

无线多媒体传感器网络实时 MAC 协议

黄志杰¹ 李峰¹ 高强²

(江苏大学计算机科学与通信工程学院 镇江 212013)¹

(北京航空航天大学电子信息工程学院 北京 100083)²

摘要 实时性是无线多媒体传感器网络的重要考虑因素之一,而 MAC 协议能否高效地使用无线信道对保障无线多媒体传感器网络的实时性起着决定性作用。针对无线多媒体传感器网络的业务数据特点,提出了一种基于时隙预留的多信道实时 MAC 协议。该协议在响应流媒体查询之前,事先建立一条从源节点到汇聚节点的时隙预留流路径,从而最小化数据包在每一次转发时的信道接入时延。仿真结果表明,该协议明显减小了流媒体数据的端到端时延与时延抖动,并且具有较好的能量有效性。

关键词 无线多媒体传感器网络,MAC,实时性,多信道

中图分类号 TP393 **文献标识码** A

Real-time MAC Protocol for Wireless Multimedia Sensor Networks

HUANG Zhi-jie¹ LI Feng¹ GAO Qiang²

(Department of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang 212013, China)¹

(School of Electronic and Information Engineering, Beihang University, Beijing 100083, China)²

Abstract Timeliness is one of the most important considerations in wireless multimedia sensor networks(WMSN), and whether the MAC protocol can efficiently use the radio channel plays a decisive role in guaranteeing timeliness of WMSN. According to the data features of the wireless multimedia sensor networks, we proposed a multi-channel real-time MAC protocol based on time slots reservation. The protocol establishes a time slots reservation based flow path from the source node to the convergent node before responding to the streaming media inquiries, thereby it can minimize each channel access delay while transmitting the data package. The simulation results show that the protocol can significantly reduce the end to end delay and the delay dithering of the streaming media data, and it also has better energy efficiency.

Keywords WMSN, MAC, Real-time, Multi-channel

1 引言

无线多媒体传感器网络(Wireless Multimedia Sensor Networks, WMSN)是在传统无线传感器网络(WSN)基础上引入了音频、视频、图像等多媒体信息感知功能的一种新型传感器网络,其节点一般装备有 CMOS 摄像头、微型麦克风以及其它具有简单环境数据采集功能的传感器,它们一般布设在无人值守的环境中自主完成指定的任务,是一种能耗敏感的无基础设施网,在军事、民用及商业领域中具有广阔的应用前景^[1]。近年来,WMSN 技术的研究已引起了国内外学术界的密切关注,并在网络体系结构、节点硬件平台、通信协议设计、协同信息处理、能量节省策略等方面积累了不少研究成果。然而,WMSN 中服务质量(Quality of Service, QoS)这一重要的研究领域尚未受到广泛关注^[2]。与 WSN 主要关注能耗问题不同,WMSN 资源受限、感知媒体丰富以及处理任务复杂等特点决定了其对 QoS 更为敏感。而媒体访问控制(media access control, MAC)协议是用于建立可靠的点到

点、点到多点或多点共享的通信链路技术,是所有数据报文和控制消息在无线信道上进行发送和接收的直接控制者。MAC 协议能否高效地使用无线信道,是保证 WMSN 通信的最关键的因素之一^[3]。因此,针对多媒体业务传输的特点设计具有 QoS 保障的 MAC 协议,是无线多媒体传感器网络的重要问题之一。

2 相关研究

目前,无线传感器网络的 MAC 协议主要分为 3 类:基于竞争的 MAC 协议、无竞争 MAC 协议以及混合型 MAC 协议^[3]。

基于竞争的 MAC 协议由于使用随机延时来减少冲突以达到降低能耗的目的,并且使用延时重来解决冲突,因此它们通常很难保证实时性要求,无法实现多媒体业务的实时传输。

非竞争方式的 MAC 协议的主要思想是为每对节点分配互不干扰的信道实现消息的传输,从而避免了共享信道的碰

到稿日期:2009-12-23 返修日期:2010-03-24 本文受 863 国家重点基金项目(2008AA01Z219)资助。

黄志杰(1983-),男,硕士生,主要研究方向为无线多媒体传感器网络,E-mail:jayzy.huang@gmail.com;李峰(1968-),男,副教授,主要研究方向为图像处理、无线多媒体传感器网络等;高强(1971-),男,副教授,主要研究方向为无线网络与无线通信。

