无线 Ad hoc 网络是一种具有高度动态拓扑结构的网络。每一个无线 Ad hoc 网络由一组移动节点组成,它们与传统的蜂窝技术的根本区别在于移动节点之间的通信是在没有固定基础设施(例如基站或路由器)支持的条件下进行的。

鉴于 TCP/IP 已经成为事实上的 Internet 标准协议栈,作为 Internet 扩展的无线 Ad hoc 网络也必将采用它。然而,由于无线 Ad hoc 网中的分组丢失主要由无线差错和节点移动性所致,而 TCP 将所有的分组丢失都归因于拥塞并启动拥塞控制和避免算法,所以若在无线 Ad hoc 网中直接采用传统的 TCP 协议,可能导致端到端的吞吐量无谓降低^[1]。因此,分析 TCP 的性能并研究如何使其适应无线 Ad hoc 网络环境就变得尤为重要。

由于 Ad hoc 网络存在着 BER(Bit Error Rate)高、带宽低、移动性及能量有限等特点,使得原本为固定主机、有线网络设计的 TCP 协议在这种环境下出现了很多不适应的问题。

(1)缺乏有效的错误恢复机制。一旦检测到丢包,TCP 便触发拥塞控制处理过程,首先重传未被确认的包,包括超时时钟指数回退、减小拥塞窗口从而降低发送速率,然后激活拥

塞控制机制,减小慢启动阈值,最后进入拥塞避免阶段以确保拥塞得以解除。如果丢包是由于 Ad hoc 网络的 BER 高或者节点移动造成路由故障时发生的,而不是网络的拥塞,那么 TCP 的这种错误恢复机制会导致协议性能下降,包括吞吐量的下降和延迟的增加。

(2)Ad hoc 网络中有限的传输带宽。由于移动节点可用带宽往往较低,从而使得 TCP 源端的发送速率受到限制,使其用较小的拥塞窗口发送数据。在这种情况下,一旦有数据包丢失,源端就不能收到足够多的重复确认包,从而触发快速重传,而只能通过超时机制恢复,因而降低了可用带宽的使用效率并且增加了延迟。

(3)由于缺乏有效的错误检测和恢复机制,TCP 在 Ad hoc 环境下的能量使用效率也不高。例如,当无线链路上发生了不频繁的随机短暂突发性错误时,TCP 源端便降低其拥塞窗口,然后很保守地逐步增加拥塞窗口的大小。在拥塞窗口缓慢地膨胀的过程中,无错(error-free)的传输机会便被浪费了,并且增加了通讯时间。而当错误持续时间较长(例如衰减信道、链路频繁的突发性错误、长时间的网络分割、网络拥塞)时,TCP 源端尽管降低了其拥塞窗口大小,但仍然在尝试着发送数据,从而造成更多数据包的丢失。尽管吞吐量会有

到稿日期:2008-05-13 本文受国家自然科学基金项目(No. 60573142),湖北省自然科学基金项目(No. 2004ABA076),湖北省教育基金重点项目(No. 2004D003, D200512002)科学研究计划技术创新基金项目(No. J200512001)资助。

邬春学(1964—),男,博士,教授,主要研究领域为网络控制系统、高速网络;杨朝梁(1985—),男,硕士,主要研究领域为网络控制系统, E-mail: lauratdcq2008@163.com.

所增加,但是却消耗了更多的能量,降低了能量使用的效率。而在 Ad hoc 网络中,节点的能量关系到自身以及网络的寿命,要做到尽可能低的能量耗费。

本文通过对改进的 AODV 协议 OAODV 进行分析,结合协议 TCP-BUS 的思想,将提高 TCP 的性能应用在 OAODV 协议上,提高移动自组网协议的 TCP 性能。这个改进后的协议称为 E-OAODV (Enhanced Optimal Ad Hoc On-demand Distant Vector)。

TCP-Bus^[2,3]是一种基于反馈的提高 TCP 性能的方法。它充分地利用了下层路由协议所提供的信息,特别适用于 ABR (Associatively Based Routing)。TCP-Bus 与 TCP-F 类似,除了由中转节点发送 ERDN (Explicit Route Disconnection Notification) 和 ERSN (Explicit Route Successful Notification) 通知发端路由的失败和恢复外,沿路各节点所缓存的分组超时值延长,从而减少了链路中不必要的重传。此外,收端采用选择性重传机制,以提高效率。TCP-Bus 采用定时器和探测帧保证控制信息(如 ERDN 和 ERSN 等)的可靠传输。

