维普资讯 http://www.cqvip.com

基于以太网的硬实时通信技术 ARTC

陈 慧 熊光泽 杨仕平
(电子科技大学计算机学院 成都 610054)

A Hard Real-Time Communication Technique Based on Ethernet: ARTC

CHEN Hui XIONG Guang-Ze YANG Shi-Ping
(Computer College, UEST, Chengdu 610054)

Abstract Ethernet as the most popular technology of local-area network, has the advantages of fast speed, low-cost, convenience of installation and good compatibility, but since it exploits CSMA/CD as the MAC protocol, which results in the "uncertainty" of data communication, it cannot be applicable to real-time communication fields, such as industry automated control. This paper presents a hard real-time communication technique based on Ethernet named ARTC, which has well-formed architecture and is easy to be implemented, at the same time, there is no need of modifying the hardware of Ethernet. ARTC utilizes the methods of command/response multiple transmission and bus table to allot low-level communication bandwidth. It supports station management, on-line participation and automatic generation of bus table. ARTC, which is oriented to message, also provides the guarantee of hard real-time communication and has characteristics of reliability and fine extensibility.

Keywords Hard real-time Ethernet, Real-time communication, ARTC, On-line schedule, MS

1. 引言

随着控制、计算机、通信和网络技术的发展及分布式硬实时系统应用需求(如指挥控制系统、工业过程控制等)的快速增长,实时网络的性能、带宽日益受到各应用实体的关注。例如在工业控制中,大量的智能设备经底层网络相互通信,实现现场设备的功能自治和系统的集中管理。但由于传统的实时网络存在带宽较窄、价格昂贵、传输距离短以及开放性等问题,已不能满足现代分布式硬实时系统发展的需要。

以太网(Ethernet)具有快速、协议简单、低耗和兼容性好等特点,获得了蓬勃的发展和应用。带冲突检测的载波侦听多路访问协议(CSMA/CD)解决了信道访问冲突,但是所采用的随机争用介质方式和二进制指数回退机制却使得 Ethernet 不具有实时性。本文在已有研究成果 RTCC[1]的基础上进行了改进和扩展,提出了一种具有完整体系结构并支持分布式控制的硬实时以太网通信技术 ARTC(Advanced Real-Time Communication)。在硬实时通信系统中,消息传输的最大延迟必须得到保障且不能容忍任何的数据丢失,即严格的时间确定性是其关键所在[2]。

ARTC 系统设计满足下列要求:①无需修改现有网络硬件,如以太网卡;② 提供面向消息的硬实时保障;③具有可靠性和灵活性,支持网络站点在线管理和加入;④支持实际应用,具有在线调度功能,简单、易用。

本文详细介绍了 ARTC 体系结构、通信机制和管理机制,并在研究硬实时调度理论基础上提出了 ARTC 在线调度策略。

2. 硬实时以太网 ARTC 通信机制

ARTC 采用主从式命令/响应多路传输方式,其拓扑结构属总线型,如图 1 所示。

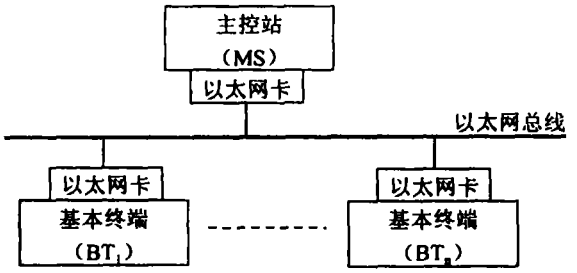

图 1 ARTC 实时以太网拓扑结构

在 ARTC 中,网络节点分为主控站(MS)和基本终端(BT)两种。在一个网段中,MS 只有一个,其余节点为 BT。在实时模式下,总线使用权完全由 MS 所控制,BT 只能响应 MS 的命令被动地发送或接收 ARTC 帧。每个节点都有一张通信表,定义了本节点每条传送数据的逻辑消息号、目的地址和目的端口号、长度等内容。MS 负责执行按调度策略生成的总线表,总线表定义了消息类型及消息的发送地址和端口号、目的地址和端口号、消息长度、启动时刻、最大时延等相关信息。MS 根据总线表的要求在规定时刻向 BT 发命令,该命令规定了 BT 发送数据的逻辑消息号及其长度,BT 收到后立即执行该命令。由此可见,ARTC 中访问控制过程和传输控制过程都由 MS 来完成,这两个过程有机地结合在一起,既能保证节点发送数据的时刻,又能控制节点占用总线的时间。与此同时,ARTC 采用无反馈流控法[15],使得接收端总能在发送端数据到来之前就将旧数据取走。