撞,因此具有较好的吞吐量和公平性。然而现有的协议大多并未针对多媒体业务的传输做出改进以提供 QoS 保障。其中 MH-TRACE^[4]利用声音数据包对时延的弱敏感特性,引入一个新颖的分簇算法将网络分成若干两跳簇,以协调信道访问及最小化干扰,并以一定的包时延来换取吞吐量和能量有效性,因此可提供一定的 QoS 保障。但只适用于传输音频数据,对于数据量更大且对时延和抖动更加敏感的视频流难以负担。

混合协议的设计包含了基于竞争协议和无竞争协议的设计要素,可分为基于复用的协议和跨层设计协议。文献[5]提出了一种 CSMA/CA 和 CDMA 相结合的多信道 MAC 协议,通过 CDMA 的伪随机码分配算法,使得每个传感器节点与所有两跳距离内节点的伪随机码都不相同,避免节点之间的无线通信干扰。该协议不需要任何同步时钟,收发器可在无数据发送或接收时关掉。仿真结果表明它比传统 MAC 协议节能 10~100 倍^[5],但主要适合于低负载的网络,难以直接应用于 WMSN。COM-MAC^[6]是最近提出的专为 WMSN 设计的、基于簇的按需多信道 MAC 协议,该协议假设簇头拥有充足的能量和强计算能力,并且有多个收发机。簇头对簇内节点分配信道,使用同一信道的节点由 TDMA 调度,簇头大部分情况直接与 sink 通信,也可以通过多跳中继传输。仿真结果表明它比传统多信道 TDMA 协议有更大的吞吐量和更低的分组延时,但对超强簇头的假定大大限制了其在多数场景的实际应用。MMSPEED^[7]是一种提供包转发机制的多路多速率路由和 MAC 联合协议,每一跳都以区分优先级作为数据包转发调度策略。该协议根据不同类型的数据有不同的实时性和可靠性要求的特点,使用多个包发送速率选项来保障实时性,并利用多路径重复转发来保障可靠性。然而该协议使用基于地理位置的路由,对节点硬件有特殊的要求,并且为保障 QoS,算法设计得过于复杂,没有考虑能量有效性,难以胜任大多数情况下节点能量受限的无线多媒体传感器网络。

综上所述,目前的传感器网络 MAC 协议难以提供能胜任大多数应用场合的 QoS 服务。因此本文针对无线多媒体传感器网络的业务数据特点,提出了一种基于时隙预留的多信道实时 MAC 协议 StreamPathMac。仿真结果表明,该协议明显减小了流媒体数据的端到端时延与时延抖动,并且具有较好的能量有效性。

3 基于时隙预留的多信道实时 MAC 协议

一般来说,WMSN 具有两种基本的服务模式,即基于事件驱动的服务模式和基于查询驱动的服务模式^[8]。不同的服务模式对 QoS 有着不同的需求。基于事件驱动的服务模式的典型代表是异常事件告警服务。异常事件告警服务只有在监测到异常事件时才会向汇聚节点传输用于描述事件的语义信息,这类数据与传统 WSN 中的传输数据类似,具有不可预测及数据量小的特点,但通常需要满足强实时性和高可靠性要求。因此,节点在检测到异常事件时,必须以竞争方式将数据经多跳转发上报汇聚节点。在基于查询驱动的服务模式中,通常由用户发起查询请求,WMSN 响应后将查询结果返回给用户。在 WMSN 中,大部分查询结果数据都是音视频流。这类数据对时延较为敏感,但考虑到网络带宽资源有限,可容忍一定的包丢失。并且,由于流媒体信息的数据量非常

大,整个网络会在响应查询期间负载变得很大,因此。需要充分利用流媒体数据的特点进行合理调度,以充分利用网络资源来满足流媒体数据的实时性需求。

在 WMSN 中,端到端时延主要由源节点和汇聚节点的处理时延(包括模数变化、压缩编码等复杂计算)和多跳转发所造成的时延所构成,后者主要由路由和 MAC 协议决定。为了最大限度地减小转发时延,应该将 MAC 和路由协议联合优化。因此,本文针对上述两种服务模式的特点,提出了一种基于时隙预留的多信道实时 MAC 协议 StreamPathMac。该协议使用基于单跳簇的两层网络结构,将竞争机制与调度机制相结合,在保障两类数据实时性的同时,兼顾能量有效性以及流媒体数据的抖动处理。

3.1 网络模型

在本研究中,假设无线多媒体传感器网络具有如下性质(不失一般性):