NS-2 下的仿真结果表明,E-OAODV 对断线点离源节点较近情况下的 TCP 性能有显著的改善。

2 OAODV 协议的分析及 TCP 性能的改进

OAODV 算法是基于 AODV 算法并加以改进而设计出来的。它继承了 AODV 算法的假设,网络中的所有链路都是双向对称的,即某个源节点通过一条路由可以到达某个目的节点时,该目的节点也可以通过这条路由的反向路由回到源节点。它同时继承了 AODV 的序列号技术和扩展环搜索技术。

2.1 OAODV 协议的路由发现过程

当源节点不具备一条到目的地的路由而又要发送数据到目的地时,源节点广播 RREQ 消息给邻居。在广播 RREQ 以前,源节点在 PATH-DISCOVERY-TIME 时间内缓存 RREQ ID 和 RREQ 源节点的 IP 地址。以这种方式,如果节点再从它的邻居收到同样的 RREQ 报文,它将不对此进行重新处理和转发。RREQ ID 和源节点的 IP 地址联合起来标志一个独一无二的 RREQ 报文。

节点广播出一个 RREQ 以后,节点等待 RREP(或者具有关于正确目的节点路由当前信息的其它控制信息)的到来。如果在 NET_TRAVERSAL_TIME 微秒的时间内没有获得路由,则节点广播另一个 RREQ,试图重新进行路由查找过程,直到在最大 TTL 值时达到了 RREQ-RETRIES 次重传。每一次新的尝试都必须增加并更新 RREQ ID。

邻居接收请求消息,记录下前两跳的 IP 地址,这样可以在逆向路径中通过这个 IP 地址记录逆向路径的下两跳 IP 地址。同时建立一条到源的逆向路径,并设置逆向路径的生存期。如果该节点没有到目的节点的路由,就把收到的 RREQ 传给自己的邻居,直到到达目的节点,或者到达某个具有“更新”路由的中间节点。如果序号小于缓存的路由序号或者已经超过跳计数的规定,则中间节点丢弃该路由请求。目的节点收到 RREQ,就建立到源节点的逆向路由,返回一个 RREP 路由应答报文。并通过 RREP 消息记录正向路径的下两跳信息,单播该 RREP 给源节点。中间节点收到 RREP,就建立到目的节点的正向路由。

2.2 OAODV 协议的路由维护过程

OAODV 路由协议采用 Hello 消息机制进行链路连通性管理,从而对有效路由进行维护。具有有效路由的节点每隔固定时间 T 便广播一个特殊的 RREP 包,即 Hello 消息。邻节点收到 Hello 消息,可对各自的相应路由进行建立或更新。若节点在连续几个 T 的时间内未收到有效路由中相邻节点的 Hello 消息,便认为该链路中断。

在 OAODV 协议中,路由都是通过接收到的数据包或者控制信息反向建立起来,真正起建立到目的节点可用路由作用的,是反向发送回来的 RREP,而 RREQ 只是起查找目的和通知目的回馈 RREP 的作用。要想建立一条到目的路由,必须在同一路径上来回传输 RREQ 和 RREP 才能实现。显然,在路由建立时,取消 RREQ 的使用是不可能的。但在路由建立以后发生断链时,由于目的节点在以前的操作中已经知道自身是传输的目的,甚至在这条路由中的中间节点也知道自己是活动路由的中间节点,因此可不必再使用 RREQ 进行查找和通知目的发出 RREP。当目的节点检测到路由由中间有断路时,可以直接发出一个类似 RREP 的消息 RERREP,来通知建立正向路由。而查找的功能可以通过广播来完成,从而节省了发送 RREQ 所造成的开销和延迟。当路由上的某条链路断开,判断节点自身是当前路由中的上游节点还是下游节点。如果节点自身是下游节点,创建以前两跳为目的 RERREP 消息;如果节点自身是上游节点,设定一个计时器,超时仍未收到 RERREP 时就向先驱列表中的节点发出 RERR 消息。

