

基于 CAN 总线的智能网桥设计

王黎明¹, 王明哲², 闫晓玲¹

(1. 海军工程大学电气与信息学院, 武汉 430033; 2. 华中科技大学控制科学与工程系, 武汉 430011)

摘要: 根据 CAN 总线扩展标识符的特点和网桥地址需求, 利用扩展标识符构建数据转发地址表, 针对网段的合理分配问题, 提出网段优先和节点优先的网段节点位分配算法, 并进行位利用率的评估。分析二维数组转发地址表的学习算法, 针对网桥环网提出单向生成转发路线的网桥端口生成树算法。在不同负载下对网桥的利用率和延迟进行实验, 结果证明该 CAN 双端口网桥设计方法可以实现 CAN 网桥的功能。

关键词: CAN 网桥; 扩展标识符; 网段优先; 节点优先; 位利用率; 生成树算法

Design of Intelligent Bridge Based on CAN Bus

WANG Li-ming¹, WANG Ming-zhe², YAN Xiao-ling¹

(1. College of Electric and Information, Naval Engineering University, Wuhan 430033;

2. Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan 430011)

【Abstract】 According to characteristics of CAN bus extended identifier and requirements of bridge address, this paper constructs transmitting data address form by using extended identifier. Aiming at available net-segment distribution, net-segment and node bit distribution algorithm with net-segment priority and node priority are established, and rate of bit utilization is evaluated. Study algorithm of two dimensions array transmitting address table is analyzed, and building tree algorithm of transmitting single orientation on bridge ports is proposed. Experiments on utilization and delay of bridge under different load and transmission validate prove that the design method of dual-port CAN bridge can implement the functions of CAN bridge.

【Key words】 CAN bridge; extended identifier; net-segment priority; node priority; rate of bit utilization; building tree algorithm

1 概述

网络的复杂化及网络内节点数目的增多导致了总线负荷过重, 低优先级节点实时性得不到保障。当前最有效的方法是将整个网络划分为多个较小的网络, 让通信量频繁的节点在同一个小网络内, 保证它们之间通信的实时性, 而将与它们通信频率较低的节点与它们隔离开, 从而避免由于某一节点发送信息导致所有其他节点不能及时发送信息。这样当各个不同速率的网段相连时, 就需要网桥设备^[1]。

网桥是具有多个网络端口的数据收发设备, 其基本功能是利用媒质访问协议从一个端口接收局部网络的数据帧并向另一端口的局部网络发送数据帧。网桥在局域网之间转发并过滤。它允许来自一个网络的节点的帧被转发到另一个网络上的某个节点, 但丢弃所有源网络和目的网络相同的帧。因此, 网桥不转发本地信息, 只转发目的为远程的网络流量, 并且工作在链路层, 可以通过检查网络接口卡的硬件地址决定是转发还是丢弃该帧。

双端口 CAN 网桥的设计不同于以太网网桥的设计, 其自身有很多特点, 网桥对基于 CAN 协议的帧格式数据进行转发, CAN 网桥应有足够的数据缓冲空间以保证数据通信的完整性, 同时, 由网桥处的高速处理性能保证通信的实时性; 由于 CAN 总线采用多主方式工作, 在其帧信息中无站地址信息, 因此应根据 CAN 数据帧的结构特点, 采用一些技术实现一般 MAC 透明网桥的功能。要完成透明网桥必须实现数据链路层源地址和目的地址的构建、网段和节点的合理分配以及环网的自动消除^[2-3]。

2 数据转发地址表的构造

对于多个子网的互联, 可以利用扩展标识符定义地址的方法生成数据转发地址表。这种方法是将标识符域按应用系统的特点进行划分, 定义目标地址和源地址, 采用类似于以太网 MAC 网桥设计的方法设计 CAN 网桥。将 11 位基本标识符的前 4 位定义为优先级, 其余 7 位表示类型; 将 18 位扩展标识符域分为两大部分, 分别表示目的地址 Destination Address 和源地址 Source Address, 如图 1 所示。优先级用于数据发送时的总线冲突仲裁; 类型可用于传输数据的类型, 如命令、状态、参数、信号量, 也可以用于表示某一节点数据包中的字节序号; 目标地址代表了接收数据的网络节点; 源地址代表了发送数据的网络节点。因此, 整个网络最大节点数为 512, 数据类型最多为 256。

