(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 109617705 A
(43)申请公布日 2019.04.12

(21)申请号 201811170304.7

(22)申请日 2018.10.08

(30)优先权数据
17194686.6 2017.10.04 EP

(71)申请人 TTTECH 电脑技术股份公司
地址 奥地利维也纳

(72)发明人 S·克拉西纳斯 R·塞尔纳奥利弗

(74)专利代理机构 上海翼胜专利商标事务所（普通合伙）31218
代理人 翟羽

(51)Int.Cl.
H04L 12/24(2006.01)
H04L 29/08(2006.01)

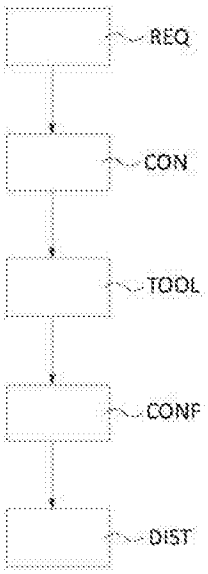
权利要求书3页 说明书8页 附图4页

(54)发明名称

用于配置实时网络的方法和设备

(57)摘要

本发明涉及一种在实时网络中配置节点的方法，节点利用链路通过交换帧互相通信，其中至少一个节点连接到至少一个队列，并且将帧放入其中，以在与该队列关联的链路上传输，该队列与能够处于打开状态或关闭状态的门电路关联。如果门电路处于打开状态，则节点从被放入链路上的队列中的帧选择以进行传输；当该门电路处于关闭状态时，节点不从该链路上的队列中选择帧进行传输。门电路随着时间从打开变为关闭和从关闭变为打开，从而形成窗口，如配置数据所示。配置数据由能够解算阵列理论中的约束的工具产生。工具接受输入，输入公式化为阵列理论中的约束。该配置数据传送到实时网络中的节点，节点将至少部分配置数据用作本地配置。



CN 109617705 A

1.一种在实时网络中配置节点(v1、.....、v8)的方法,所述实时网络的节点(v1、.....、v8)利用链路(E1a、.....、E8a、E1b、.....、E8b)直接地或者通过其他节点(v1、.....、v8)间接地互连,并且利用所述链路(E1a、.....、E8a、E1b、.....、E8b)通过交换帧(F1、.....、F4)互相通信,

其中至少一个节点(v1、.....、v8)使至少一个队列(Q1、.....、Q3)与至少一个链路(E1a、.....、E8a、E1b、.....、E8b)关联,其中所述至少一个节点(v1、.....、v8)连接到所述至少一个队列(Q1、.....、Q3),并且

其中所述至少一个节点(v1、.....、v8)将帧(F1、.....、F4)放入所述至少一个队列(Q1、.....、Q3)中,以在所述关联链路(E1a、.....、E8a、E1b、.....、E8b)上传输,

并且其中所述至少一个队列与能够处于打开状态或关闭状态的门电路(G1,...,G3)关联,

并且其中如果与所述队列关联的所述门电路的状态处于打开状态,则所述至少一个节点从被放入在与所述队列关联的至少一个链路上的所述至少一个队列中的帧选择(TS)以进行传输,并且当所述门电路处于关闭状态时,不从所述链路上的队列中选择帧进行传输;

并且其中所述门电路随着时间的推移使所述状态从打开变更为关闭和从关闭变更为打开,从而形成窗口(W1、.....、W4),如配置数据(GCL)所示,

其特征在于,

所述配置数据(GCL)由工具产生,其中所述工具能够解算阵列理论中的约束,并且其中

所述工具接受输入,所述输入公式化为阵列理论中的约束(CON),并且其中

利用人手用户的输入和/或来自所述实时网络中的一个、两个、或多个节点的通信请求,所述输入提供给所述工具,并且其中

所述配置数据(GCL)传送到所述实时网络中的一个、两个或多个节点(v1、.....、v8),节点或各节点将至少部分所述配置数据(GCL)或所述配置数据(GCL)用作本地配置。

2.根据权利要求1所述的方法,其特征在于,阵列理论中的一些所述约束是格式正确的窗口约束(CON-WELLFORMED)和/或队列分配约束(CON-QUEUEASS)和/或串流实例约束(CON-STREAMINST)和/或有序窗口约束(CON-ORDWIND1、CON-ORDWIND2)和/或帧-窗口分配约束(CON-FRTOWIND)和/或窗口大小约束(CON-WINDSIZE1、CON-WINDSIZE2)和/或串流约束(CON-STREAM)和/或串流隔离约束(CON-STREAMISOL)。

3.根据权利要求1或2所述的方法,其特征在于,使用SMT解算器作为所述工具,能够解算阵列理论中的表达式。

4.根据权利要求1至3中的任何一项所述的方法,其特征在于,OPC/UA和/或DDS和/或NETCONF/YANG和/或SNMP和/或FTP用于将所述通信要求传送到所述工具并且/或者将配置数据(GCL)传送到所述节点(v1、.....、v8)。

5.根据权利要求1至4中的任何一项所述的方法,其特征在于,所述实时网络的每个节点是由以太网交换机、路由器和端站组成的组中选出。

6.根据权利要求1至5中的任何一项所述的方法,其特征在于,所述状态从打开变更为关闭和从关闭变更为打开是根据IEEE 802.1Qbv标准执行的。

7.根据权利要求1至6中的任何一项所述的方法,其特征在于,优选地利用IEEE 1588标准和/或IEEE 802.1AS标准和/或SAE AS6802标准或所述标准的任一版本,在所述节点中建

立同步时间。

8.一种在实时网络中配置节点(v1、.....、v8)的设备(CNC),所述实时网络的节点(v1、.....、v8)利用链路(E1a、.....、E8a、E1b、.....、E8b)直接地或者通过其他节点(v1、.....、v8)间接地互连,并且利用所述链路(E1a、.....、E8a、E1b、.....、E8b)通过交换帧(F1、.....、F4)互相通信,

其中至少一个节点(v1、.....、v8)使至少一个队列(Q1、.....、Q3)与至少一个链路(E1a、.....、E8a、E1b、.....、E8b)关联,其中所述至少一个节点(v1、.....、v8)连接到所述至少一个节点(v1、.....、v8),并且

其中所述至少一个节点(v1、.....、v8)将帧(F1、.....、F4)放入所述至少一个队列(Q1、.....、Q3)中,以在所述关联链路(E1a、.....、E8a、E1b、.....、E8b)上传输,