- 所有节点的时间是同步的;
- 节点部署后不再移动,且无需人为维护;
- 所有节点具有相同的能力,且地位平等;
- 节点的收发机至少具有两个功率级别:当节点作为成员时,使用较小功率发送,以节省能量;当节点作为簇头时,使用较大功率发送,以便和相邻簇头进行通信;
- 节点的收发机是半双工的,且具有 CDMA 编解码功能。

3.2 分簇方法

在采用单层网络结构的 MAC 协议中,节点为了节省能量,通常需要周期性休眠。因此总有一部分时延是等待下一跳节点被唤醒造成的,并且网络的可扩展性较差。因此,为了保障数据的实时性和网络的吞吐量,在 WMSN 的 MAC 协议设计中通常会引入分簇的思想,例如 COM-MAC^[6]。现有的分簇协议已有不少,其中 HEED^[9]是一种能量有效的分布式分簇算法,性能较好且易于实现。因此本文采用 HEED 协议对网络进行分簇。分簇完成后,成员节点只作为采集数据的源节点,只与簇头进行交互。簇头本身不采集数据,只负责转发数据。为了避免簇间干扰,在分簇完成后可采用文献[5]提出的启发式分布式信道分配算法,使得相邻两跳内的簇头工作在不同信道,并广播本簇所用信道,使其成员节点能够与其进行通信。因为相邻的簇使用了不同的 CDMA 码,所以簇头间通信时发送方必须调整到接收方的信道。传统基于簇的 MAC 协议在分簇完成后,成员节点对簇头的访问方式有两种:一种是簇内 TDMA 方式,另一种则是簇内成员节点与所有邻居簇头以 CSMA 方式对本簇头进行访问。与前者相比,后者具有较小的延迟及较大的灵活性,因此本协议是在传统基于簇的 CDMA 与 CSMA 混合协议的基础上进行优化设计的。

由于本协议还需要与上层路由协议协同设计,因此为了便于讨论,我们假定节点部署后周期性地以下列步骤来重新初始化网络:

- (1)所有节点使用同一信道,按 HEED 协议进行分簇,MAC 层使用 CSMA/CA;
- (2)所有簇头使用文献[5]提出的启发式算法进行信道分配,每个簇头都分得一个伪随机码;
- (3)由汇聚节点开始广播路由建立消息,由近及远在所有

簇头与汇聚节点之间建立单播路由；

(4)在所有簇头与汇聚节点之间建立一棵以汇聚节点为根的广播树,以便当汇聚节点需要向网络发送命令时可以先沿此广播树广播给所有簇头,再由簇头通知各自的成员节点;

(5)簇头节点广播自己将使用的信道 ID(即 CDMA 伪随机码),以便成员节点和簇头之间可以通过该信道进行通信。

图 1 为初始化后的一个网络模型实例。

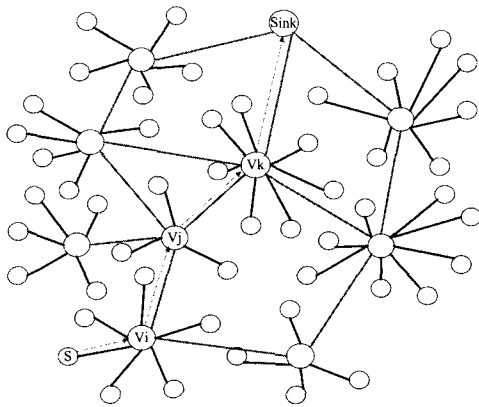


图 1 网络模型实例

3.3 时间帧结构

对于多媒体传感器节点,在采集视频或音频数据时都是有一定的参数指标的,如采样率、分辨率、压缩比等。以视频流为例,若帧率为 N ,则表示源节点在流媒体查询期间每隔 $1/N$ 秒发送一帧图像。并且,由于节点收发机是半双工的,该帧图像必须在 $1/(2N)$ 秒内就能完成接收或发送操作。那么网络初始化后,每个节点以图 2 所示的时间帧为单位进行调度。