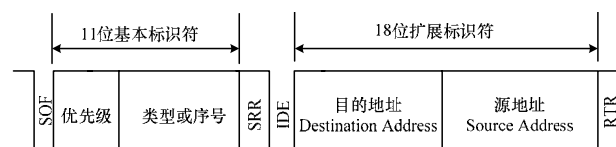


图1 数据转发地址表结构

基金项目: 国家自然科学基金委创新研究群体基金资助项目 (50421703); 国防科技重点实验室基金资助项目

作者简介: 王黎明(1978-), 男, 博士研究生, 主研方向: 模式识别, 分布式控制, 人工智能, 图像处理, 着色 petri 网; 王明哲, 教授、博士生导师; 闫晓玲, 工程师

收稿日期: 2009-04-11 **E-mail:** icesoar@163.com

根据扩展标识符结构,其目标地址与源地址都由9个仲裁位组成,某网段内最多可以挂接512个节点。分析可知,这种构造数据转发地址的方法可以在完全不改变CAN总线特性的情况下,实现CAN网桥的目的地址和源地址,同时网段最多可挂接512个节点,这完全可以满足CAN总线节点的需求。很多研究人员对子网和网段按照节点速率的原则进行分配,但是在节点数较多时,按速率原则分配容易造成节点利用率下降和驱动能力不足的问题。

3 网段和节点数分配算法

本文提出网段和节点数分配算法。假定源地址和目的地址的位数相等: $B_{SA} = B_{DA} = 18/2$ (在整个网络中,“源地址和目的地址是对称分布”仅仅是相对概念,因此,应取的位数相等)。为了保证在各网段内尽可能多地接入节点,满足多网段情况下多个网桥同时工作的情形,同时为了表示子网的网段号位数以及子网内节点位数,定义变量:子网段网号位数 B_{Net_mark} 和子网内节点位数 $B_{Net_in_node}$, 且

$$B_{Net_mark} + B_{Net_in_node} = B_{SA} = B_{DA} = 9 \quad (1)$$

那么其表示的网段数目为

$$N_{Net_mark} = 2^{B_{Net_mark}} \quad (2)$$

网络中使用的双端口网桥数量最多为 $N_{Net_mark} - 1$, 表示的子网内节点数为

$$N_{Net_in_node} = 2^{B_{Net_in_node}} \quad (3)$$

从式(3)可以看出,子网网段数(或双端口网桥数)和子网内节点数是相互制约的,增大子网网段数的同时,必然减少同一网段内节点的数量。在一般应用中,首先确定是否满足最大节点数,其次确定总线单子网络的最大节点数量,然后根据最大节点数量灵活地选择子网内节点数 $N_{Net_in_node}$ 和网段数目 N_{Net_mark} , 从而确定子网内节点位数 $B_{Net_in_node}$ 和子网段网号位数 B_{Net_mark} 。

CAN总线上的节点数以及子网节点数主要取决于总线驱动电路,如图2所示,即主要取决于CAN驱动器所能驱动的最小负载电阻 $R_{L,min}$ 。如CAN驱动器PCA82C250提供的负载驱动能力为 $R_{L,min} = 45 \Omega$ 。

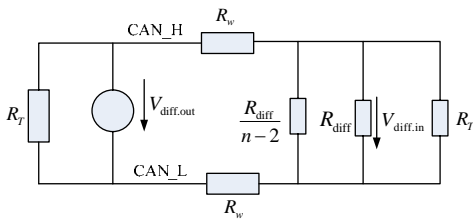


图2 CAN总线节点驱动电路模型

最大节点数满足如下关系(假设总线电阻 R_w 为0,此时为最坏情况):

$$\frac{R_{T,min} \times R_{diff,min}}{(n_{max} - 1) \times R_{T,min} + 2R_{diff,min}} > R_{L,min} \quad (4)$$

