



(12)发明专利申请

(10)申请公布号 CN 106506404 A

(43)申请公布日 2017.03.15

(21)申请号 201610952257.6

(22)申请日 2016.10.27

(71)申请人 昆明理工大学

地址 650093 云南省昆明市五华区学府路
253号

(72)发明人 唐菁敏

(51)Int.Cl.

H04L 12/875(2013.01)

H04L 12/863(2013.01)

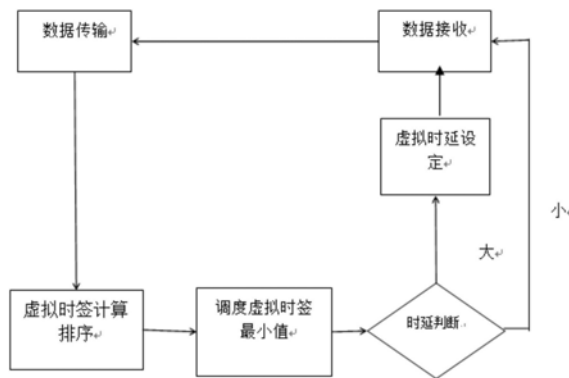
权利要求书2页 说明书7页 附图5页

(54)发明名称

一种智能电网通信的数据传输时延的优化方法

(57)摘要

本发明公开了一种智能电网通信的数据传输时延的优化方法,属于智能电网数据传输领域。本发明通过对不同类型的数据引入相应的不同的“虚拟时延”进行优化,结合其数据包的到达时间和相应的截止时间,得出一个虚拟的时签,动态的调整其调度顺序。时延越是敏感的数据,其“虚拟时延”越小,虽然其数据包的实际时签没有变化,但“虚拟时延”使其虚拟时签发生改变,这样一方面保证时延敏感的数据包优于时延不敏感的数据包,另一方面,“虚拟时延”的引入也解决了多米诺骨牌现象,在不影响时延不敏感数据的基础上,提升时延敏感数据的时延性能。



1. 一种智能电网通信的数据传输时延的优化方法,其特征在於:包括如下步骤:

步骤01) 在某一个交换节点处,首先开始计算虚拟时签,即计算最先到达的六类数据的数据包的虚拟时签,同时进行排序,然后从六类数据包中找出最小虚拟时签的数据包,被选定的调度数据包的虚拟时签假定为 $TStamp_virtual_i^j$,式中表示第i类数据中的第j个数据包的虚拟时签;

步骤02) 利用上一次的计算结果,当下一个数据包进行调度时,第一步先在调度候补中放入第i类数据中的第(j+1)个数据包,第二步用第i类数据中的第(j+1)个数据包的虚拟时签 $TStamp_virtual_i^{j+1}$ 和其他剩余的五个虚拟时签中最小的一个值先进行比较,假如该虚拟时签 $TStamp_virtual_i^{j+1}$ 比其他剩余的五个虚拟时签中的最小值还要小的话,则直接对虚拟时签 $TStamp_virtual_i^{j+1}$ 中所对应的数据包进行相应的调度,否则,就调度虚拟时签为 $TStamp_virtual_m^n$ 所对应的数据包,即调度虚拟时签的最小值;

步骤03) 在此用插入的方式,将第m类数据中的第(n+1)个数据包对应的虚拟时签 $TStamp_virtual_m^{n+1}$ 和第i类数据中的第(j+1)个数据包对应的虚拟时签 $TStamp_virtual_i^{j+1}$ 加入到新的六维数组中,与此同时实现排序;

步骤04) 当调度进行到下一数据包时,通过该交换节点之后的某个数据包,它的结束时间和它在物理链路上传播所消耗的时延之和,即成为下一个交换节点的到达时间。

2. 根据权利要求1所述的智能电网通信的数据传输时延的优化方法,其特征在於:所述的虚拟时签的获得方法为:

对于每一类数据的虚拟时延,其定义如公式下所示:

$$Vl_i = \begin{cases} 0 & \text{当 } i=1 \text{ 时} \\ \sum_{m=1}^{i-1} F_m^j & \text{当 } i \neq 1 \text{ 时} \end{cases}$$