并且其中所述至少一个队列与能够处于打开状态或关闭状态的门电路(G1,...,G3)关联,

并且如果与所述队列关联的所述门电路的状态处于打开状态,则所述至少一个节点仅可以从被放入在与所述队列关联的链路上的所述至少一个队列中的帧选择(TS)以进行传输,并且当所述门电路处于关闭状态时,不从所述链路上的队列中选择帧进行传输;

并且其中所述门电路随着时间的推移使所述状态从打开变更为关闭和从关闭变更为打开,从而形成窗口(W1、.....、W4),如配置数据(GCL)所示,

其特征在于,

所述设备通过执行工具能够产生配置数据(GCL),其中所述工具能够解算阵列理论中的约束,并且其中

所述工具配置成接受输入,所述输入公式化为阵列理论中的约束(CON),其中

利用人手用户的输入和/或来自所述实时网络中的一个、两个、或多个节点的通信请求,将所述输入提供给所述工具,并且其中所述设备配置成将所述配置数据(GCL)传送到所述实时网络中的一个、两个或多个节点(v1、.....、v8),所述一个、两个或多个节点(v1、.....、v8)将至少部分所述配置数据(GCL)用作本地配置。

9.根据权利要求8所述的设备,其特征在于,阵列理论中的一些所述约束是格式正确的窗口约束(CON-WELLFORMED)和/或队列分配约束(CON-QUEUEASS)和/或串流实例约束(CON-STREAMINST)和/或有序窗口约束(CON-ORDWIND1、CON-ORDWIND2)和/或帧-窗口分配约束(CON-FRTOWIND)和/或窗口大小约束(CON-WINDSIZE1、CON-WINDSIZE2)和/或串流约束(CON-STREAM)和/或串流隔离约束(CON-STREAMISOL)。

10.根据权利要求8或9所述的设备,其特征在于,使用SMT解算器作为所述工具,能够解算阵列理论中的表达式。

11.一种包括节点(v1、.....、v8)的实时网络,所述实时网络的节点(v1、.....、v8)利用链路(E1a、.....、E8a、E1b、.....、E8b)直接地或者通过其他节点(v1、.....、v8)间接地互连,并且利用所述链路(E1a、.....、E8a、E1b、.....、E8b)通过交换帧(F1、.....、F4)互相通信,

其中至少一个节点(v1、.....、v8)使至少一个队列(Q1、.....、Q3)与至少一个链路(E1a、.....、E8a、E1b、.....、E8b)关联,其中所述至少一个节点(v1、.....、v8)连接到所述至少一个队列(Q1、.....、Q3),并且

其中所述至少一个节点(v1、.....、v8)将帧(F1、.....、F4)放入所述至少一个队列(Q1、.....、Q3)中,以在所述关联链路(E1a、.....、E8a、E1b、.....、E8b)上传输,

并且其中所述至少一个队列与能够处于打开状态或关闭状态的门电路(G1,...,G3)关联,

并且其中如果与所述队列关联的所述门电路的状态处于打开状态,则所述至少一个节点从被放入在与所述队列关联的至少一个链路上的所述至少一个队列中的帧选择(TS)以进行传输,并且当所述门电路处于关闭状态时,不从所述链路上的队列中选择帧进行传输;

并且其中所述门电路随着时间的推移使所述状态从打开变更为关闭和从关闭变更为打开,从而形成窗口(W1、.....、W4),如配置数据(GCL)所示,

其特征在于,

所述实时网络包括至少一个根据权利要求8至10中的任何一项所述的设备,用于配置所述实时网络的节点(v1、.....、v8)。

12.根据权利要求11所述的网络,其特征在于,OPC/UA和/或DDS和/或NETCONF/YANG和/或SNMP和/或FTP是用于将所述通信要求传送到所述工具并且/或者将配置数据(GCL)传送到所述节点(v1、.....、v8)。

13.根据权利要求11或12所述的网络,其特征在于,所述实时网络的每个节点是由以太网交换机、路由器和端站组成的组中选出。

14.根据权利要求11至13中的任何一项所述的网络,其特征在于,所述状态从打开变更为关闭和从关闭变更为打开是根据IEEE 802.1Qbv标准执行的。

15.根据权利要求11至14中的任何一项所述的网络,其特征在于,优选地利用IEEE 1588标准和/或IEEE 802.1AS标准和/或SAE AS6802标准或所述标准的任一版本,在所述节点中建立同步时间。

用于配置实时网络的方法和设备

技术领域

[0001] 本发明涉及一种在实时网络中配置节点的方法，实时网络的节点利用链路直接地或者通过其他节点间接地互连，并且利用所述链路通过交换帧互相通信，其中至少一个节点使至少一个队列与至少一个链路关联，其中所述至少一个节点连接到所述至少一个队列，并且其中所述至少一个节点将帧放入所述至少一个队列中，以在关联链路上传输，并且其中所述至少一个队列与能够处于打开状态或关闭状态的门电路关联，并且其中如果与所述队列关联的门电路的状态处于打开状态，则所述至少一个节点从被放入在与所述队列关联的至少一个链路上，所述至少一个节点选择被放入的所述至少一个队列中选择帧以进行传输，并且当所述门电路处于关闭状态时，不从所述链路上的队列中选择帧进行传输；并且其中所述门电路使状态随着时间的推移从打开变更为关闭和从关闭变更为打开，从而形成窗口，如配置数据所示。

[0002] 此外，本发明还涉及一种如上所述在实时网络中配置节点的设备。

[0003] 最后，本发明涉及一种实时网络，该实时网络包括至少一个根据本发明配置节点的设备。

[0004] 本发明涉及在保证实时的情况下产生网络配置数据。例如，这种实时网络可以实现IEEE 802.1Qbv[1]，IEEE 802.1Qbv定义传输已排程帧的时间意识形成机制(time-aware shaping mechanism)。时间意识形成器是在队列的出口侧执行的门控机制(gating mechanism)。该门控机制动态地开启或停用基于预定配置数据从各自队列选择帧。在IEEE 802.1Qbv的示例中，该配置数据GCL被称为门控列表(Gate Control List)。