从而计算得最大节点数 n_{max} 为

$$n_{max} < 1 + R_{diff,min} \times \left(\frac{1}{R_{L,min}} - \frac{2}{R_{T,min}} \right) \quad (5)$$

假设PCA82C250的最小差分输入电阻 $R_{diff,min} = 20 \text{ k}\Omega$, 当 $R_T = 120 \Omega$ 和 $R_L = 45 \Omega$ 时,能连接的最大节点数为112。实验证明,除了收发器的驱动能力外,节点数量还与总线长

度有密切关系。为此,本文提出2种网段节点位分配算法:网段优先法和节点优先法。

(1)网段优先法:取单子网内最大节点数满足

$$N_{Net_in_node} = 2^{B_{Net_in_node}} < n_{max}$$

即单子网节点数小于最大节点数。该方法可提高位利用率 η 。

(2)节点优先法:取子网内节点位数满足

$$B_{Net_in_node} = \text{lb}(n_{max} + z), \quad z > 0$$

这种方法以消耗位为代价,使单子网内节点数连接最大,即降低位利用率。

设单子网最多可以挂接110个节点,按照上述2种方法,分别设定如表1和表2所示的各源地址和目标地址,其中,网段优先法取 $B_{Net_in_node} = 6$; 节点优先法取 $B_{Net_in_node} = 7$ 。

表1 网段优先法各节点源地址、目标地址的配置

目的地址		源地址	
子网段	对应网段节点地址编号	子网段	对应网段节点地址编号
000:子网0	000000-111111	000:子网0	000000-111111
001:子网1	000000-111111	001:子网1	000000-111111
010:子网2	000000-111111	010:子网2	000000-111111
011:子网3	000000-111111	011:子网3	000000-111111
100:子网4	000000-111111	100:子网4	000000-111111
101:子网5	000000-111111	101:子网5	000000-111111
110:子网6	000000-111111	110:子网6	000000-111111
111:子网7	000000-111111	111:子网7	000000-111111

表2 节点优先法各节点源地址、目标地址的配置

目的地址		源地址	
子网段	对应网段节点地址编号	子网段	对应网段节点地址编号
000:子网0	0000000-1101110	000:子网0	0000000-1101110
001:子网1	0000000-1101110	001:子网1	0000000-1101110
010:子网2	0000000-1101110	010:子网2	0000000-1101110
011:子网3	0000000-1101110	011:子网3	0000000-1101110

分析可知,网段优先法整个网络节点数最多为 $N_{max1} = 2^3 \times 2^6 = 512$, 而节点优先法整个网络节点数最多为 $N_{max2} = 2^2 \times 110 = 440$, 网络节点位利用率 η 为

$$\eta_1 = \frac{N_{max1}}{N_{max}} = \frac{512}{512} = 100\% \quad (6)$$

$$\eta_2 = \frac{N_{max2}}{N_{max}} = \frac{440}{512} = 86\% \quad (7)$$

因此,节点优先法的网络节点位利用率低于网段优先法。本文提出的2种方法为网段和节点的分配提供了2个边界,这样在设计节点和网段分配时,可以按照相关的方法进行定义。

4 转发地址表的自学习算法

某端口的转发地址表存储在该端口的RAM中,该端口的微控制器将所接收报文的地址信息与转发地址表中的地址信息进行比较,从而决定转发策略。

当任一节点上电完成初始化工作后,首先向CAN总线发送报文,其标识符的源地址作为节点编号,源地址包含两部分信息:网段数和单段节点数,此时取目的地址低8位为0。网桥的CAN接口收到此报文后,将源地址取出,并读取实时时钟内的当前时间,将这些信息以一条记录存储下来,保存于转发地址表中,这样转发地址表中将保存所有已上电工作的本网段内节点的地址。同样当一个节点工作完成、退出工作前,也会向网桥接口发送离线信息,在转发地址表中删除该节点的地址信息^[4]。

转发地址表由一个 $[N_{Net_in_node}, 4]$ 的二维数组组成，如图 3 所示，数组每一行代表一个网段地址信息的记录，其中，每一项信息代表一个节点的地址信息记录，每项记录由 4 个字节组成：第 1 字节用来存储发送节点的 $B_{Net_in_node}$ 位源地址编码；第 2 字节~第 4 字节用来更新时间信息，单位精确到毫秒级。