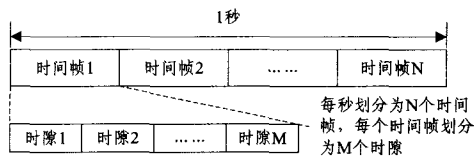


图 2 时间帧结构

即每秒钟分为 N 个时间帧,每个时间帧由 M 个时隙组成,每个时隙恰可以保证节点完整地接收或发送一个流媒体数据包。其中,每个时间帧的时隙数 M 是由时间帧的长度、流媒体数据包的大小和收发机的发送速率决定的。这些时隙又分为 4 种类型,即竞争时隙、发送时隙、接收时隙和广播时隙。竞争时隙用于发送异常事件告警数据或协议所需的控制分组;发送时隙和接收时隙是流媒体查询期间分配给某个流专用的;广播时隙仅用于簇头向其成员节点广播消息。初始状态下,除了每个时间帧的最后一个时隙为广播时隙外,其它时隙均为竞争时隙。簇头只有在广播时隙下才会向成员节点广播消息,因此成员节点在广播时隙下必须处于监听状态。簇头节点在担任簇头期间一直处于工作状态,而成员节点除了广播时隙外一旦没有数据需要发送就进入休眠状态以节省能量。为了避免碰撞,每个簇头需存储自己的时间表(即一个时间帧内的时隙分配情况)和所有邻居簇头的时间表;而成员节点由于只与自己所属簇头交互,因此只需存储自己及其簇头的表。

综上,网络初始化后,对于异常事件告警数据或协议所需

的控制分组,发送节点需检查是否收发双方的当前时隙均为竞争时隙,且该时隙的剩余时间足够完成一个竞争周期。若条件满足,则将收发机调整为接收方所用信道并参与信道竞争。然而,对于流媒体数据,这样的方法显然是低效的。因此我们提出了一种基于时隙预留的流路径优化方案。

3.4 基于时隙预留的流路径优化

当某一个成员节点接收到来自 sink 的查询请求时,如果发现自己满足查询条件,则开始周期性地采集、压缩并发送流媒体数据。通常,这个媒体数据流是每隔一定时间间隔就有一块数据需要发送。以视频流为例,若帧率为 N ,则表示每个时间帧内必须发送出一帧图像,而且发送一帧图像所需的时隙数是可预知的。根据这个特点,可以在传输流媒体数据之前先建立一条从源节点到汇聚节点的时隙预留流路径,以保障流媒体数据的实时传输。

为便于描述,我们引出上邻节点集和下邻节点集的概念。网络初始化后,汇聚节点与网络中的簇头已经建立了路由,且所有簇头都知道自己距汇聚节点的跳数以及邻居簇头距汇聚节点的跳数,则有如下定义:

定义 1(上邻节点集) 距汇聚节点的跳数比本簇头小(即离汇聚节点更近)的所有邻居簇头组成的集合。

定义 2(下邻节点集) 距汇聚节点的跳数比本簇头大(即离汇聚节点更远)的所有邻居簇头组成的集合。

源节点在发送流媒体数据之前,先向簇头发送一个包含流参数的描述包,告之有关即将发送的流的特性,包括这个流欲申请的起始时隙编号及每个时间帧需预留的时隙数。簇头接收到这样的一个描述包时,首先根据即将接收的流的时隙需求,将自己的时间表中相应的时隙设置为该流专用的接收时隙;然后修改描述包中的起始时隙编号,并将新的描述包转发给上邻节点集中可以满足该流的时隙需求的某个簇头;最后根据新描述包把相应的时隙设置为该流专用的发送时隙。以此类推,直到汇聚节点接收到此描述包,然后汇聚节点发送一个“流路径已建立”消息,沿原路径返回给源节点。路径上的簇头接收到“流路径已建立”消息后,把自己的时间表广播给簇内节点以及相邻簇头节点。源节点接收到“流路径已建立”消息后,开始周期性地发送流媒体数据。如果在转发描述包过程中,某个簇头的上邻节点集中所有簇头均无法满足该流的时隙需求,则向上一跳节点发送一个“流路径建立失败”消息。上一跳节点如果是源节点,则不响应此次流媒体查询;如果是簇头,则重新转发该流的描述包。

流路径建立后,流路径上的每一个簇头都分配了特定的时隙,专门用于转发该流媒体数据,而且它们的相邻簇头会根据双方时间表调整对流路径上的簇头的访问时间,簇内节点则调整其休眠周期。这样,从源节点到汇聚节点便建立了一条无碰撞干扰的流媒体数据传输路径,并且汇聚节点从接收到第一帧流媒体数据开始每个时间帧都能保证接收到一帧流媒体数据。查询服务结束后,汇聚节点沿流路径向源节点发送一个“结束查询”命令,路径上的簇头接收到此命令便收回分配给该流的时隙和存储器资源,并将新时间表发给相邻簇头及簇内节点。

为了说明流路径上各节点的调度情况,考虑图 1 中用虚线表示的一条从源节点 S 到汇聚节点 Sink 的流路径,该流路径上各节点在响应流媒体查询期间的时间表如图 3 所示。其