背景技术

[0005] 特别是，IEEE 802.1Qbv定义出口端口的每个队列的门电路，在给定时间，该门电路能够处于两个定义状态中的一个：打开或关闭。当门电路处于打开状态时，可以从各自队列中选择帧，从而以先进先出顺序传输到物理链路。如果该门电路处于关闭状态，则不从各自队列选择帧。在相对于网络中的同步时间工作时，离线估算状态变更的时间点，并且然后执行状态变更。

发明内容

[0006] 本发明的目的在于简化实时网络的配置并且改善产生配置数据的性能。

[0007] 该目的由上面描述的方法实现，其特征在于，所述配置数据由一工具产生，其中所述工具能够解算阵列理论中的约束，并且其中所述工具接受输入，该输入公式化为阵列理论中的约束，并且其中利用人手用户的输入和/或来自实时网络中的一个、两个、或多个节点的通信请求，所述输入提供给所述工具，并且其中所述配置数据传送到实时网络中的一个、两个或多个节点，节点或各节点将至少部分所述配置数据或所述配置数据用作本地配置。

[0008] 此外，该目的由上面描述的设备实现，其特征在于，该设备能够通过执行工具产生

配置数据，其中所述工具能够解算阵列理论中的约束，并且其中所述工具配置成接受输入，该输入公式化为阵列理论中的约束，其中利用人手用户的输入和/或来自实时网络中的一个、两个、或多个节点的通信请求，所述输入提供给所述工具，并且其中该设备配置成将所述配置数据传送到实时网络中的一个、两个或多个节点，该一个、两个或多个节点将至少部分所述配置数据用作本地配置。

[0009] 最后，该目的由技术领域中描述的实时网络实现，其中该实时网络包括上面描述的至少一个设备，用于配置该实时网络的节点。

[0010] 公开了一种配置方法：通过对一些约束的公式化，对尤其表示已排程窗口的打开时间瞬时和关闭时间瞬时的变量之间的依赖性及对特定窗口分配已传递的帧进行编码。可以不同方式公式化这些约束的编码，使得使用类似专门解算器的工具能够找到满意的解决办法。我们特提议将阵列的一阶理论(在本申请中亦称为阵列的理论或阵列理论)[2]的适当性作为对由排程问题产生的约束进行编码的适当方式，然后，该排程问题就能够由类似SMT解算器的通用工具解算。

[0011] 围绕两个解释功能码元构建阵列的一阶理论($\mathcal{T}_A$)：选择，该选择用于从特定索引返回阵列的元素；以及存储，该存储用于将元素写入阵列内的特定索引处。除了线性整数运算中的常用运算符，我们还使用[2]中出现的语法来引入阵列理论并且表达排程器约束。通常，$\mathcal{T}_A$的特征(signature)被定义为Σ_A：{•[•]，•<•←•>，=}。在[2]中。分类阵列，elem和索引分别用于表示阵列、元素和索引。此外，语法a[i]用于表示阵列a中的索引i处的元素的选择功能，并且a<i←e>用于表示阵列a中的索引i处的元素e的存储操作。阵列理论中的两个主要公理是[2]：

[0012] $\forall a{:}array, \forall i,j{:}index, \forall x{:}elem$

[0013] i＝j>a<i<x>[j]＝x

[0014] i≠j→a<i←x>[j]＝a[j]

[0015] 这些与线性整数运算的公理一起形成整数索引阵列($\mathcal{T}_A^{(\mathbb{Z})}$)的理论，我们利用该理论表达排程约束。

[0016] 下面将详细描述上面描述的方法、设备和实时网络的有利实施例：

[0017] ●阵列理论中的一些约束可以是格式正确的窗口约束和/或队列分配约束和/或串流实例约束和/或有序窗口约束和/或帧-窗口分配约束和/或窗口大小约束和/或串流约束和/或串流隔离约束。

[0018] ●SMT解算器可用作能够解算阵列理论中的表达式的所述工具。

[0019] ●OPC/UA和/或DDS和/或NETCONF/YANG和/或SNMP和/或FTP用于或可用于将通信要求传送到所述工具并且/或者将配置数据传送到节点。

[0020] ●实时网络的每个节点可以是由以太网交换机、路由器和端站组成的组中选出。

[0021] ●可根据IEEE 802.1Qbv标准执行从打开变更为关闭和从关闭变更为打开的状态变更。

[0022] ●优选地，利用IEEE 1588标准和/或IEEE 802.1AS标准和/或SAE AS6802标准或所述标准的任一版本，可在节点中建立同步时间。

附图说明

[0023] 下面为了进一步说明本发明，讨论如图所示的说明性的非限制性实施例。附图中：

[0024] 图1示出根据现有技术的实时网络的示例，

[0025] 图2示出根据现有技术的串流和帧的示例，

[0026] 图3示出根据现有技术的节点的示例性实现，

[0027] 图4示出根据现有技术的通信状况的示例，

[0028] 图5示出根据本发明的方法的示例的流程图，

[0029] 图6示出根据本发明的方法中采用的阵列理论的约束，以及

[0030] 图7示出根据本发明用于在实时网络中对节点产生配置的设备。

具体实施方式

[0031] 接着，我们将讨论本发明的许多实施中的一些实施。如果未另外说明，则对特定示例描述的所有细节不仅对该示例有效，而且适用于本发明的整个保护范围。

[0032] 图1示出包括利用链路E1a、.....、E8a、E1b、.....、E8b互相连接的8个节点v1、.....、v8的实时网络的示例。通常，网络建模为图$G=(V,E)$，其中V是例如v1、.....、v8的一组节点，并且ε是将节点互相连接的一组有向链路(directed links)，例如，E1a、.....、E8a、E1b、.....、E8b。如果在两个节点v_i和v_j之间存在物理连接，则该物理连接提供两个链路，每个通信方向一个链路，即，在两个节点之间可在两个方向上进行通信。严格地说，如果节点v_i和v_j互相连接，则定义两个有向链路$e_{i,j},e_{j,i}\in\varepsilon$。节点可以是帧的源或目的地，也可以将帧转发到其他节点。在一种实现中，链路可以是以太网链路，并且实时网络的每个节点是从由以太网交换机、路由器和端站组成的组中选择的。

[0033] 节点互相以串流和帧的概念通信。串流(或流)是从一个发送器(发送机)到一个或者多个收听器(接收机)的周期性多点传播数据传输。我们用$\mathcal{S}$表示网络中的一组串流。我们将串流$s_i\in\mathcal{S}$通过中间节点(即，交换机)v_2、v_3.....、v_{n-1}从发送器v_1路由到收听器v_n的路线表示为$\mathcal{R}_i=[e_1,\ldots,e_{n-1}]$。