节点接收到 RERREP 包后,首先取出 RERREP 中的目的节点地址,然后在路由表中进行检索。如果在本地节点的路由表中没有到达目的节点的路由,将 RERREP 中的信息拷贝到路由表中,建立新的到达目的节点的表项;或者虽然有相应的路由表项,但是不可达或者可达目的节点而其跳数大于 RERREP 中的跳数值加 1,则节点更新已有的路由信息,重新建立可用的到目的节点的路由条目,或用更短的路由来代替现有的路由。完成了路由表的更新之后对 TTL 值进行一次判断,以决定是否转发 RERREP 消息。

如果计时器超时,说明本地修复失败,则断开处的节点发送 RERR 包到它的所有前驱节点。每个收到 RERR 包的节点继续把 RERR 包发送给自己的前驱节点。这样,清除所有用到这条断开链路的路由,然后由源节点重新发出 RREQ 查找路由。

2.3 提高 TCP 性能的改进

当 Ad hoc 中的 TCP 发生中断后,TCP 会暂时停止传送数据。等待链路恢复后,才继续传送数据,而这个等待的过程可能会很长,使得 TCP 的效率很低。实际上,在这个长的等待过程中,TCP 并不是一定没有路径来传送数据,可能仅仅是没找到。实际上由于 TCP 的特性,数据接收方在收到数据后会以 ACK 进行应答。ACK 信号与数据传输途径可能一致也可能不一致,所以当原数据传输途径中断后,ACK 信号仍可能未中断。即当 TCP 发现数据传输中断后,若还能收到 ACK 信号,则说明到目的节点的链路依然存在,故此时 TCP 在等待原数据通道重建的过程中,仍可使用 ACK 信号所携带的路由信息,继续向目的节点传输数据,从而降低等待时间,提高 TCP 效率。同时,相应地修订 OAODV,努力减少路

由失效期间的数据包丢失。

通过对 OAODV 的修改来达到利用 ACK 路径实现路由重建的目的,我们把这个机制称为 E-OAODV(Enhanced Optimal AODV)。其主要的思想来自于本文引言中提及的 TCP-BUS。我们把这种做法实现在 OAODV 上。在 OAODV 进行本地修复 local repair 时,断线点除了广播路由请求信息之外,还会回传信息给源节点,告知已经断线但正在本地修复。断线点到源节点间的节点收到此信息后,对将要到目的节点的数据进行缓存;当断线点收到 reply 时,将回传信息给源节点,告知本地修复 local repair 成功。从断线点到源节点间的节点收到此信息时,将开始缓存的数据释放。以下我们再添加更多的新想法。

当 TCP 收到断线通知后,会先暂停数据发送,暂停 RTO 定时器,并在此时检测是否有回传的 ACK 信号。如果有来自目的节点的 ACK,则将这个 ACK 包的 MAC 源地址换成 IP 地址,来更新前往目的节点的路由表的下一跳(next hop)地址。再加上 OAODV 在路由建立时会生成反向链路的路由表,所以数据包就可以沿着回传 ACK 的路径到达目的节点。在断线点寻找重建路由成功的信息回来前,E-OAODV 就可以利用回传的 ACK 将数据发送出去。

TCP 利用回传的 ACK 路径开始传送数据后,还必须处理因为断线而丢失的数据包。这些丢失的数据包与因为网络拥塞造成的丢失数据不同,必须另外处理。TCP 在收到第一次重复 ACK 时,就会重传丢失的数据,并且不会降低 CWND。而重传的范围将因不同的情形而不同。若此前收到的断线信息是断线,但不做本地修复 local repair,说明没有数据缓冲在中间节点上,此时将重传在收到断线信息之前未被 ACK 应答的所有分组;若此前收到的断线信息是断线,但会进行本地修复 local repair,由于此时有数据被中间节点缓冲了,故我们仅重传重复 ACK 应答号对应的分组。若丢失的数据包不止一个时,步骤同样是收到一次重复 ACK 就重传丢失的数据包,直到断线修复后才恢复原状。

选择在第一次收到重复 ACK 后,就开始重传丢失的分组是为了加快重传进度。在断线点寻找新的路径期间,E-OAODV 不但可以利用回传的 ACK 将数据发送出去,还可以补齐因为断线而丢失的数据包,可以减少 TCP 因为寻找新路由而暂停数据传送的时间,从而提高 TCP 性能。