中假设源节点在时隙 m 接收到流路径建立成功的消息,那么在响应流媒体查询期间的每个时间帧内,源节点从第 $m+1$ 个时隙开始每隔一个时隙向簇头 V_i 发送一个流媒体数据包(因为 V_i 是半双工的)。如果碰到广播时隙,则绕过并从下一个时间帧继续,直到发送完一帧数据。流路径上的每个簇头在接收到一个流媒体数据包后,均在下一时隙立刻转发。而汇聚节点则在接收到第一个流媒体数据包后,每隔一个时隙接收到来自源节点的流媒体数据,直至接收完一帧。

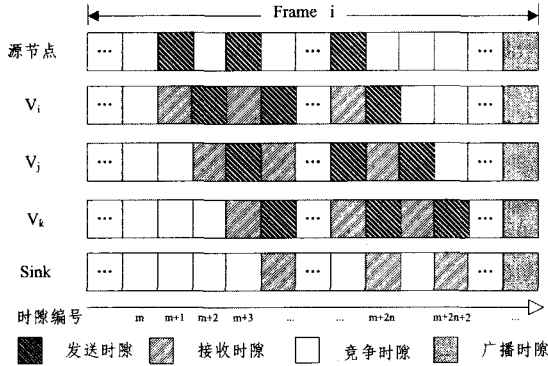


图3 流媒体查询期间流路径上各节点的时间表

3.5 协议实现

(1) 流路径的建立

流路径建立期间,每个簇头均可能接收到“流参数”、“流路径已建立”、“流路径建立失败”3种消息。根据接收到的消息进行不同处理,伪代码如下:

```

1. If(msgName="StreamPathInfo")
2.     If(checkSlotsReqState(msg))
3.         updateMyTimeSlots ( startSlot, slotsReqNum, RE-
CEIVE);
4.         If(myNetwAddr=sinkAddr)
5.             addNewStreamEntry ( streamID, lastHopMac, star-
tSlot, sinkAddr);
6.             replyStreamPathInfo(msg, "StreamPathBuilt");
7.             announceNewTimeSlotTable();
8.         Else
9.             If(nextHops=0)
10.                initNextHops();
11.                forwardStreamPathInfo(msg);
12.            Else //向上一跳节点发送一个‘路径建立失败’的消息
13.                replyStreamPathInfo(msg, "StreamPathFailed");
14.        ElseIf(msgName="StreamPathBuilt")
15.            forwardReplyMsg(msg);
16.            announceNewTimeSlotTable();
17.        ElseIf(msgName="StreamPathFailed")
18.            recoverMyTimeSlots(msg);
19.            removeStreamEntry(msg);
20.            updateNextHops(msg);
21.            spInfo ← createStreamPathInfo(msg); //重现流路径消息并
重新转发
22.            forwardStreamPathInfo(spInfo);

```

簇头在接收到“流参数”消息后,需进行调整并转发,以下为转发部分的伪代码:

```

23. nextHopMac ← -1; //初始化,若能找到满足申请的下一跳
nextHopMac 必为正数
24. nextHopMac ← findNextHopForStreamPath(spInfo);

```

```

25. If(nextHopMac ≥ 0)
26.     addNewStreamEntry ( streamID, lastHopMac, startSlot,
nextHopMac);
27.     updateStreamPathInfo(spInfo); //修正起始时隙
28.     sendDown(spInfo, nextHopMac);
29.     updateMyTimeSlots(startSlot+1, slotsReqNum, SEND);
30.     updateTimeSlotBelong ( startSlot + 1, slotsReqNum, stream-
ID);
31. Else
32.     updateMyTimeSlots(startSlot, slotsReqNum, CONTENTD);
33.     replyStreamPathInfo(msg, "StreamPathFailed");

```

簇头接收到“流参数”消息时,首先确认一下该流所申请的时隙是否均为竞争时隙(第2行)。若是,再进行下一步操作,这是因为邻居簇头的时间表并不能完全保证是最新的。nextHops(第9行)保存着有可能作为“流路径”下一跳的簇头地址,初始化会将本簇头的所有上邻节点均存在 nextHops 中(第10行),而返回“流路径建立失败”的上邻节点会被从 nextHops 中移除(第20行);流媒体路由表保存每一个流的路由信息(第5,26行),为流路径的建立与拆除及流媒体数据包的转发提供支持。此外,当簇头把某些时隙标识为某一个流专用的发送时隙时,还应指定这些时隙所属的流(第29,30行),以便发送时隙到来时可以从对应的流队列获取流媒体数据包进行发送。