对于第i类数据中的第j个数据包来讲,其发送时间如下公式所示:

$$F_i^j = \frac{P_i}{C}$$

则对该数据包来讲,其时签为:

$$TStamp_i^j = Tarriv_i^j + Deadline_i$$

引入虚拟时延之后,其对应的虚拟时签为:

$$TStamp_virtual_i^j = Tarriv_i^j + Deadline_i + Vl_i$$

上述各个概念公式的含义如下:

m: 表示第m类数据,

$\sum_{m=1}^{i-1} F_m^j$ 表示第m类数据的第j个数据包,是从m=1即第一类数据包开始累加至第i-1类

数据包的所有虚拟时延之和,

$Tarriv_i^j$: 第*i*类数据的第*j*个数据包的到达时间, 其中*i*=1,2,3,4,5,6; *j*=1,2,3,4...

F_i^j : 第*i*类数据的第*j*个数据包的发送时间, 和数据包大小以及端口速率有关,

C: 代表交换节点处端口的速率,

P_i : 代表第*i*类数据包的大小,

Vl_i : 引入的第*i*类数据的虚拟时延,

$Deadline_i$: 表示第*i*类数据包的截止时间,

$TStamp_i^j$: 第*i*类数据中的第*j*个数据包的时签,

$TStamp_virtual_i^j$: 第*i*类数据中第*j*个数据包的虚拟时签。

一种智能电网通信的数据传输时延的优化方法

技术领域

[0001] 本发明属于智能电网数据传输领域,具体来说,涉及一种智能电网通信的数据传输时延的优化方法,有利于智能电网通信中数据传输的性能提升。

背景技术

[0002] 作为下一代电网技术,智能电网将有机地结合信息技术、自动控制、传感技术以及智能电力设备等。它通过提供双向互动、高度集成的服务和模式,使得电网更加安全、可靠、高效和智能。然而对于智能电网通信网来说,一方面负着未来数十年的电网支撑的重任,随着未来电力设备更多的引入,电力系统更多的升级,从规模上以及运行复杂度上都将面临众多挑战;另一方面,更多新能源的介入以及电力逐渐市场化的趋势,智能电网通信网需要提供更多快捷的功能服务以及增值服务,这些都将对智能电网通信网中不同系统之间的信息共享以及信息交互提出了更高的要求。

[0003] 为了提高智能电网中的数据传输速率,虚拟时延概念被提出来。在智能电网通信网中,其数据分为硬实时、准实时、非实时等多种类型,其对应的时延要求从8ms到5秒不等。不同于传统的数据业务时延,智能电网通信网中的时延要求跨度极大,而且其数据时延不仅关系着信息及时的交互,更是关系着整个电网的安全以及电力系统的快速反应。尤其是对于时延敏感的数据来讲,越是情况糟糕关键信息的实时性要求就越高、越严格智能电网通信网中各类数据的快速实时的传输是牵一发而动全身的,它对智能电网通信网的建设提出了前所未有的挑战。不难发现,解决数据传输时延难题不仅是建设智能电网通信网的关键所在,也是保证未来智能电网通信网可靠、安全、高效、智能的核心基础,该问题的研究及解决意义重大。

发明内容

[0004] 本发明要解决的技术问题是提供一种智能电网通信的数据传输时延的优化方法,目的在于克服现有技术不足,提供基于最早截止时间优先的改进型VLO(虚拟时延优化)-EDF调度算法,提升智能传输网络的性能。

[0005] 本发明采取的技术方案是:一种智能电网通信的数据传输时延的优化方法,包括如下步骤:

[0006] 步骤01)在某一个交换节点处,首先开始计算虚拟时签,即计算最先到达的六类数据的数据包的虚拟时签,同时进行排序,然后从六类数据包中找出最小虚拟时签的数据包,被选定的调度数据包的虚拟时签假定为 $TStamp_virtual_i^j$,式中表示第i类数据中的第j个数据包的虚拟时签;