[0034] 我们假定对于每个串流，已知发送机节点和接收机节点v_1、v_n以及将其连接的已划定通信通路。

[0035] 串流$s_i\in\mathcal{S}$由元组$<C_i,T_i,L_i,J_i)$定义，该元组分别表示以字节为单位的帧大小、周期、最大允许端到端等待时间、以及串流的最大允许抖动。

[0036] 路由通过链路$e\in\varepsilon$的串流$s_i\in\mathcal{S}$的实例由一组帧表示$f_0\in\mathcal{F}_i^e$，其中$\mathcal{F}_i^e\subset\mathcal{F}^e$是将通过链路e排程的串流$s_i$的全部帧的组。我们用$\mathcal{F}^e$表示路由通过链路e的全部帧的组。由于串流可以具有不同的周期(period)，导致总排程循环(schedule cycle)(也称为超周期)大于任一单独串流周期，因此当构建GCL时，我们必须考虑排程循环之前重复的全部特定串流例。因此，组$\mathcal{F}_i^e$具有T_s/T_i个帧，其中T_s是网络中的全部已排程串流(scheduled stream)的排程循环，算出的排程循环为全部串流$s_i\in\mathcal{S}$的周期的最小公倍数。此外，链路$e\in\varepsilon$上的每个这种周期性帧的特征是基于串流s_i的数据量C_i和与物理链路e关联的出口端口的速度估算的帧传输时长δ_e。例如，84字节和1542字节的最小以太网帧和最大以太网帧(包含IEEE 802.1Q标志)在1Gbit/sec的链路上分别具有0.672μsec和12.336μsec的时长。

[0037] 图2示出图1所示示例性实时网络中的示例性串流和帧的示例。在该例中，示出串

流S1，其中节点v1是发送机，而v7和v8是接收机节点。以帧的形式执行串流的通信，帧被示为F1、F2、F3、和F4。在一种实现中，所传送的帧是以太网帧。

[0038] 图3示出节点的内部结构的一种可能实现的示例。该示例可以是图1所示示例性网络中的节点v4的实现。该节点在其输入链路E1a、E2a、E3a中接收帧。交换逻辑SW1判定必须使帧对哪个输出链路E5a接入和需要将帧置于与各自的输出链路E5a关联的一组队列Q1、Q2、Q3中的哪个队列中。此外，该图还示出与队列Q1、Q2、Q3关联的门电路G1、G2、G3。这些门电路或者处于打开状态，或者处于关闭状态，如配置数据GCL所示，并且随着时间发展，如本地时钟C1所示。节点还保持传输选择块TS，该传输选择块TS从队列Q1、Q2、Q3中选择下一个帧进行传输。如果关联门电路G1、G2、G3处于打开状态，则传输选择块TS仅从给定队列Q1、Q2、Q3中选择帧。

[0039] 图4示出图2所示示例性串流S1的帧F1至F4在图1所示示例性网络中是如何排程和通信的示例。门电路的状态从关闭变更为打开和从打开变更为关闭的连续事件定义窗口W1至W4。阵列$\mathcal{Q}_e$表示对链路$e\in\varepsilon$上的每个窗口分配的队列。我们利用we表示求得的每个链路$e\in\varepsilon$的窗口的最大数。

[0040] 为了在$\mathcal{X}^{(2)}$中编码排程问题，我们在分类阵列中对每个链路e定义两个阵列Φ^e和τ^e，这两个阵列Φ^e和τ^e分别对于窗口的打开时间瞬时O1至O4和关闭时间瞬时C1至C4含有整数变量，对于与链路e关联的出口端口，由在两个阵列中的位置索引该窗口。此外，我们在分类索引中对每个帧例$f_i\in F$定义窗口索引ω_i，该分类索引表示上述两个阵列中的帧-窗口分配索引。相对于网络中的同步时间测量打开时间瞬时O1至O4和关闭时间瞬时C1至C4。例如，利用IEEE 1588标准和/或IEEE 802.1AS标准和/或SAE AS6802标准或所述标准的任一版本，能够在节点中建立同步时间。

[0041] 图5示出根据本发明的配置方法的流程图。该流程图包括采集通信要求的第一步REQ。这能够例如由人手用户输入或由来自网络中的元素的自主请求实现。在一种实现中，OPC统一架构(OPC Unified Architecture)OPC/UA[3]用于公式化这些请求。在另一种实现中，网络配置协议(Network Configuration Protocol)(NETCONF)[4]和YANG[5]用于公式化这些请求。在另一种实现中，数据分布服务(Data Distribution Service)(DDS)[6]用于公式化这些请求。在另一种实现中，简单网络管理协议(Simple Network Management Protocol)(SNMP)[7]用于公式化这些请求。通信请求一被采集到，就在第二步CON将其公式化为阵列理论中的约束。在第三步TOOL，将阵列理论中的约束提供给工具，该工具能够解算阵列理论公式化的约束。在第四步CONF，该工具返回满足来自第二步CON的阵列理论约束的配置数据。在第五步DIST，至少一些配置数据分布于网络中，并且网络中的至少一些节点将至少一些配置数据改编作其本地配置。这种分布可由类似OPC/UA、DDS、NETCONF/YANG、SNMP等的协议执行。

[0042] 图6示出阵列理论中的约束。通常，下面出现的所有约束都是必需的，然而，在特定应用中，仅提供下面公式化的一个或一些约束。

[0043] ●格式正确的窗口约束：

[0044] 我们首先对出口端口的所有窗口形式化逻辑约束。由于每个物理链路将一个出口端口连接到一个进口端口，所以我们假定出口端口与所连接的有向边(物理链路)在形式论

(formalism)中相等。我们将该链路上定义的每个窗口的打开和关闭的事件限制为大于或者等于0并且小于或者等于网络中的所有串流的排程循环。因此，我们有约束CON-WELLFORMED：