ARTC 应用模型如图 2 所示。图中各网络节点在以太网 MAC 层上增加了 ARTC 服务层(ARTCS)和 ARTC 协议层

陈 慧 博士研究生,主要研究方向为实时通信技术与应用。熊光泽 博士生导师,主要研究方向为实时计算系统及应用。杨仕平 博士研究生,主要研究方向为实时操作系统的防危核技术与实现。

维普资讯 http://www.cqvip.com

(ARTCP),为上层应用提供可靠的硬实时保障。其中 ARTC 服务层是 ARTC 面向上层应用的接口,负责接收上层信息,并根据信息类型和 CMIB(通信管理信息库)规定的对应规则对其进行分类,然后加上相应的参数,转换成 ARTC 协议层可接受的格式,送 ARTC 协议层进行传输。ARTCP 运行在以太网 MAC 之上,负责收/发 ARTC 帧,用以实现 ARTC 中的消息调度、数据的接收和发送、站点在线管理、消息查询、非实时模式的切换等功能。在 CMIB 信息库中包含了 ARTC 通信所需的各种管理控制信息和应用规则,如节点配置信息、映射规则、网络节点活动表、总线表和通信表、时钟控制信息等。

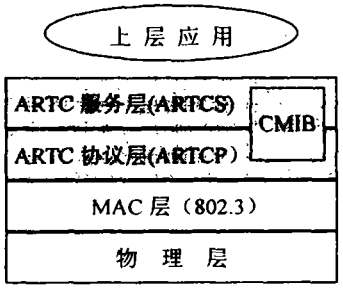

图 2 ARTC 应用模型

ARTCP 是 ARTC 技术的核心。为了实现网络管理和数据传送,ARTCP 定义了帧格式和消息类型、实时模式与非实时模式切换过程等内容。

2.1 帧和消息

ARTCP 定义了命令帧、数据帧和应答帧三类帧格式,如图 3 所示。命令帧只能由 MS 发往 BT,用于命令 BT 发送数据以及探测节点、查询状态和模式切换等;应答帧只能由 BT 传给 MS,用来响应 MS 的命令帧以及收到数据后的应答;数据帧用于进行数据传输,MS 和 BT 都可以发送数据帧。在命令帧和应答帧中,命令参数和应答参数由 ARTCP 根据所传数据的类型设置,用于区分命令帧和应答帧的具体类型。

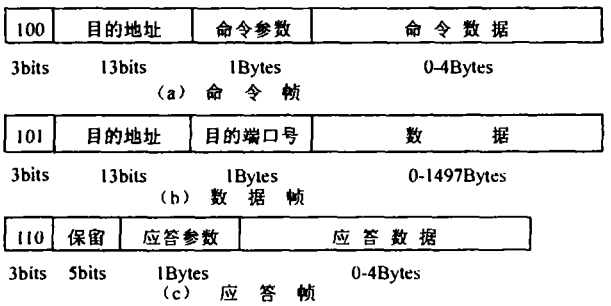

图 3 ARTC 帧格式

消息是分布式硬实时网络传输和调度的基本单位。在 ARTCP 中,消息指包括命令帧、应答帧、数据帧和状态响应间隔在内的传输序列,一条消息的执行表明了指定帧序列的一次完整的数据传送过程。ARTCP 定义了 7 种消息:MS→BT 数据传输消息、BT→MS 数据传输消息、BT→BT 数据传输消息、MS→BT 方式命令消息、MS→BT(s)数据广播消息、BT→BT(s)数据广播消息和 MS→BT(s)方式命令消息。消息按照传播方式可以分为两类:点到点消息(前四种)和广播消息(后三种)。对于点到点消息,BT 接到命令帧或数据帧后立即向 MS 发应答帧,将 BT 当前状态信息(如数据接收是否正确、对命令的应答等)及时反馈给 MS。而对于广播消息,BT(s)不回传应答帧,必要时可以由 MS 逐个进行查询。ARTCP 消息还可以根据传输帧的内容分为数据传输消息和方式命令消息,数据传输消息用于实现网络节点间的数据传送,方式命令消息则用于管理网络事务,如模式切换、站点管理、状态查询等。由于篇幅所限,本文不对具体消息类型的规范进行详细介绍。