[0007] 步骤02)利用上一次的计算结果,当下一个数据包进行调度时,第一步先在调度候补中放入第i类数据中的第(j+1)个数据包,第二步用第i类数据中的第(j+1)个数据包的虚拟时签 $TStamp_virtual_i^{j+1}$ 和其他剩余的五个虚拟时签中最小的一个值先进行比较,假

如该虚拟时签 $TStamp_virtual_i^{j+1}$ 比其他剩余的五个虚拟时签中的最小值还要小的话，则可以直接对虚拟时签 $TStamp_virtual_i^{j+1}$ 中所对应的数据包进行相应的调度，否则，就调度虚拟时签为 $TStamp_virtual_m^n$ 所对应的数据包，即调度虚拟时签的最小值；

[0008] 步骤03) 在此用插入的方式，将第m类数据中的第(n+1)个数据包对应的虚拟时签 $TStamp_virtual_m^{n+1}$ 和第i类数据中的第(j+1)个数据包对应的虚拟时签 $TStamp_virtual_i^{j+1}$ 加入到新的六维数组中，与此同时实现排序；

[0009] 步骤04) 当调度进行到下一数据包时，通过该交换节点之后的某个数据包，它的结束时间和它在物理链路上传播所消耗的时延之和，即成为下一个交换节点的到达时间。

[0010] 进一步地，所述的虚拟时签的获得方法为：

[0011] 对于每一类数据的虚拟时延，其定义如公式下所示：

$$[0012] \quad V_l_i = \begin{cases} 0 & \text{当 } i=1 \text{ 时} \\ \sum_{m=1}^{i-1} F_m^j & \text{当 } i \neq 1 \text{ 时} \end{cases}$$

[0013] 对于第i类数据中的第j个数据包来讲，其发送时间如下公式所示：

$$[0014] \quad F_i^j = \frac{P_i}{C}$$

[0015] 则对该数据包来讲，其时签为：

[0016] m: 表示第m类数据，

[0017] $\sum_{m=1}^{i-1} F_m^j$ 表示第m类数据的第j个数据包，是从m=1即第一类数据包开始累加至第

i-1类数据包的所有虚拟时延之和，

$$[0018] \quad TStamp_i^j = Tarriv_i^j + Deadline_i$$

[0019] 引入虚拟时延之后，其对应的虚拟时签为：

$$[0020] \quad TStamp_virtual_i^j = Tarriv_i^j + Deadline_i + V_l_i$$

[0021] 上述各个概念公式的含义如下：

[0022] $Tarriv_i^j$: 第i类数据的第j个数据包的到达时间，其中 $i=1, 2, 3, 4, 5, 6$; $j=1, 2, 3, 4, \dots$,

[0023] F_i^j : 第i类数据的第j个数据包的发送时间，和数据包大小以及端口速率有关，

[0024] C: 代表交换节点处端口的速率，

[0025] P_i : 代表第i类数据包的大小，

[0026] V_l_i : 引入的第i类数据的虚拟时延，

[0027] $Deadline_i$: 表示第i类数据包的截止时间，

[0028] $TStamp_i^j$: 第i类数据中的第j个数据包的时签，

[0029] $TStamp_virtual_i^j$: 第i类数据中第j个数据包的虚拟时签。

[0030] 本发明的工作原理是:采用VLO-EDF调度算法,对不同的数据类型引入不同的“虚拟时延”,当最后一类数据包的第一个数据包的时签和第一类数据的第一个数据包的时签差值小于引入的“虚拟时延”时,最后一类数据包的虚拟时签可以由原来的最小变为最大,第一类数据包的虚拟时签也可以从之前的最大变为最小,因而,同样的情况,采用VLO-EDF调度算法之后,其调度顺序可能会导致虚拟时签数字越小越先被调度。

[0031] 该算法有两个输入,第一个输入是按照通信网业务模型产生的六类数据,包括数据包大小,到达时间,到达规律以及数据速率。第二个输入是交换节点处的端口速率以及所经过的网络,这里主要考虑网络节点数目和两个节点之间的物理链路距离等。当输入确定下来之后,接下来要计算出各类数据对应的虚拟时延,并依据每类数据中最早到达的数据包进行虚拟时签的计算,找到虚拟时签最小的数据包,进行调度。然后进入到VLO-EDF调度算法处理中。