[0045] $\forall e \in \mathcal{E}: \forall k \in \{1,\ldots,\mathcal{W}^e\}:$

[0046] $(\phi^e[k] \geqslant 0) \wedge (\tau^e[k] < T_s)$

[0047] 其中如上所定义的，T_s是网络中所有通信串流的排程循环(超周期)。

[0048] ●队列分配约束

[0049] 每个窗口分配到在范围[0..MAX]中排程的外出队列，因此，我们对队列分配阵列CON-QUEUEASS增加界限：

[0050] $\forall e \in \mathcal{E}: \forall k \in \{1,\ldots,\mathcal{W}^e\}:$

[0051] $0 \leqslant \kappa^e[k] < MAX$

[0052] ●串流实例约束

[0053] 如上所述，网络部署中的通信很少表现标准化周期。相反，串流以多个速率发生，这样导致超周期，该超周期将排程表的长度定义为至少是所包括的所有周期的最小公倍数。帧的分配是，并且因此每个窗口的长度也是排程器的结果。具有不同周期、路由通过同一个链路的串流将贡献许多帧例，在超周期之前出现的每个周期例内排程每个帧例。因此，每个窗口的打开和关闭界限是设定窗口的进一步约束。

[0054] 对于路由通过e的每个串流s_i，我们建立下面的约束CON-STREAMINST：

[0055] $\forall s_i \in \mathcal{S}: \forall e \in \mathcal{E}: \forall j \in \left[0, \frac{T_s}{T_i} - 1\right]:$

[0056] $(\phi^e[\omega_{i,j}^e] \geq j \times T_i) \wedge$

[0057] $(\tau^e[\omega_{i,j}^e] < (j+1) \times T_i)$

[0058] ●有序窗口约束

[0059] 确定论(determination)必需的约束是不会将两个帧排程通过同一个外出链路发送，以致在时域中发生重叠。此外，我们明确禁止多个窗口同时保持打开，以避免争用引起的不确定论。

[0060] 从理论上说，该约束的公式化不允许任何两个窗口于同一个链路上而重叠，我们将此定义为CON-ORDWIND1：

[0061] $\forall e \in \mathcal{E}: \forall i,j \in \{1,\ldots,\mathcal{W}^e\}, i \neq j:$

[0062] $(\tau^e[i] \leqslant \phi^e[j]) \vee (\tau^e[j] \leqslant \phi^e[i])$

[0063] 请注意，该公式化产生大量带析取运算符(disjunction operator)的确证，已经证明该确证是计算密集型的。然而，由于不事先限制窗口的分配帧，并且任意帧可以分配到任意窗口，所以如果窗口在每个链路上的顺序被离线预定，因此，将其各自的打开和关闭事件设定为有序，我们就能够简化该约束。因此，可以使用下面的另一公式化CON-ORDWIND2：

[0064] $\forall e \in \mathcal{E}: \forall i \in \{1,\ldots,\mathcal{W}^e - 1\}:$

[0065] $\tau^e[i] \leqslant \phi^e[i+1]$

[0066] ●帧-窗口分配约束

[0067] 帧-窗口分配变量定义各自出口端口的三个阵列(打开、关闭、和队列分配)中的索

引。因此，我们将该变量限制到不大于每个端口的可配置最大的窗口数目CON-FRTOWIND：

[0068] $\forall e \in \mathcal{E}: \forall f_{[,]}^{e} \in \mathcal{F}^{e}:$

[0069] $(\omega_{[,]}^{e} \geq 1) \wedge (\omega_{[,]}^{e} \leq \mathcal{W}^{e})$

[0070] ●窗口大小约束

[0071] 由于排程器对窗口分配帧，即，其事先不知道，所以窗口的大小由对其分配的全部帧的时长的累积和获得。因此，我们必须确保每个窗口的关闭事件允许有足够的时间发送一组分配帧。我们注意到该约束是在我们的形式论中要求阵列理论的存储操作的唯一的第一约束。

[0072] 我们通过将每个打开变量的未经解释的项存储于关闭阵列的各自位置来开始CON-WINDSIZE1：

[0073] $\forall e \in \mathcal{E}: \forall k \in \{1, \ldots, \mathcal{W}^{e}\}:$

[0074] $\tau^{e}\langle k \leftarrow \Phi^{e}[k] \rangle$

[0075] 这等于将全部打开事件于相同索引设定为等同关闭事件，因此，将窗口的长度初始化为0。请注意，这样，所有没有分配帧的窗口的关闭事件将保持等于各自的打开事件。

[0076] 现在，我们利用帧-窗口分配索引在关闭阵列的每个位置处建立对该窗口分配的全部的帧的时长的和CON-WINDSIZE2：

[0077] $\forall e \in \mathcal{E}: \forall f_{[,]}^{e} \in \mathcal{F}^{e}:$

[0078] $\tau^{e}(\omega_{[,]}^{e} \leftarrow \tau^{e}[\omega_{[,]}^{e}] + l_{[,]}^{e})$

[0079] 该构造对全部帧反复将帧时长加到该帧分配到的窗口的关闭事件的之前的未解释的值，并且以相同索引将该结果存储为新的未解释的表达。

[0080] ●串流约束

[0081] 串流约束描述从发送器到收听器的通信的有序性质。一般条件是属于同一个串流的帧必须沿路由选择的通信路径根据时间顺序地排程。因此，我们有CON-STREAM：

[0082] $\forall s_{i} \in \mathcal{S}: \forall e_{k} \in \mathcal{R}_{i}, k \in \{1, \ldots, n-1\}:$

[0083] $\forall j \in \left\{0, \cdots, \frac{T_{i}}{T_{i}} - 1\right\}:$

[0084] $\forall f_{[,]}^{e_{k}} \in \mathcal{F}_{i}^{e_{k}}:$

[0085] $\forall f_{[,]}^{e_{k+1}} \in \mathcal{F}_{i}^{e_{k+1}}:$

[0086] $\tau^{e_{k}}[\omega_{[,]}^{e_{k}}] + \delta \leq \phi^{e_{k+1}}[\omega_{[,]}^{e_{k+1}}].$

[0087] 其中δ是表示网络精度的恒定值。

[0088] 换句话说，串流的帧的传播沿路径遵循顺序次序。因此，每个帧的窗口打开事件必须大于或者等于对前任的帧分配的窗口关闭事件，加网络精度常数以补偿两个跳线(hops)之间的时钟差。

[0089] ●串流隔离约束

[0090] 在运行时，网络似乎可能遭受帧丢失或串流在其周期性有用负载(payload)大小方面表现出差异。因此，为了确保运行时执行排程符合离线规划，我们需要计算在任何给定时间瞬时保证每个队列的确定性状态的排程。