2.2 实时模式和非实时模式

ARTCP 既支持实时模式又支持非实时模式,这两种模式的切换由 MS 控制。在实时模式下,所有节点对总线的使用都在 MS 的控制中,保证了在任意时刻只有一个节点使用总线通信。因此,在 ARTCP 正常运行时,帧不会发生碰撞。在 ARTC 中碰撞属于一种传输错误。为了获得最优的实时性能,ARTC 允许传输 IEEE802.3 短帧(帧长小于 64 字节)。对于超过1497字节的数据,节点将其放入多个IEEE802.3 MAC 帧中连续发送。

为了提高网络实时业务轻载时的利用率,ARTC 允许在没有实时消息传送的时段内进行非实时通信,此时 MS 通过广播命令将所有网络节点切换到非实时模式。该命令包含非实时模式的使用时间,这段时间为下一条实时消息启动时间减去当前时间再减去该命令帧的传送开销。BT(s)收到切换命令后,随即进入非实时模式,即所有节点按照传统的 CSMA/CD 竞争使用信道。当非实时模式使用时间结束,所有节点都自动切换到实时模式下等待 MS 的命令。因为 CSMA/CD 采用二进制指数回退机制,所以在非实时模式时间期满时,某些节点的网卡缓冲区中还可能留有回退数据。此时节点将清空缓冲区中的这些数据,这对于保证 ARTC 实时模式的正确执行是非常重要的。

3. ARTC 主控站及网络管理

3.1 主控站的竞争和移交及网络地址分配

在 ARTC 网络初始化前,网络节点包括两种类型:一种是有能力成为 MS 的那些节点,称为逻辑控制节点(Logic Control Node, LCN);另一种不具备这一能力的节点称为普通节点(General Node, GN)。网络初始化时,MS 竞争在非实时模式下进行,每个 LCN 的 CMIB 库中都有一个离线分配的 5bits 逻辑 ID,最先完成启动和配置的 LCN 将成为 MS,其它的 LCN 和所有的 GN 则构成 BT(s)。即在一个网段上可以有多个 LCN,但在某一时刻只能有一个 LCN 成为 MS,其余构成 MS 的冗余。GN 则不分配逻辑 ID。最先完成配置成为 MS 的 LCN 通过命令帧广播其初始化成功的消息及其逻辑 ID,所有网络节点收到该消息后便用缺省地址请求 MS 分配特定的网络地址。当请求节点获得地址后,就进入实时模式。如果 MS 在 1023×2τ(以太网最长回退时间)时间内既没有收到地址分配请求,也没检测到网络冲突,说明节点地址分配全部完成,且所有节点都在实时模式下等待 MS 命令。于是,MS 广播网络节点活动表中包含所有在线节点的网络地址和以太网地址映射信息。ARTCP 网络地址共 13bits,由 5bits 的 MS 逻辑 ID 加上 8bits 的节点代码组成,节点代码通过 MS 在线分配,而 MS 节点地址则由 5bits 的逻辑 ID 和 8 比特 0 组成。

网络初始化后,在使用过程中会有两种情况使得 MS 发生改变:一种情况是 MS 故障,另一种情况是指定某个 LCN 成为 MS。当冗余的 LCN 发现 MS 故障时,将采用上述网络初始化时的方法竞争产生新的 MS。而如果是指定某个在线 LCN 成为 MS, LCN 首先必须通过应答帧向 MS 提出请求。

维普资讯 http://www.cqvip.com

如果 MS 认为不能将主控站移交给该 LCN，它将很快拒绝此请求；如果 MS 同意该请求，为了避免消息丢失，它将选择能完成移交的最近两条相邻消息的空闲时段，向指定 LCN 发命令帧，待收到应答帧后当前 MS 广播移交信息，所有网络节点将其网络地址的前 5bits 改为新任 MS 的逻辑 ID。至此，移交工作完成，新任 MS 将向成为 BT 的前任 MS 分配新的网络地址。