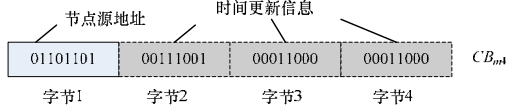


图 3 地址转发表项信息组成

转发地址表为

$$\begin{bmatrix} CB_{11} & CB_{12} & \dots & CB_{14} \\ CB_{21} & CB_{22} & \dots & CB_{24} \\ \vdots & \vdots & & \vdots \\ CB_{m1} & CB_{m2} & \dots & CB_{m4} \end{bmatrix}$$

其中， $m = N_{Net_in_node}$ 。

转发地址表在自学习和地址更新的同时，当一个节点意外地连接到一个以上的网桥上时就会形成一个活动环路。每次该节点向另一网段内的节点发送数据帧时，每个网桥都要发送一个该帧的样本，这将导致 2 个相同的帧在网络中传输，从而产生网络环路，网络环路可能为桥接和网络带来严重的问题，因为环路不必要地复制了数据帧。这些多余的数据流量可能会严重降低网络的性能。

5 网桥端口生成树算法

使用生成树算法可避免由于网络环路所造成的不必要的数据帧复制。生成树是一系列经由网络的、设备到设备的路径，这样网络中任何 2 个设备之间就有且只有一条路径。生成树算法通过网桥和网桥之间的一系列协商构建一棵生成树。这些协商决定哪些路径可以用于传输，哪些路径至少是暂时不可用的。这些协商的结果是每个网桥都有一个端口被置于转发状态，其他端口则被置于阻塞状态。该过程将保证网络中任何 2 个设备之间只有一个通路，并可以防止出现任何形式的网络环路^[5-6]。双端口网桥的连接方法如图 4 所示。

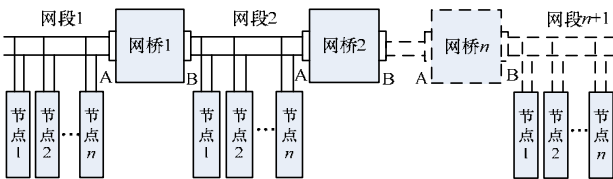


图 4 双端口网桥的连接方法

假定网段数量为 N_{Net_mark} ，则双端口网桥数量为 $N_{Net_mark} - 1$ ，子网内节点数为 $N_{Net_in_node}$ 。

消除环网的思想即网络中任何设备只有一条路径。单向生成转发路线的方法即以网桥一端向相反方向单向转发的方法进行分析，如图 5 所示。

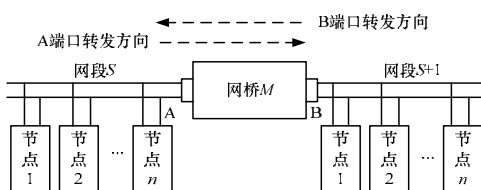


图 5 双端口网桥数据转发方向

(1) 网桥 A 端口转发路线

对于网桥 1 的 A 端口，当目的地址中的网段号 S 满足 $N_{Net_mark} > S > 1$ 时，该网桥 A 端口前向转发；当目的地址中网段号 $S = 1$ 时，丢弃该帧。

对于网桥 2 的 A 端口，当目的地址中的网段号 S 满足 $N_{Net_mark} > S > 2$ 时，该网桥 A 端口前向转发；当目的地址中网段号 $S = 2$ 时，丢弃该帧。

考虑一般情况，对于网桥 M 的 A 端口，当地址中的网段号 S 满足 $N_{Net_mark} > S > M$ ，其中， M 满足 $M \leq N_{Net_mark} - 1$ 时，该网桥 A 端口前向转发；当目的地址中网段号 $S = M$ 时，丢弃该帧。

(2) 网桥 B 端口转发路线

对于网桥 1 的 B 端口，当目的地址中的网段号 S 满足 $S = 1$ 时，该网桥 A 端口后向转发；当目的地址中网段号为 $N_{Net_mark} > S > 1$ 时，丢弃该帧。