(2) 流路径的拆除

Sink 查询结束后会向源节点发送一个‘取消查询’命令。这个消息会沿着流路径转发,路径上的每一个簇头接收到此消息后做如下处理:

```

1. forwardReplyMsg(pkt);
2. updateMyTimeSlots(startSlot, slotsReqNum, CONTENTD);
3. announceNewTimeSlotTable();
4. clearStreamQueue(streamID);
5. removeStreamRoutingEntry(streamID);

```

(3) MAC 调度

MAC 层在接收到来自上层的数据包时,根据数据包的类型将其存放到不同的队列。由于簇头可能为多个流服务,因此每一个流都有一个专门的队列,用流的源节点地址作为流标识;需要在簇内广播的数据存放在广播队列里;报告数据包和其它控制数据包均存放在竞争队列。

网络初始化后,簇头和成员节点都根据自己的时间表来决定当前的动作。每当一个新的时隙到来,先检查时隙的类型,不同时隙采取不同的操作。由于接收时隙是流路径上的簇头专用于接收上一跳的流媒体数据包的,而成员节点只作为源节点而不负责转发,因此成员节点的时间表不会出现接收时隙。

```

1. If(isClusterHead())
2.     If(curSlot=CONTENTD)
3.         setActiveChannel(getChannelOfNode(myMacAddr));
4.         destMac ← -1;
5.         pkt ← selectPktToSend(); //寻找目的节点的当前时隙为
竞争时隙的数据包
6.         destMac ← getDestMacOfPacket(pkt);
7.         If(destMac ≥ 0 and contendCycle(pkt) < availableTime
())
8.             setActiveChannel(getChannelOfNode(destMac));

```

```

9.         beginNewCycle();//启动一个竞争周期
10.        Else//如果找不到满足条件的数据帧,则调回自己的信
            道进行监听
11.            setActiveChannel (getChannelOfNode (myMacAd-
            dr));
12.        ElseIf(curSlot=RECEIVE)//如果是接收时隙,则调到自
            己的信道上进行监听
13.            setActiveChannel (getChannelOfNode (myMacAd-
            dr));
14.        ElseIf(curSlot=SEND)
15.            scheduleStreamQ();//从对应的流队列取出数据包进
            行发送
16.        ElseIf(curSlot=BROADCAST)
17.            scheduleBroadcastQ();//从广播队列取出数据包进行
            发送
18. Else//如果是成员节点
19.     If(curSlot=CONTENTD)
20.         If(contendCycle(getFrontOfContendQueue()) < avail-
            ableTime())
21.             switchRadioToRecv();//因为成员节点之前可能在
            休眠
22.             beginNewCycle();
23.         Else //剩余时间不够完成一个竞争周期
24.             switchRadioToSleep();
25.         ElseIf(curSlot=SEND)
26.             scheduleStreamQ();
27.         ElseIf(curSlot=BROADCAST)
28.             switchRadioToRecv();//成员节点中广播时隙处于接收
            状态已接收来自簇头的命令

```

4 实验结果与分析

为了验证协议的性能,笔者在 OMNET++4.0 环境下对本文所提出的协议和传统基于簇的 CDMA 与 CSMA 混合协议在流媒体数据包的端到端时延、流媒体数据包的时延抖动、流媒体查询的响应时间、报告数据包的端到端时延和节点响应一次流媒体查询的能量消耗 5 个方面进行了仿真实验对比分析。其中各初始参数如表 1 所列,仿真结果如图 4—图 8 所示。

表 1 仿真参数

参数名	参数值
网络面积	1000 米×1000 米正方形区域
节点数目	300
节点布置	随机散布
通信速率	2Mbps
工作电流:收/发	36 毫安
睡眠模式电流	12 微安
通信半径	80 米
流媒体数据源	40k 字节/秒
流媒体数据包长	1000 字节
报告数据包长	100 字节

图 4 为本协议与传统基于簇的 CDMA 与 CSMA 混合协议在响应一次流媒体查询时流媒体数据包的端到端时延随跳数变化的情况。这里的跳数指的是源节点所属簇的簇头到汇聚节点的跳数,而时延指流媒体数据包从产生到被汇聚节点接收之间的时间间隔。不难看出,两种协议的流媒体数据包端到端时延均随着跳数的增加而增加,但本协议的时延明显