3.2 主控站功能及网络维护

ARTC 主控站的主要功能包括两个方面：执行调度策略为 ARTC 硬实时通信提供保障以及进行网络管理和维护。除网络初始化时节点地址分配外，网络管理及维护大多是通过消息实现的，如新节点加入、在线节点退出及故障节点删除等。

加入的新节点使用缺省网络地址，MS 将周期性地进行探测，为新加入节点分配网络地址并传送网络节点活动表。另一方面，当在线节点需要退出时，它将通过应答帧向 MS 提出请求，得到 MS 同意后退出。如果在线节点连续 n 次（n 在网络初始化时设定）无应答，则说明此节点故障，于是 MS 删除该节点。MS 将对网络节点的所有上述变动进行广播，以便 BT(s) 修改 CMIB 库中的节点活动表。

按照调度策略自动生成总线调度表进行网络调度是 MS 的最主要功能，ARTC 硬实时网络性能的体现和保障完全依赖于总线表排列的优劣程度，因此 ARTC 实时调度策略是 ARTC 的核心和关键。下面我们将对 ARTC 调度机制进行详细分析和介绍。

4. ARTC 硬实时调度机制

ARTC 中由于 MS 完全控制总线带宽的分配，所有节点都只能被动地发送和接收 ARTC 帧，因而更易于满足实时消息的死线。根据消息的周期性，硬实时系统消息分为周期消息和非周期消息两类。ARTC 调度机制基于调度周期任务的单调比率调度算法 RM[5,6] 和调度距离约束任务系统的风车调度算法 Sr[10~12]，以及处理非周期消息的周期服务器法[8,13]，确保网络硬实时通信的实现。该机制可用于静态总线调度表的自动生成，也可用于系统初始化和变化时总线调度表的在线生成和重排，具有较高的总线利用率。

4.1 周期任务模型与单调比率调度算法 RM

周期任务模型描述如下：设一实时任务集为 $S=\{\tau_1,\tau_2,\cdots,\tau_n\}$，任务按周期大小以升序排列，周期为 $T_1,T_2,\cdots,T_n$，执行时间为 $C_1,C_2,\cdots,C_n$，任务 τ_i 按优先级可以被抢占。任务集 S 的处理器利用率定义为 $U=\sum_{i=1}^{n}(C_i/T_i)$。

单调比率调度算法（RM，Rate Monotonic）被证明是最优单处理器静态可抢占硬实时调度算法，其任务的优先级按任务周期 T 来分配，周期越小优先级越高，调度的结果是保证每一个任务在其周期 T_i 内获得 C_i 的执行时间。对于 RM，有如下定理：

定理 1 由 n 个周期任务组成的实时任务集，若 $U\leqslant n(2^{1/n}-1)$，则可由 RM 调度。

4.2 距离约束任务模型与风车调度算法 Sr

距离约束是指同一周期任务相邻两次完成的时间间隔必须小于或等于某一段时间，这样的实时系统称为距离约束任务系统（DCTS，Distance-Constrained Task Systems）。DCTS 系统中，实时任务集模型描述为：DCTS 任务集 $S=\{\tau_1,\tau_2,\cdots,\tau_n\}$，任务按距离约束升序排列，任务 τ_i 的执行时间为 C_i，距离约束为 D_i，距离约束越小优先级越高，任务 τ_i 按优先级可以被抢占。