[0091] 考虑在设备a上分别从不同链路e_x和e_y收到两个串流s_i和s_j并且通过链路e_a将二者转发到同一个出口端口的情况。如果将其所述帧放入同一个队列中，则在运行时，帧在该队列中的顺序可根据确切到达顺序的最小变化或交换机结构中的进口端口的处理机制而不同。此外，一个或者另一个串流中的帧的丢失同样可以在每个周期实例处导致队列状态产生差异。因此，外出队列的离线排程打开和关闭可在运行时由队列的非确定性状态导致有效发生不同的行为。

[0092] 保证确定论意味全部帧遵守其在整个网络的操作中计算过的窗口分配。为此，我们需要将上述示例中的两个串流各自的帧分配给同一个窗口，或者在时域中隔离它们，在另一个串流的帧被排程输出之前，限制收到第二串流的帧。作为另一种选择，如果多个队列可用于排程业务，则我们能够将不同队列的窗口中的两个帧隔离，在这种情况下，这两个帧也在重叠间隔内被接收，而不改变运行时的行为。

[0093] 因此，对于通过链路$e_{a,b}$传递的串流s_i和s_j，我们将串流隔离条件公式化为CONSTREAMISOL：

[0094] $$\forall k \in \left[0, \frac{T_s}{T_i} - 1\right]; \forall l \in \left[0, \frac{T_s}{T_j} - 1\right];$$

[0095] $$((\tau^{e_a}[\omega_{i,k}^{e_a}] + \delta \leq \phi^{e_a}[\omega_{j,l}^{e_a}]) \vee$$

[0096] $$(\tau^{e_a}[\omega_{j,l}^{e_a}] + \delta \leq \phi^{e_a}[\omega_{i,k}^{e_a}])) \vee$$

[0097] $$(\kappa^{e_a}[\omega_{i,k}^{e_a}] \neq \kappa^{e_a}[\omega_{j,l}^{e_a}]) \vee$$

[0098] $$(\omega_{i,k}^{e_a} = \omega_{j,l}^{e_a}).$$

[0099] 其中该3个析取条件保证：或者当两个帧中的一个帧在接收时另一个已经被转发(通过将对每个帧分配的窗口各自的打开事件和关闭事件的顺序进行比较)，或者每个帧被分配到不同的队列(并且因此，分配到不同的窗口)；或者两个帧都被分配到同一个窗口(并且因此，分配到同一个队列)。

[0100] 图7示出与图1所示网络类似的实时网络，然而，在根据本发明的网络的该示例中，使用额外节点v9。该节点v9实现设备CNC，其中所述工具能够解算阵列理论中的约束。例如，通过用户接口，人手用户能够将通信要求传送到设备CNC。在另一种实现中，例如，通信要求能够由网络中的节点v1至v8中的一些节点利用诸如OPC/UA、DDS、NETCONF/YANG、SNMP等的标准网络协议和数据模型自主提供。在另一种实现中，节点v9可以常驻于云基础设施中，并且链路E9a和E9b可以是互联网连接(可包括许多以太网链路、交换机和路由器)。

[0101] 参考资料

[0102] [1]Institute of Electrical and Electronics Engineers,"802.1Qbv-Enhancements for Scheduled Traffic,"2017.[Online].可取:http://www.ieee802.org/1/pages/802.1bv.html.

[0103] [2]A.R.Bradley,Z.Manna and H.B.Sipma,"What's Decidable About Arrays?,"在VMCAI,2006.

[0104] [3]OPC Unified Architecture,可取自https://opcfoundation.org/about/opc-technologies/opc-ua/

[0105] [4]Network Configuration Protocol(NETCONF)IETF RFC 6241,可取自

https://tools.ietf.org/html/rfc6241

[0106] [5]YANG-A Data Modeling Language for the Network Configuration Protocol(NETCONF)IETF RFC 6020,可取自https://tools.ietf.org/html/rfc6020

[0107] [6]Data Distribution Service(DDS)可取自http://www.omg.org/spec/DDS/1.4

[0108] [7]Simple Network Management Protocol(SNMP)e.g.,version 3IETF RFC 3411-3418,可取自https://www.ietf.org/rfc.html

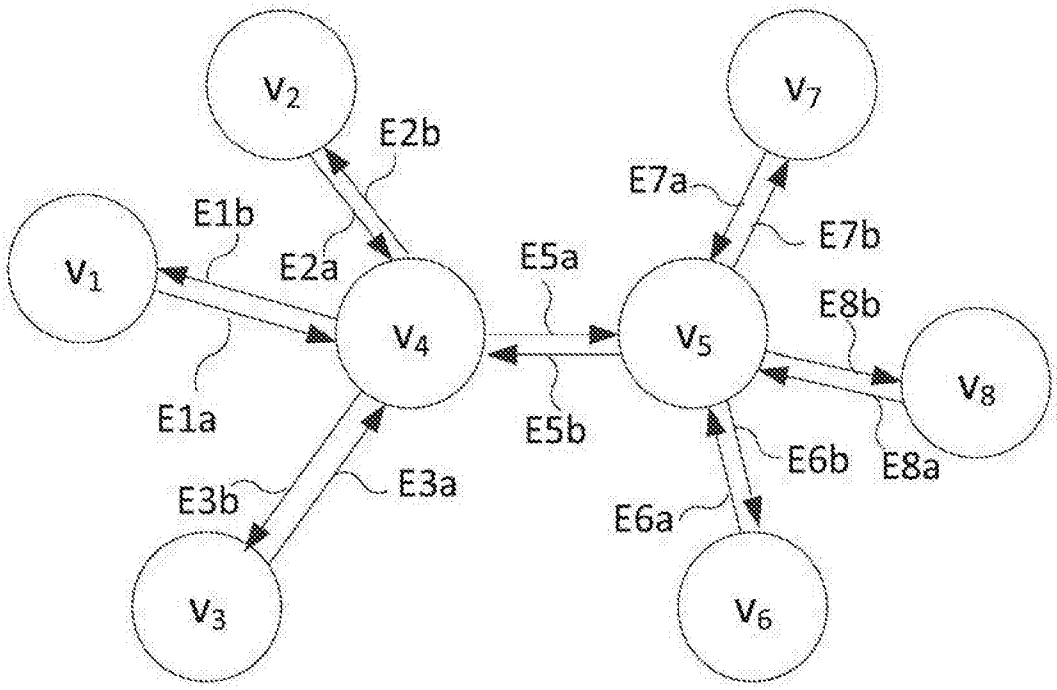


图1(现有技术)