[0032] 数据包到达时间和转发时间都是无法改变的,但可以引入“虚拟时延”,时延越是敏感的数据,其对应的虚拟时延越小,时延越是不敏感的数据,其对应的虚拟时延越大。这样,原来可能时延不敏感数据时签小于时延敏感数据时签的情况就有可能变成现在的时延敏感数据虚拟时签小于时延不敏感数据的虚拟时签。在该情况下,调度的时候,越是时延敏感的数据包,就会越优先被考虑,促使其服务开始时间提前,进而提升了时延敏感数据包的排队等待时间。一方面在网络负载正常时,调度不存在冲突的情况,各类数据包的服务开始时间都近似于到达时间,即使引入了“虚拟时延”的概念,也不会影响其时延性能,另一方面,在网络负载较大时,各类数据包都会出现等待,而这时引入的“虚拟时延”就可以更好的服务于时延敏感数据,即使出现了突发和偶尔的拥塞,也可以更好的被优先调度,改善了多米诺骨牌现象,通过多跳的调整和调整,极大的提升了其等待时延性能。

[0033] 在负载比较小的时候,VLO-EDF调度算法时延敏感数据的时延抖在突发性的关键事件发生时,可以根据不同时延门限符合智能电网通信网的特点和时延要求。而时延不敏感数据的时延抖动会对影响较小。而当负载增大并出现部分拥塞时,VLO-EDF调度算法在时延抖动上就有了很好的效果。

[0034] 本发明的有益效果是:通过基于最早截止时间优先的改进型VLO(虚拟时延优化)-EDF调度算法,提升智能传输网络的性能。

附图说明

[0035] 图1为智能电网通信网中的时延分解示意图;

[0036] 图2为本发明的系统流程框图;

[0037] 图3为EDF调度算法实例;

[0038] 图4为LBFS调度举例;

[0039] 图5为仿真的负载为88%时所对应的三种调度算法平均时延图;

[0040] 图6为负载流量为88%时所对应的三种调度算法的最大时延图;

[0041] 图7为仿真满足通信时延要求的最大负载流量图;

[0042] 图8为采用VLO-EDF对稳定数据包进行调度后的时延图;

[0043] 图9对实体交换机测试所得时延图。

具体实施方式

[0044] 下面结合附图和具体实施方式对本发明的技术方案作进一步描述。

[0045] 实施例1:如图1-9所示,一种智能电网通信的数据传输时延的优化方法,其特征在于:包括如下步骤:

[0046] 步骤01) 在某一个交换节点处,首先开始计算虚拟时签,即计算最先到达的六类数据的数据包的虚拟时签,同时进行排序,然后从六类数据包中找出最小虚拟时签的数据包,被选定的调度数据包的虚拟时签假定为 $TStamp_virtual_i^j$,式中表示第i类数据中的第j个数据包的虚拟时签;

[0047] 步骤02) 合理利用上一次的计算结果,当下一个数据包进行调度时,第一步先在调度候补中放入第i类数据中的第(j+1)个数据包,第二步用第i类数据中的第(j+1)个数据包的虚拟时签 $TStamp_virtual_i^{j+1}$ 和其他剩余的五个虚拟时签中最小的一个值先进行比较,假如该虚拟时签 $TStamp_virtual_i^{j+1}$ 比其他剩余的五个虚拟时签中的最小值还要小的话,则可以直接对虚拟时签 $TStamp_virtual_i^{j+1}$ 中所对应的数据包进行相应的调度,否则,就调度虚拟时签为 $TStamp_virtual_m^n$ 所对应的数据包,即调度虚拟时签的最小值;

[0048] 步骤03) 在此用插入的方式,将第m类数据中的第(n+1)个数据包对应的虚拟时签 $TStamp_virtual_m^{n+1}$ 和第i类数据中的第(j+1)个数据包对应的虚拟时签 $TStamp_virtual_i^{j+1}$ 加入到新的六维数组中,与此同时实现排序;