风车调度 Sr 用一种周期转换技术把所有任务的距离约束转换成调和数，并将该调和数作为任务的调度周期。转化方法如下：取 r 为 $(D_1/2,D_1]$ 间的某个实数，令 $b_1=r,b_i=r\times 2^j\leqslant D_i<r\times 2^{j+1}=2b_i$（$j$ 为 $\geqslant 0$ 的整数），上述过程称为将集合 $D=\{D_1,D_2,\cdots,D_n\}$ 用 $\{r\}$ 进行描述，于是 D 转换成了 $D'=\{b_1,b_2,\cdots,b_n\}$，$S$ 也相应转换为 S'。例如，假设一个系统有 5 个任务 $S=\{\tau_1,\tau_2,\tau_3,\tau_4,\tau_5\}$，距离约束为 $D=\{9.7,10.5,10.7,22.8,25.6\}$，执行时间为 $C=\{1.5,2.0,3.4,1.4,3.0\}$。如取 $r=9.7$，将 D 用 $\{9.7\}$ 表示，则这 5 个任务新的距离约束为 $D'=\{9.7,9.7,9.7,19.4,19.4\}$。然后，任务以转换后的距离约束为周期，按周期性任务进行调度。调度方式类似于单调比率调度算法 RM，任务的优先级按距离约束进行分配。设 $\rho(\tau_i)=C_i/D_i$ 为任务 τ_i 的密度，对于 Sr，有如下定理：

定理 2 设 $\rho(\tau')=\sum_{i=1}^{n}C_i/b_i$ 是整个 DCTS 任务集 S′ 的总密度。如果 $\rho(\tau')\leqslant 1$，则 S 用 Sr 可调度。

使用 Sr 调度，由于任务的周期为调和数，每个任务的调度执行将不存在抖动（jitter），并且可以满足其距离约束。如果将任务集 S 的距离约束 D_i 视作 RM 中的周期 T_i，对于 Sr 具有如下定理：

定理 3 如果一任务集能由 Sr 调度，那么该任务集也必定能被 RM 调度。

4.3 周期服务器法

实时系统中非周期消息是在运行时随机到达的消息，因此必须动态地调度。其优先级根据该消息对时间紧急程度的限制进行分配，而不是消息产生的时间间隔。周期服务器法通过为每一个非周期消息源维护一个周期服务器，即把非周期消息转化成周期消息进行处理，每一个服务器在其周期中都将被保证获得一定的信道访问时间。若已知最坏情况下非周期报文的特性，周期服务器方法可在系统设计阶段对非周期报文提供处理。同时，由于非周期消息的随机性和突发性，消息到达时系统通过确定可用带宽，为非周期消息的执行提供保证。

4.4 ARTC 在线硬实时调度机制

在 ARTC 系统中，消息表示一次完整的传送过程，是网络传输和调度的基本单位，因此不允许悬挂和中断。ARTC 消息符合周期任务模型，消息集用 $M=\{M_i=(C_i,D_i)\}\mid 1\leqslant i\leqslant n\}$ 进行描述，其中 C_i 表示消息所需的执行时间（包括传输开销和节点处理开销），D_i 表示消息的死线（等于周期），M_i 按 D_i 大小升序排列。对一条具体的消息来说，构成消息的各帧长度、总线传输时间、硬件响应时间和协议处理时间（由主机的性能决定）都是确定的，因而最坏情况下消息的执行时间 C_i 是事先确定的。ARTC 在线硬实时调度包括以下三个阶段：收集调度信息，建立消息集 M；进行消息集可调度性判定；按照调度算法生成总线调度表。

（1）*系统初始化并建立消息集合* 系统初始化时，MS 通过命令帧询问各网络节点，根据 BT(s) 的应答信息建立网络消息集 M，并将 M 中的消息按消息死线大小升序排列，即 $M=\{M_i=(C_i,D_i)\}\mid 1\leqslant i\leqslant n\}$，$D_1\leqslant D_2\leqslant\cdots\leqslant D_{n-1}\leqslant D_n$。

（2）*消息集可调度性判定* 现有的调度算法（包括 RM

维普资讯 http://www.cqvip.com

和 Sr)大多属可抢占调度算法，即消息在执行过程中可能被更高优先级的消息所抢占而中断。但是在通信系统中，消息传输是不能被中断的。为了解决这一问题，Sha 等人提出了优先级继承协议[9]，使用该协议必须计算出每条消息被阻塞(由于低优先级消息占用总线而导致较高优先级消息就绪后被阻塞)的时间。然而，Mok 研究表明[14]，在实际应用中，要计算出这一时间进行可调度判定，是一个非常困难的问题。为了避免上述问题，使可调度性判定可以快速地在线进行，ARTC 基于 Sr 调度算法将可调度性判定分为两步：基础判定和仿真判定。在这两个判定过程中，均将消息 M_i 的周期 D_i 作为 Sr 调度中的距离约束。