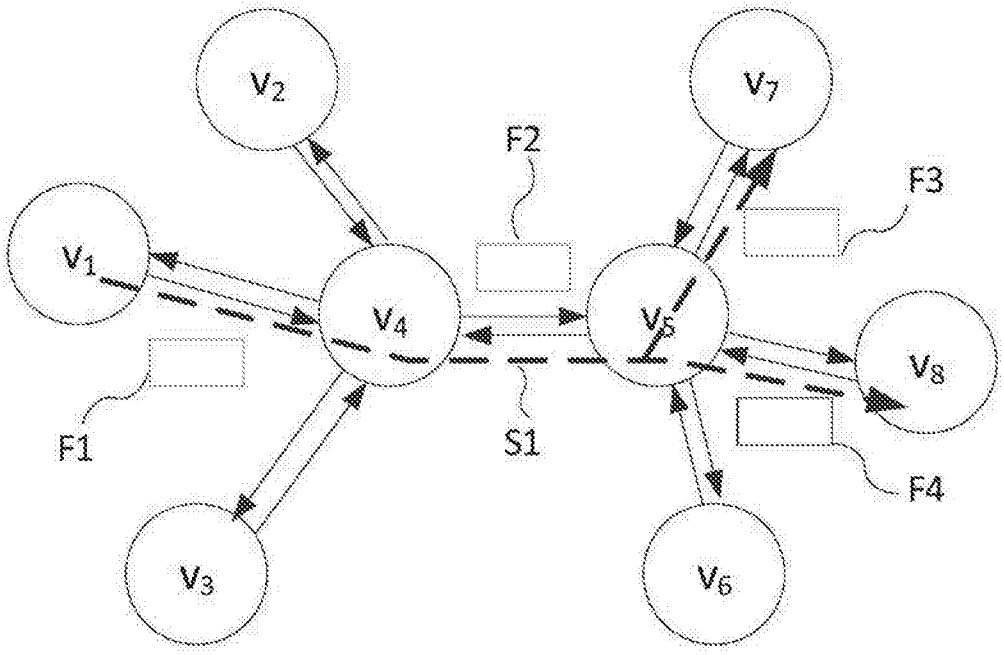


图2(现有技术)

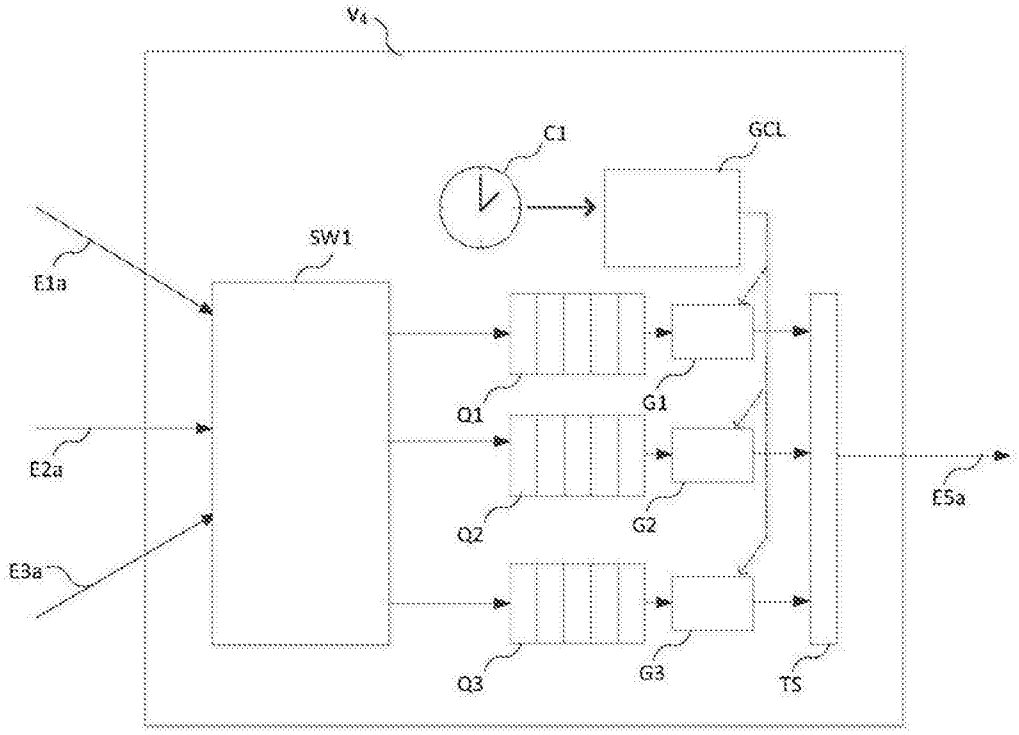


图3(现有技术)

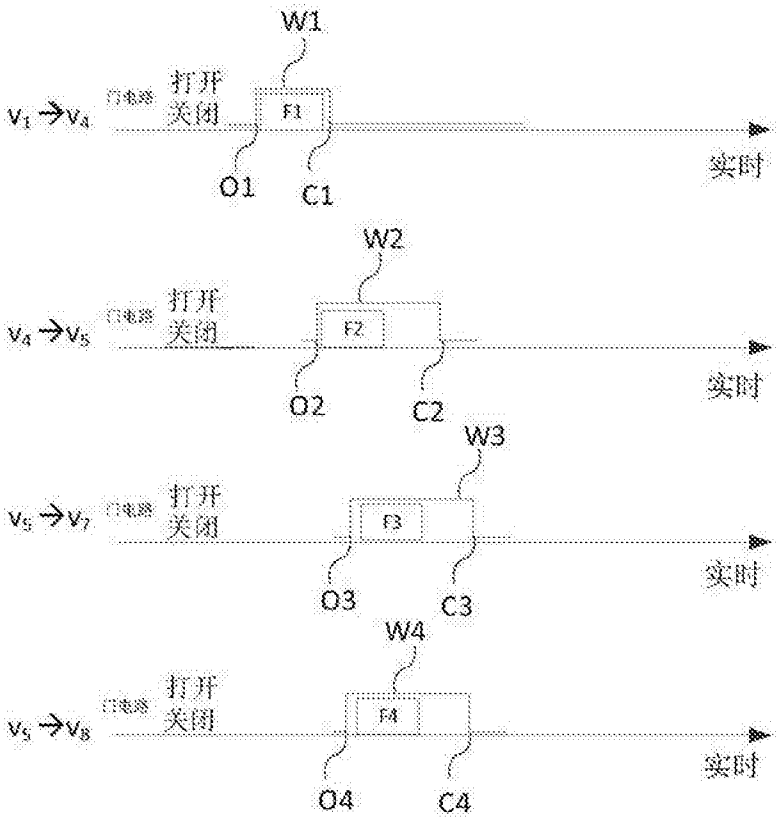


图4(现有技术)