[0049] 步骤04) 当调度进行到下一数据包时,通过该交换节点之后的某个数据包,它的结束时间和它在物理链路上传播所消耗的时延之和,即成为下一个交换节点的到达时间。

[0050] “虚拟时签”是建立在“虚拟时延”的基础上所提出的一个概念,以下是对虚拟时签的解释及具体获得的方法:

[0051] 其中对于每一类数据的“虚拟时延”,其定义如公式下所示:

$$[0052] \quad V_i = \begin{cases} 0 & \text{当} i=1 \text{时} \\ \sum_{m=1}^{i-1} F_m^j & \text{当} i \neq 1 \text{时} \end{cases}$$

[0053] 对于第i类数据中的第j个数据包来讲,其发送时间如下公式所示:

$$[0054] \quad F_i^j = \frac{P_i}{C}$$

[0055] 则对该数据包来讲,其时签为:

$$[0056] \quad TStamp_i^j = Tarriv_i^j + Deadline_i$$

[0057] 引入“虚拟时延”之后,其对应的虚拟时签为:

$$[0058] \quad TStamp_virtual_i^j = Tarriv_i^j + Deadline_i + Vl_i$$

[0059] 上述各个概念公式的含义如下：

[0060] m: 表示第m类数据，

[0061] $\sum_{m=1}^{i-1} F_m^j$ 表示第m类数据的第j个数据包，是从m=1即第一类数据包开始累加至第

i-1类数据包的所有虚拟时延之和，

[0062] $Tarriv_i^j$: 第i类数据的第j个数据包的到达时间。其中i=1,2,3,4,5,6; j=1,2,3,4...

[0063] F_i^j : 第i类数据的第j个数据包的发送时间，和数据包大小以及端口速率有关，

[0064] C: 代表交换节点处端口的速率，比如常见的交换机其端口可支持1Gbps的速率，

[0065] P_i : 代表第i类数据包的大小，

[0066] Vl_i : 引入的第i类数据的“虚拟时延”，

[0067] $Deadline_i$: 表示第i类数据包的截止时间，对于一般的截止时间来讲，都是一个常数向量，故把截止时间简单定义为每类数据包的传输时间，

[0068] $TStamp_i^j$: 第i类数据中的第j个数据包的时签，

[0069] $TStamp_virtual_i^j$: 第i类数据中第j个数据包的虚拟时签。

[0070] 本发明结合智能电网通信网的特点和数据业务特点，首先引入了一个“虚拟时延优化”(Virtual Latency Optimization)的概念。在智能电网通信环节中采用引入虚拟时延优化(VLO)的方法解决在运用最早截止时间优先算法(EDF)时出现的非时延敏感数据优于时延敏感数据调度的情况，即基于最早截止时间优先的改进型VLO(虚拟时延优化)-EDF调度算法。智能电网通信网中，在直接运用EDF算法进行调度时，主要决定因素取决于数据的时延敏感性和非时延敏感型。

[0071] 本发明图1中: 其中，D1代表的是发送节点的处理时延；D2代表的是发送节点将发送数据放在缓冲区引起的排队时延；D3代表的是网络时延，严格意义上来讲，其包括数据包的发送时延、接收时延、交换时延、转发表查询时延、传播时延、排队时延；D4为对应的接收节点的排队时延；D5对应的是接收节点的处理时延。

[0072] 图2中: 首先对最先到达的六类数据的数据包进行虚拟时签的计算，并排序，找到其虚拟时签最小的数据包。在下一个数据包的调度时，充分利用上次计算的结果然后用其虚拟时签和其余五个虚拟时签中的最小值先比较，如果比其余五个虚拟时签的最小值小，则只需一次比较就可以对该数据包进行调度。否则，对虚拟时签最小值所对应的数据包进行调度。然后可利用插入的方式将虚拟时签加入到新的六维数组中，同时也完成了排序的功能。进入到下一数据包的调度中，当某个数据包经过该交换节点之后，其结束时间加上在物理链路上传播所消耗的时延，就成为下一个交换节点的到达时间。然后，和上述单个交换节点的处理一样，再次进入到相似的处理过程中。