ARTC 基础调度判定假设 M_i 可抢占，对集合 M 应用 Sr 进行可调度判定。方法如下：分别取 $r=D_1$ 或 $r=D_m/2^{\lceil \log_2(D_m/D_1)\rceil}$，将集合 $D=\{D_1,D_2,\cdots,D_n\}$ 用 $\{r\}$ 进行描述，D 转换为 $D'=\{b_1,b_2,\cdots,b_n\}$。令 $\rho(r)=\sum_{i=1}^{n}C_i/b_i$，如果 $\rho(r)\leqslant 1$，则 M 满足基础判定，并进行下一步仿真判定。注：D_m 为密度最大消息的距离约束，即 $C_m/D_m=\max\{C_i/D_i,1\leqslant i\leqslant n\}$。

在上述基础判定过程中，为了减少应用计算量，距离约束基值 r 并未遍历所有取值，而是确定了两个典型值 D_1 和 D_m。D_m 易于使得转化后的消息集总密度最小，并具有至少 $1/2+C_m/D_m$ 的总线利用率[10]，D_1 则最易于满足下述的仿真判定条件。

ARTC 仿真判定是在既不出现抢占也不允许阻塞的情况下消息集 M 使用 Sr 调度算法的可调度性。由于使用 Sr 调度算法，将 $D=\{D_1,D_2,\cdots,D_n\}$ 用 $\{r\}$ 进行描述，转换成 $D'=\{b_1,b_2,\cdots,b_n\}$，具有如下性质：$b_j|b_i(j\geqslant i)$。于是，转化后的消息集 M′，其距离约束的最小公倍数等于 b_n，由此我们只需判定消息级 M′在时长 b_n 内的可调度性。为了保证消息执行既不中断也不被阻塞，我们采用下面的方法进行判定：首先将 b_n 分为 k 段，每段长为 b_1，接着依次对当前消息结合中的最高优先级消息进行调度判断。如果能在当前时段内执行完毕，则对该消息执行调度，否则移至下一时段；如果在 b_n 时长内所有的消息均能满足其距离约束，完成调度，则说明消息级 M 可用进行 Sr 调度，且消息执行无中断、不阻塞。这一判定过程被称为可调度性仿真判定。

由于消息集 M 经调和数转换后，所有消息的距离限制都小于、等于该消息的原有死线(周期)，因此由定理 3 得：若 M 经仿真判定可由 Sr 调度，则其也必定能在无中断、不阻塞的情况下被 RM 调度。另一方面，如果消息集 M 在基础判定或仿真判定时无法调度，系统将采取相应措施(如将本网段分为两个网段)处理。

(3)ARTC *在线调度算法*　ARTC 按照 RM 调度算法进行调度决策，并且保证消息不出现中断和阻塞。其过程如下：

①确定集合 M 中尚待传输的最高优先级消息 M_i。

②用 d_j 表示对于当前时间来说消息 j 到达下一周期的剩余时间。如果消息 M_i 的执行时间 C_i 小于或等于 $\min\{d_j\}$ $(1\leqslant j<i)$(即在接下来的时间 C_i 内没有优先级比 M_i 高的消息就绪)，则分配时间 C_i 给该消息，否则分配 $\min\{d_j\}$ $(1\leqslant j<i)$ 时长给非实时模式。

通过上述两步的不断循环执行，MS 就可自动生成总线调度表。当收到新的消息请求，MS 将重新排列消息集 M，并进行可调度性判定。如果满足条件，则将其插入消息集合，参与调度，否则拒绝该请求。

(4)*非周期消息处理和非实时模式应用*　系统在初始化期间为非周期消息预留下一定的总线带宽，并采用周期服务器方式进行询问。MS 将根据非周期消息源节点的特性分别设置询问周期，BT(s)应答时对产生的非周期数据进行相应描述，BT(s)也可在当前消息的应答帧中捎带非周期数据传输请求。调度策略将根据该非周期消息的执行时间和死线判断其可接受性，然后进行调度或拒绝该请求。