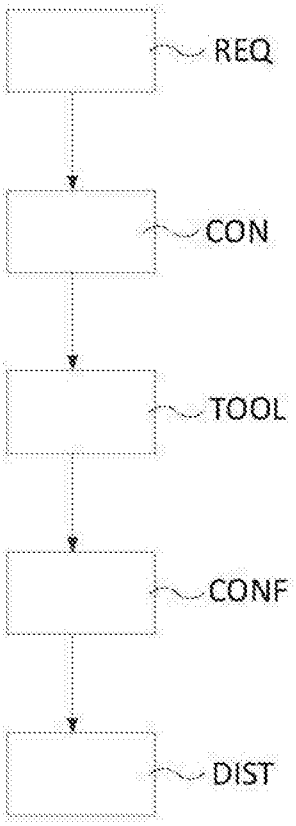

REQ
CON
TOOL
CONF
DIST

图5

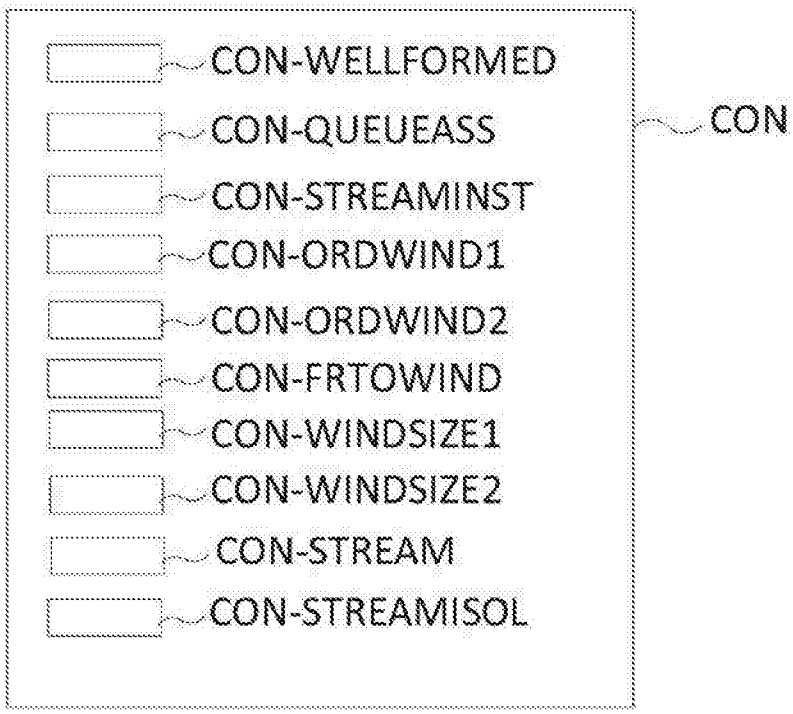

CON-WELLFORMED
CON-QUEUEASS
CON-STREAMINST
CON-ORDWIND1
CON-ORDWIND2
CON-FRTOWIND
CON-WINDSIZE1
CON-WINDSIZE2
CON-STREAM
CON-STREAMISOL
CON

图6

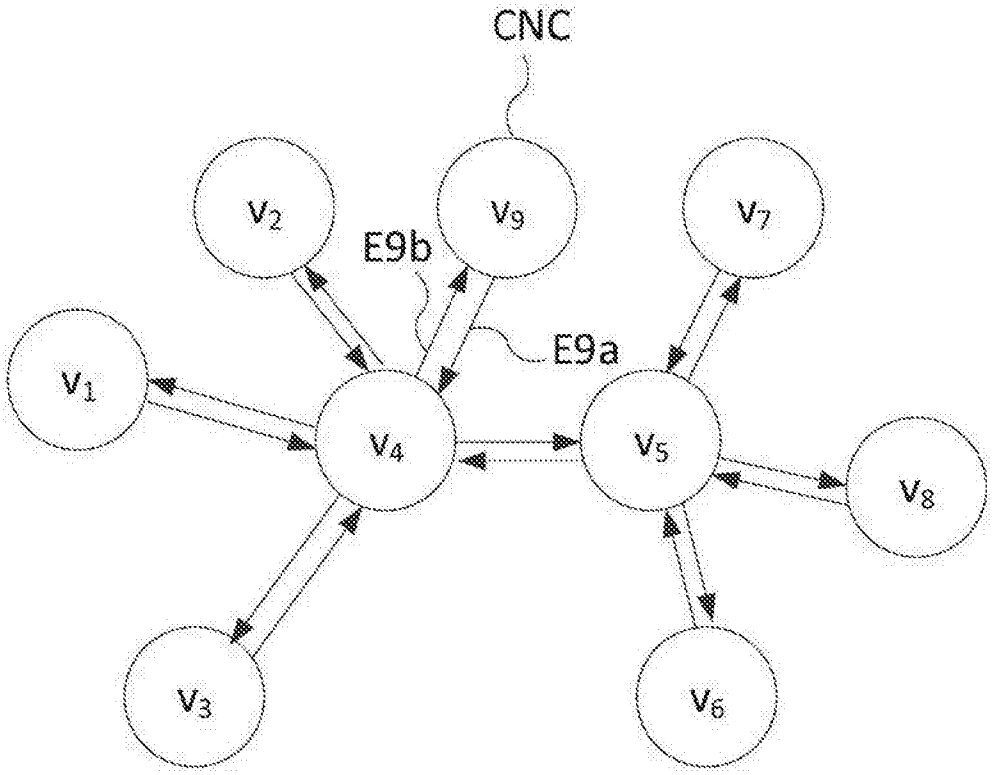


图7