对于网桥 2 的 B 端口，当目的地址中的网段号 $S = 2$ 时，该网桥 A 端口后向转发；当目的地址中网段号 S 满足 $N_{Net_mark} > S > 2$ 时，丢弃该帧。

考虑一般情况，对于网桥 M 的 B 端口，当地址中的网段号 $S = M$ 时，该网桥 B 端口前后转发；当目的地址中网段号 S 满足 $N_{Net_mark} > S > M$ ，其中， M 满足 $M \leq N_{Net_mark} - 1$ 时，丢弃该帧。

6 实验

本文的网桥网段节点选取方法、地址表生成方法以及消除环路的生成树法已通过仿真验证，使用的仿真工具为 CANoe5.1。同时通过实践验证，验证方法为：将网桥用于某发电厂原动机参数、电站参数的采集测量中，整个发电厂共有 4 台机组，因此，考虑到整体的布置，整个系统采用了 4 个网段，每个网段有采集点 88 个。

为了获得网桥的各项技术指标，从而验证网桥设计的有效性以及本文提出的节点分配方法、地址表生成、消除环路方法的有效性，系统采用了 3 种本地/远程的报文比率。

图 6 评估了设计的网桥的利用率，主要为网桥接口的利用率，从图中可以看出，当总线负载率超过每秒 5 000 帧时，网桥的 CAN 总线接口利用率小于 50%，说明设计的网桥节点分布较为合理，与单子网节点的 CAN 网络执行效能相当。

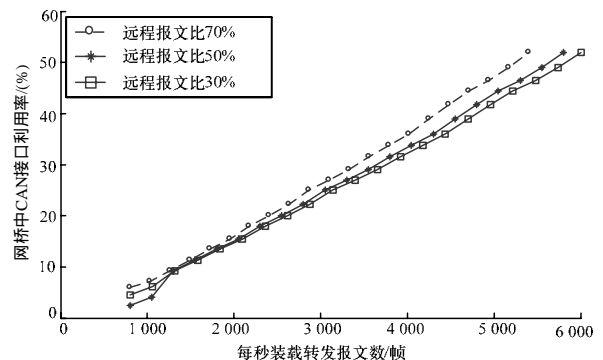


图 6 不同远程报文比情况下网桥的利用率

图 7 显示在不同远程报文比的情况下，网桥内核的利用率情况。可以看出，随着远程报文比的增大，网桥内核的利用率几乎呈线性增长，而不是无限地增大。同时当报文发送量较大时，网桥内核的利用率保持在小于 45%。这从另一方面证明了本文设计的网桥实现了地址表的自动生成，并消除

了网络环路。

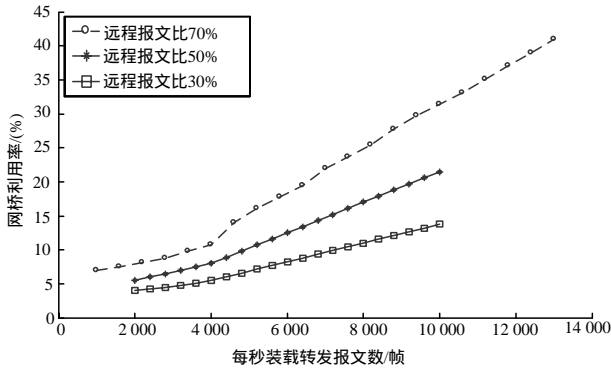


图7 不同远程报文比情况下网桥内核的利用率

图8给出网桥端口在不同负载和不同远程报文比情况下的网桥平均处理时间。图9为重传帧的CAN接口延时测试结果。随着远程报文的增加，网桥处理时间在增加，同时随着负载的增加，网桥的处理时间也在增加。处理器时间是1.5倍的发送和接收时间，但是对于CAN网络来说，这样的延迟时间是完全可以接受的。因此，本文设计的网桥满足CAN总线网桥的要求。