非周期消息的可用时间包括将其转化为周期消息所拥有的执行时间和非实时模式执行时间，即仅当既无周期消息又无非周期消息传输时，总线才可为非实时模式所使用。

上述机制在 MS 的 ARTCP 层运行，MS 按照自动生成总线调度表逐条执行调度，保证了所有消息 M_i 在其周期 D_i 内都获得 C_i 长度的执行时间。对于上层应用来说，由于消息的执行时间确定并受 MS 调度机制的控制，此外 ARTCS 以及上层处理开销对于既定的系统可以通过测试得到，所以对上层特定业务来说最坏情况下的传输响应时间是确定的。由此可见，利用 ARTC 技术可以建立一个完全意义上的以太网硬实时应用系统。

总结　ARTC 硬实时以太网技术在我室研究成果 RTCC 的基础上进行了下列改进、扩展和创新，形成了完整的体系结构并支持实际应用：

①扩展了网络节点的在线管理机制，完善了主控站功能，支持主控站的竞争和移交，支持基本终端在线加入和退出，增强了网络的灵活性和健壮性。

②完善了 ARTC 体系结构，在 ARTC 协议层之上增加了 ARTC 服务层，建立了健全的 CMIB 管理体制，能更好支持上层业务要求及网络管理。

③改进了命令帧、数据帧和应答帧的格式，建立了地址分配和管理体制，增强了网络的可靠性和灵活性，为管理跨网段业务奠定了基础。

④创造性地提出了一种支持总线表自动生成的 ARTC 在线硬实时调度机制，该机制既支持周期消息又支持非周期消息，并保证消息在传输时不中断和阻塞，能较好地满足硬实时通信实际应用需要。

⑤ARTC 在线调度机制具有较高的总线利用率，易于移置到其它硬实时通信系统中。该机制不但可以自动生成静态总线调度表，还可用于系统初始化和变化时总线调度表的在线生成和重排，开销较小、可实施性好。

然而，如何满足应用需求，提供高效、灵活、可靠的实时网络通信却是一项长期而艰巨的任务，ARTC 技术仍需不断地修正和完善。我们将致力于不断提升该技术的可靠性和可扩展性，更好地支持实际应用。

参考文献

1 Wang Zhiping, Xiong Guangze, Zhou Wanlei. Design and implementation of a hard real-time communication protocol. World Computer Congress 2000, Beijing, China, Aug. 2000

2 Malcolm N, Zhao W. Hard real-time communication in multiple-access networks. Journal of Real-Time Systems, 1995, 8(1): 35～77

3 Tanenbaum A S. Computer Networks. Prentice Hall, 1996

4 Venkatramani C, Chiueh T. Design, implementation and evaluation of a software-driven real-time Ethernet protocol. In: Proc. SIGCOMM Conf. of ACM, 1995. 27～37

5 Liu C L, Layland J W. Scheduling Algorithm for Multiprogramming in a Hard-Real-Time Environment. Journal of the ACM, 1973, 20(1): 40～61

6 Sha L, Sathaye S. A Systematic Approach to Designing Distributed Real-Time Systems. IEEE Computer, 1993, 26(9): 68～78

维普资讯 http://www.cqvip.com

无线局域网 Sniff 技术研究

卜勇华　胡爱群　王兴建
（东南大学无线电系　东南大学信息安全研究中心　南京 210096）

Sniff Technology of Wireless LAN

BU Yong-Hua　HU Ai-Qun　WANG Xiang-Jian
(Research Center of Information Security, Southeast University, Nanjing, 210096 China)

Abstract　This paper researches the sniff technology of 802.11b wireless LAN. It describes the channel, topology, authentication and privacy of WLAN, points out the way to capture wireless data, then presents the analyse procedure of wireless frames. At last an actual frame is given.

Keywords　Wireless LAN(WLAN), IEEE802.11, Sniff

1. 引言

无线局域网(WLAN)为人们提供了一种便捷的网络接入方式，北京大运会上无线局域网的综合成功运用，使人们充分地认识到了无线局域网的高效和快捷。无线局域网可以和现有有线网实现无缝连接，本身更具有布局的简单性、灵活性，可以在移动中实现通信。美国 IDC 公布了全球有关无线 LAN LSI 市场的调查结果。调查表明，该市场在 2000 年～2005 年期间将以年平均 29%的增长率扩大。[1]