[0073] 图3中: 假设class1到class6分别代表具有不同时延要求的六类数据，其时延从敏感到不敏感，即class1数据是时延最敏感的数据，class6是时延最不敏感的数据。根据EDF

调度算法,按照时签的大小,其数据包的调度顺序可能会出现如图中数字顺序所示的情况,其中数字越小表示越先被调度。

[0074] 图4中:在没有LBFS调度情况下,时延敏感数据流和时延不敏感数据流的路由如图中的实线箭头所示。在某一时间段内,假设flow1是平均速率为130Mbps的从Host1到Host3的时延敏感数据流,flow2是平均速率为300Mbps的从Host2到Host3的时延不敏感数据流。而当前网络中各个链路上的网络流量负载如图所示。

[0075] 图5、图6、图7为仿真图,仿真实验中所定义的三个参考因数的选择是为了证明所提出的VLO-EDF算法的有效性。

[0076] 图5为仿真的负载为88%时所对应的三种调度算法平均时延。在传输电路平均时延仿真试验中,引入了六个不同类别的数据,对于不同类别的数据,传输要求也应当不同,因此可以区别出三种不同的调度算法下的传输时延的估计结果。

[0077] 图6为负载流量为88%时所对应的三种调度算法的最大时延。引入了六个不同类别的数据,仿真结果表明了三种不同的调度算法下的传输时延,验证了VLO-EDF算法的优越性。

[0078] 图7为仿真满足通信时延要求的最大负载流量。在三种算法下的传输时延抖动,可以看出当负载增大并出现部分拥塞时,尤其是在时延敏感数据的调度时,本发明提出的VLO-EDF算法优化更为效果明显。

[0079] 本发明的具体试验步骤为:

[0080] 步骤01) 对于数据源,按照图1的数据模型,六类数据的数据包大小分别为50、76、80、200、200、1500字节,对于第二类、第三类数据,其特点是周期性的数据,对于第四类数据,其特点是连续性的,对于第五类和第六类数据随机数据,我们采用泊松分布。对于第一类数据,我们采用随机的突发数据。

[0081] 步骤02) 在仿真中,为了验证三种调度算法对时延的影响,我们仅考虑buffer相同的情况。同时,在仿真中也结合智能电网通信网的特点,加入了跳数的控制,在仿真中我们设网络跳数为5。为了仿真的准确性,对数据进行50次处理并求平均值。

[0082] 步骤03) 对于交换节点出的线卡速率,我们采用1Gbps的端口。

[0083] 步骤04) 在仿真实验中,通过对网络流量负载的控制,查看三种调度算法对六类数据时延的影响。

[0084] 仿真要求:具体的实验中,采用的处理器为Intel(R) Core (TM) i3-2370M CPU@2.4GHz2.4GHz,其内存为6.00GB (5.6GB可用)的笔记本,该机器为64位操作系统。

[0085] 单节点仿真结果和实际测量结果比较步骤为:

[0086] 步骤01) 采用VLO-EDF调度算法对大小为1500Bytes,且接近匀速的数据包进行调度,采用了六个速率,分别为100Mbps、200Mbps、500Mbps、750Mbps900Mbps、980Mbps。则该类数据包的时延如附图8所示。

[0087] 步骤02) 我们借助IXIA仪表,在Brocade的NetIron_CES_2000_DS型号交换机上对大小为1500Bytes的数据包进行测试,其端口速率也是1Gbps。所采用的六个速率和VLO-EDF调度算法一样。其结果如附图9所示。

[0088] 步骤03) 从图8、图9两个图中可以看出,当流量不大时,交换节点处没有拥塞,其时延非常小,一般在us量级,而当流量较大时,尤其是接近或者大于900Mbps时,其时延会出现