低于传统基于簇的 CDMA 与 CSMA 混合协议。

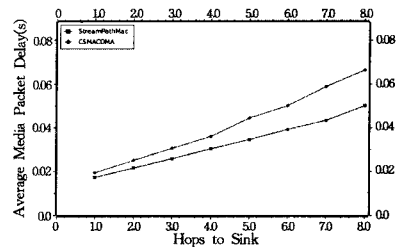


图 4 流媒体数据包的端到端时延随跳数的变化

流媒体数据包的时延抖动是无线多媒体传感器网络的一项重要 QoS 参数。图 5 显示了两种协议的流媒体数据包端到端时延的变化(抖动性)与跳数之间的关系。这里的时延抖动用一次查询期间流媒体数据包端到端时延的标准方差来表示。不难看出本协议的时延抖动较小,并且几乎不随跳数变化。同时我们看到当跳数为 8 跳时,流媒体数据包的端到端时延变化较大,这是由于部分流媒体数据包在多跳转发途中遇到了广播时隙而不得不等待一个时隙的时间。

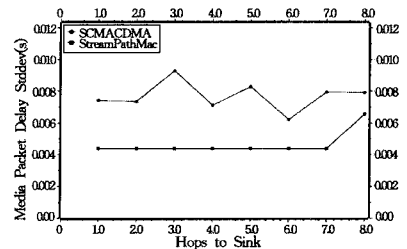


图 5 流媒体数据包的抖动性随跳数的变化

流媒体查询的响应时间是指由汇聚节点向网络中广播流媒体查询命令开始,直到接收到来自源节点的第一个流媒体数据包的时间间隔。如图 6 所示,本协议的流媒体查询响应时间均比传统基于簇的 CDMA 与 CSMA 混合协议要大,并且随着跳数的增加差距越来越明显。其原因是本协议在传输实际的流媒体数据之前,需要事先建立一条源节点到汇聚节点的时隙预留流路径,直到收到路径建立成功的消息才开始流媒体传输。但本协议的流媒体查询响应时间仍然控制在一定的范围内,即使源节点所属簇头距汇聚节点达 8 跳远,响应时间也仅仅 0.28s 左右,这对于绝大部分客户而言都是可以忍受的。

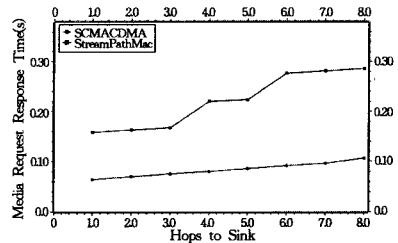


图 6 流媒体查询的响应时间随跳数的变化

图 7 显示了当无线多媒体传感器网络中正在响应一个流媒体查询时基于事件驱动的报告数据包的平均端到端时延与跳数之间的关系。本协议的报告数据包端到端时延比传统基于簇的 CDMA 与 CSMA 混合协议略大。其原因是由于流路径的建立,路径上的簇头有部分时隙已被预留,若报告数据包在多跳转发途中经过流路径上的簇头,则有可能需要等待

(下转第 96 页)

[8] Microsoft Corporation, Hosting ActiveX Controls in the .NET Compact Framework 2.0 [EB/OL]. <http://msdn.microsoft.com/en-us/library/aa446515.aspx>

[9] Bommaiah E, Guo K, et al. Design and Implementation of a Caching System for Streaming Media over the Internet[C]//IEEE Real Time Technology and Applications Symposium, 2000:111-121

[10] Hofmann M, Guo K, et al. Caching techniques for streaming multimedia over the internet [R]. BL011345-990409-04TM, 1999

[11] 尹浩, 林闯, 文浩, 等. 大规模流媒体应用中关键技术的研究[J].

[12] Beveridge J, Wiener R. Multithreading Applications in Win32 [M]. Boston: Addison-Wesley Professional, 1996: 397-413

[13] Stevens W R. TCP/IP 详解, 卷 1: 协议[M]. 范建华, 等译. 北京: 机械工业出版社, 2000: 107-132

[14] de Cuetos P, Ross K W. Adaptive Rate Control for Streaming Stored Fine-Grained Scalable Video[C]//Proc. of NOSSDAV. Miami, Florida, 2002: 3-12