另一方面，无线局域网通过无线电波发送数据，无线节点间通过 CSMA/CA 机制共享一个频段，无线局域网就相当于一个大的 HUB 系统，在一个局部区域内的无线节点都能收到无线通信数据，因此无线局域网的监听比有线网络更为容易。而且由于处于 Sniff 机制下的无线节点无法发送数据，完全处于被动监听状态，对 Sniff 的探测也更为困难。

2. 802.11 WLAN 相关技术

1997 年，IEEE 确定 802.11 作为第一个国际认可的无线局域网标准。1999 年 9 月发布了 802.11b 标准。802.11b 标准描述了 OSI 模型的最底层(physical)以及一部分数据链路层(data link)协议。

WLAN 和我们熟悉的有线网如以太网的最大区别在于传送数据的介质不同。WLAN 信息的携带依赖于对无线电波的调制。802.11b 工作在 2.4GHz 频段。美国 2.4GHz 频段(2.40GHz 到 2.45GHz)的实现，分为 11 个可用的信道。为了限制干扰，实际使用小于信道带宽的一半。所有网络硬件设计为可以在任何一个信道上实现监听和发送，但为了直接通信，发送者和接收者必须在同一个信道上。对于 Sniff 来说，可以先进行信道扫描，确定存在通信的信道，然后固定在一个信道上进行监听。

在 IEEE802.11 中定义了几种常用的拓扑结构：基本服务组 BSS (Basic Service Set)；分布式系统服务 DSS (Distribution System Services)；扩展服务组 ESS (Extended Service Set)。

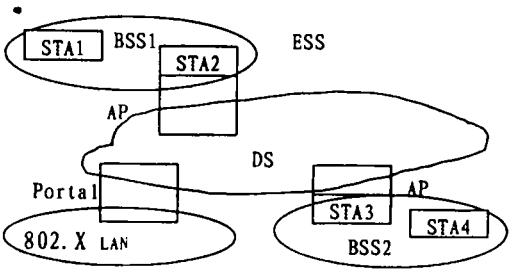

图 1　无线局域网拓扑图

基本服务组 BSS 是组成一个 802.11 局域网的最基本的单元。如图 1 中 BSS1 所示。一个基本服务组中可包含若干个站点 STA (Station)。在一个 BSS 中的站点之间可以直接通信。一个单独的 BSS 组成的网络(IBSS)是 IEEE802.11 中最基本的类型，它没有中枢链路基础结构，最少包含两台站点。一个基本服务组所能覆盖的区域受到环境和站点收发机特性

卜勇华　硕士，从事无线通信、网络安全的研究。胡爱群　博士，教授，博士生导师。王兴建　硕士，从事无线通信、网络安全的研究。

7 Tindell K W, Burns A, Wellings A J. Analysis of Hard Real-Time Control Applications. In: Proc. Real-Time Technology and Applications Symposium, May 1995. 240～249
8 Sprunt B, Sha L, Lehoczky J. Aperiodic task scheduling for hard real-time systems. The Journal of Real-Time Systems, 1989, 1: 27～60
9 Sha L, Rajkumar R, Lehoczky J P. Priority inheritance protocols: An approach to real-time synchronization. IEEE Transactions on computers, 1990, 39(9): 1175～1185
10 Han C C, Lin K J. Scheduling distance-constrained real-time tasks. In: Proc. IEEE Real-Time Systems Symposium, 1992. 300～308
11 Hsueh C, Lin K J, Fan N. Distributed pinwheel scheduling with end-to-end timing constraints. In: Proc. of IEEE Real-Time Systems Symposium, 1995. 171～181
12 Lin K-J, Herkert A. Jitter Control in Time-Triggered Systems. In: Proc. of the 29th IEEE annual Hawaii intl. conf. on system sciences, 1996
13 Strosnider J K, Lehoczky J P, Lui Sha. The deferrable server algorithm for enhanced aperiodic responsiveness in hard real-time environments, Computers, IEEE Transactions on, 1995, 41(1): 73～91
14 Mok A K. Fundamental design problems of distributed systems for hard real time environment: [Ph. D Thesis]. M. I. T., 1983
15 王志平，熊光泽，刘锦德. 1553B 实时网络流控问题研究. 计算机科学，1999, 26(7): 80～82