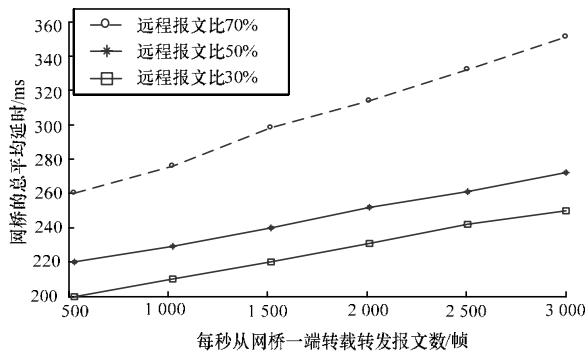


图8 不同负载和远程报文比的情况下网桥延迟时间

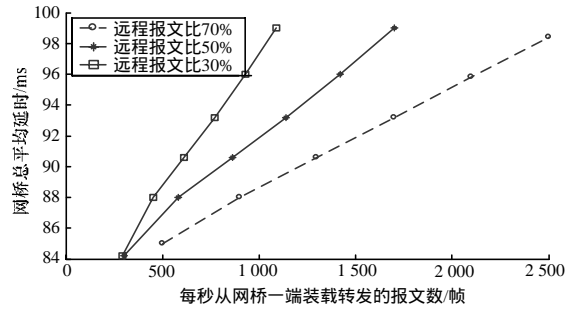


图9 重传帧的CAN接口延时

7 结束语

本文提出一种智能网桥构建的优化算法，包括数据转发地址表、网段和节点数分配算法、转发地址表的自学习算法以及网桥端口生成树算法。实验验证该智能网桥能够满足CAN网桥的要求，有效地降低网络负载率。

参考文献

- [1] 谭杰. CAN总线分布式测控系统中网桥的研究与设计[D]. 武汉: 武汉理工大学, 2006.
- [2] 曹小华, 赵成, 陶德馨. 基于现场总线CAN的实时监控系统设计[J]. 武汉理工大学学报, 2005, 29(6): 910-912.
- [3] 贺毅, 赵望达, 刘勇求. 现场总线技术应用及其发展趋势探讨[J]. 工业计量, 2005, 15(1): 21-23.
- [4] Ekiz H, Powner E T, Ruth A. Design and Implementation of a CAN/CAN Bridge[C]//Proceedings of IEEE ISPAN'96. Beijing, China: IEEE Press, 2006.
- [5] Ekiz H, Powner E T, Kutlu A. Design and Implementation of a CAN/Ethernet Bridge[C]//Proc. of the 3rd International Conference on CAN. Paris, France: [s. n.], 2006.
- [6] Have B, Kirby A. Transparent Interconnection of Local Area Network with Bridges[J]. J. of Telecommunication Network, 2004, 3(2): 116-130.

编辑 张帆

(上接第40页)

参考文献

- [1] Andrews T, Curbera F, Dholakia H, et al. Business Process Execution Language for Web Services Version 1.1. Specification[Z]. (2003-05-01). [http://msdn.microsoft.com/en-us/library/ee251596\(BTS.10\).aspx](http://msdn.microsoft.com/en-us/library/ee251596(BTS.10).aspx).
- [2] Kavantzias N. Web Services Choreography Description Language Version 1.0.W3C[Z]. W3C Working Draft, 2004.
- [3] Nakajima S. Model-checking Verification for Reliable Web Service[C]//Proc. of OOPSLA Workshop on Object-oriented Web Services. Washington D. C., USA: [s. n.], 2002.

- [4] Koshkina M, Breugel F. Modelling and Verifying Web Service Orchestration by Means of the Concurrency Workbench[J]. ACM SIGSOFT Software Engineering Notes, 2004, 29(5): 1-10.
- [5] Hamadi R, Benatallah B. A Petri Net-based Model for Web Service Composition[C]//Proc. of the 14th Australasian Database Conf. on Database Technologies. Adelaide, Australia: [s. n.], 2003.
- [6] Jensen K. Colored Petri Nets: Basic Concepts, Analysis Methods and Practical Use[M]. Berlin, Germany: Springer-Verlag, 1997.
- [7] CPN Tools[Z]. (2005-01-01). <http://www.daimi.au.dk/CPNtools>.

编辑 顾姣健