[15] 吴枫, 李世鹏, 张亚勤. 渐进、精细的可伸缩性视频码[J]. 计算机学报, 2000, 23(12): 1276-1282

(上接第 85 页)
争时隙到来才能转发。

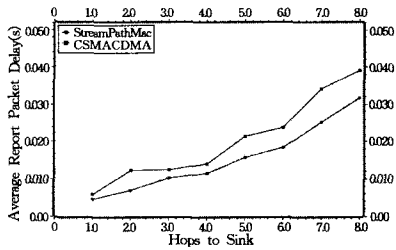


图 7 报告数据包的端到端时延随跳数的变化

基于簇的无线多媒体传感器网络中, 簇头是一直处于工作状态的, 只有成员节点会在无数据需要发送或接收时转入休眠。图 8(a) 显示了本文协议与传统基于簇的 CDMA 与 CSMA 混合协议在响应一次流媒体查询期间某个满足条件的源节点处于各状态的时间长度。不难看出, 在本文协议中源节点的休眠时间比前者长得多, 而处于发送和接收模式的时间均明显较短。在此基础上, 图 8(b) 显示了本实验两种协议源节点的能耗对比, 显然较长的休眠时间使本协议在源节点响应流媒体查询时能节约更多的能量。

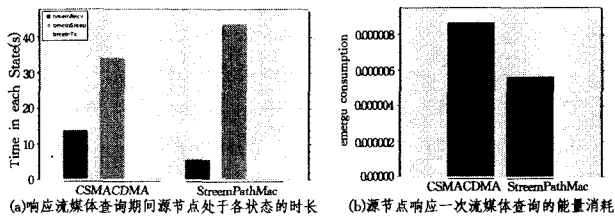


图 8 响应一次流媒体查询源节点能耗对比

结束语 本文针对无线多媒体传感器网络提出了一种基于时隙预留的多信道实时 MAC 协议 StreamPathMac。该协议充分利用了流媒体数据的周期性特点, 在源节点响应流媒体查询之前, 事先建立一条从源节点到汇聚节点的流路径, 路径上的簇头在每个时间帧均为该流预留特定的时隙, 使得源节点的流媒体数据在每一跳都能被及时地接收并转发, 最小化数据包在每一次转发时的信道接入时延。同时, 由于时间表调度与簇间多信道机制的引入, 延长了被查询源节点的睡眠时间, 并且避免了流媒体数据包的碰撞冲突。仿真实验结果表明, 本文协议明显减小了流媒体数据的端到端时延与时

延抖动, 并且达到了较好的能量有效性。虽然本协议对报告数据包的实时性稍有影响, 并且流媒体查询时需要较长的响应时间, 但这并不影响协议的整体性能。最后本协议在设计时并未考虑流媒体数据包的传输差错, 传统基于请求重传的机制显然无法满足流媒体数据的实时性要求, 因此如何针对流媒体数据包较长且可以容忍一定差错率的特点, 结合本协议设计适合流媒体数据的可靠性机制, 是我们将来进一步研究的课题。

参考文献

[1] 罗武胜, 翟永平, 鲁琴. 无线多媒体传感器网络研究[J]. 电子与信息学报, 2008, 30(6): 1511-1516

[2] 孙岩, 马华东. 无线多媒体传感器网络 QoS 保障问题[J]. 电子学报, 2008, 36(7): 1412-1420

[3] 李瑞芳, 李仁发, 罗娟. 无线多媒体传感器网络 MAC 协议研究综述[J]. 通信学报, 2008, 29(8): 111-123

[4] Tavli B, Heinzelman W B. MH-TRACE: multihop time reservation using adaptive control for energy efficiency[J]. IEEE Journal on Selected Areas in Communications, 2004, 22(5): 942-953

[5] Guo C L, Zhong L Z C, Rabaer J M. Low power distributed MAC for ad hoc sensor radio networks[A]//Global Telecommunications Conference[C]. 2001: 2944-2948

[6] Li Cheng, Wang Pu, Chen Hsiao-Hwa, et al. A Cluster Based On-demand Multi-Channel MAC Protocol for Wireless Multimedia Sensor Networks[C]// Proc. IEEE ICC 2008. Beijing, China, May 2008: 2371-2376

[7] Akayya K, Younis M. An Energy-aware QoS Routing Protocol for Wireless Sensor Networks[A]// the Proceedings of IEEE Workshop on Mobile and Wireless Networks (MWN2003)[C]. Providence, RI(2003)

[8] Kulkarni P, Ganesan D, Shenoy P, et al. SensEye: A multi-tier camera sensor network [A]//Proc. of the 13th Annual ACM International Conference on Multimedia '05 [C]. ACM Press, 2005: 229-23

[9] Yonis O, et al. HEED: A hybrid, energy-efficient, distributed clustering approach for ad-hoc sensor networks[J]. IEEE Trans on Mobile Computing, 2004, 3(4): 366-379