3 仿真性能分析

本文采用 NS-2 仿真,每个节点的路由由缓存限定参照文献^[4],最佳路由由缓存数量的设定参照文献^[5]。每个路由表记载着 4 个下一跳和每个下一跳的信号强度范围。其余的条件同 AODV 协议。将本文中提出的改进方法 E-OAODV 和原有的 AODV,OAODV 方法从分组投递率、端到端延迟、网络开销 3 个方面进行比较。分组投递率如图 1 所示。

在不同的速度条件下和分组投递率方面,比较 OAODV 协议和对 OAODV 改进后的 E-OAODV 协议。结果证明,改进了的协议具有良好的性能,优于原有的 OAODV 协议。在较低的速度下,原有的协议和改进的协议性能都较好。速度较高时,原 OAODV 因路由频繁无效,无效路径的使用导致分组大幅度丢弃,性能的差别变得更明显,改进的协议具有明显的优势,最大处的优势可达到 20%。端到端延迟如图 2 所

示。

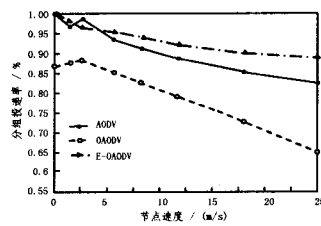


图 1 分组投递率

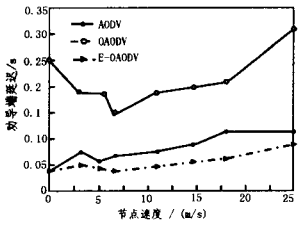


图 2 端到端延迟示意图

从图 2 可以看出,从平均端到端延迟方面来比较,相对 OAODV,改进后的 E-OAODV 在所有速度下,性能都是最好的。网络开销如图 3 所示。

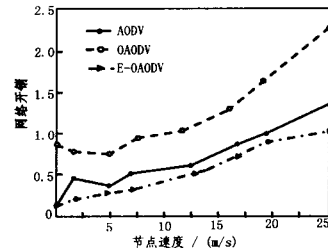


图 3 网络开销示意图

从图 3 可以看出,相对 OAODV,改进后的 E-OAODV 协议具有很低的网络控制开销。在较高的速度下,由于原节点发起路由的次数增加,导致开销上升。

从仿真结果可以看出,相对 OAODV,改进后的 E-OAODV 协议在低速或高速环境下均具有良好的性能,因此对原有的协议是一种优化,性能上更加优越。

结束语 本文首先介绍了 AODV 的改进型协议 OAODV 的运行机制,然后结合 OAODV 和已有的 TCP-BUS 的基本思想提出一种改进的提高 TCP 性能的自组网路由协议。这种利用 TCP 中 ACK 应答信号所携带的路由信息来在发生断路时尽快恢复路由的重建方案,本文称为 E-OAODV。为了能在断路发生时尽快恢复数据传输,而在路由重建时能尽量减少数据包的丢失,这种方案既要修改 OAODV,又要修改 TCP。本文最后给出了该方案的仿真结果,通过比较,改进的 E-OAODV 协议在分组投递率、端到端延迟和网络开销方面均有较显著的改善。

参考文献

- [1] Fu Zhenghua, Meng Xiaoqiao, Lu Songwu. How bad TCP can perform in mobile ad hoc networks // Seventh International Symposium on Computers and Communications, Dallas, Texas, USA, 2002; 298-330
- [2] Prakash R, Sahasrabudhe M. Modifications to TCP for improved performance and reliable end-to-end communication in wireless networks // Wireless Communications and Networking Conference, Sept. 1999; 938-942
- [3] Kim Dongkyun, Toh C K, Choi Yanghee. TCP-BuS: improving TCP performance in wireless ad hoc networks // IEEE International Conference on Communications, Anchorage, Alaska, USA, June 2000; 1707-1713
- [4] Nasipuri A, Castaneda R, Das S R. Performance of multi-path routing for on-demand protocols in mobile ad hoc networks [J]. ACM Mobile Networks and Applications (MONET), 2001, 5: 3392-3491
- [5] Cidon I, Rom R, Shavitt Y. Analysis of multi-path routing [J]. IEEE/ACM Transactions on networking, 1999, 7(5): 8852-8951