急剧增加。当数据不是稳定的,而且有突发时,其时延会更大,这一部分将在下面小节中看到。

[0089] 步骤04) 两个图中所展示的时延规律都是接近的,而且其时延数值也相差不大,这验证了VLO-EDF调度算法的可行性。

[0090] 以上结合附图对本发明的具体实施方式作了详细说明,但是本发明并不限于上述实施方式,在本领域普通技术人员所具备的知识范围内,还可以在不脱离本发明宗旨的前提下作出各种变化。

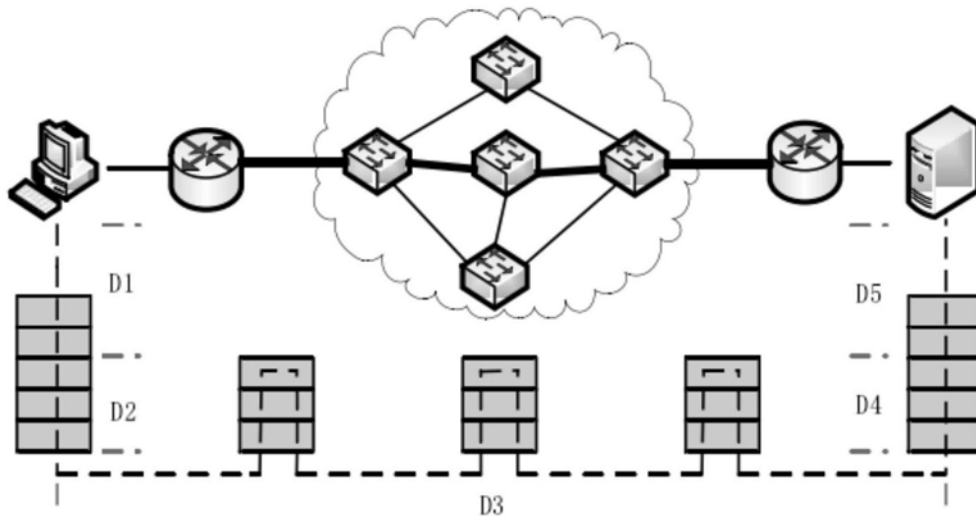


图1

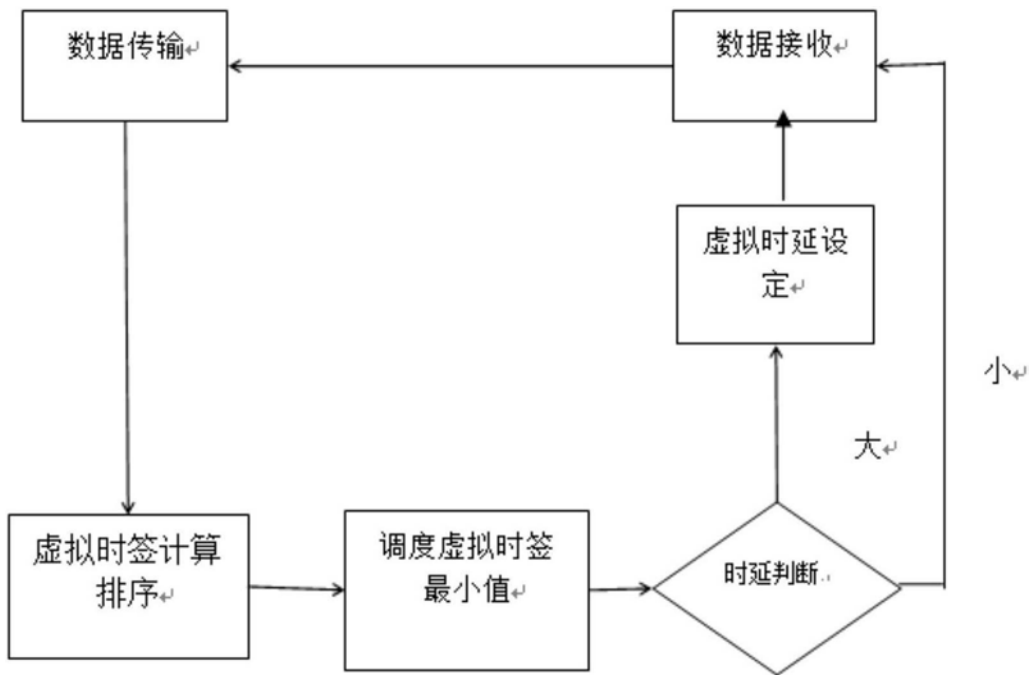


图2

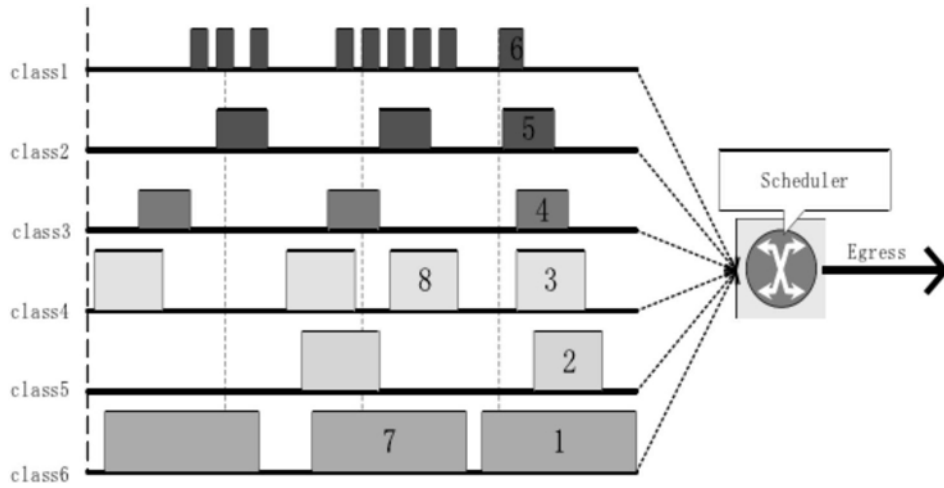


图3

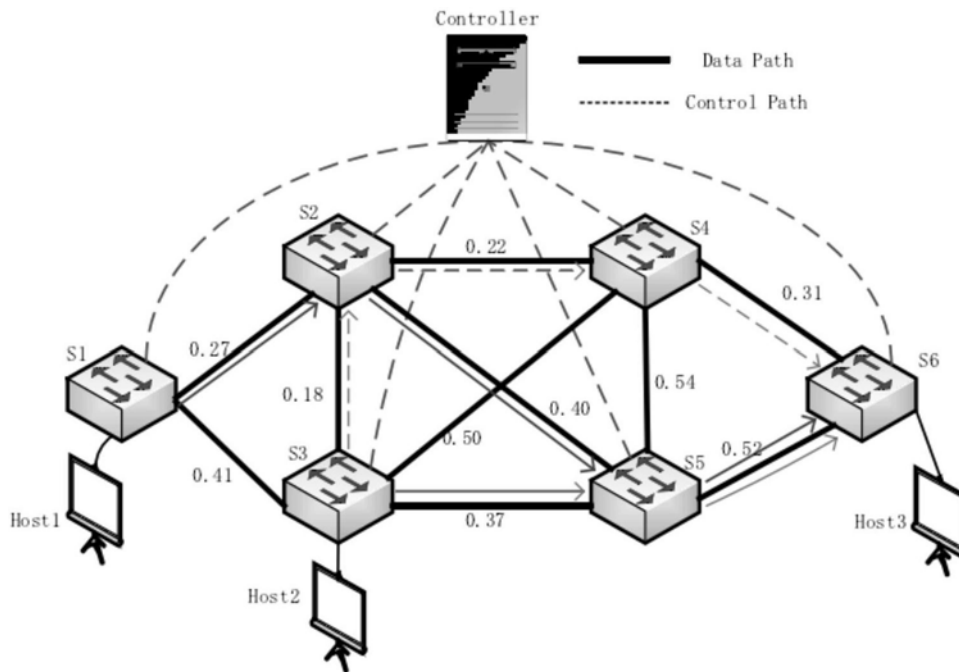


图4

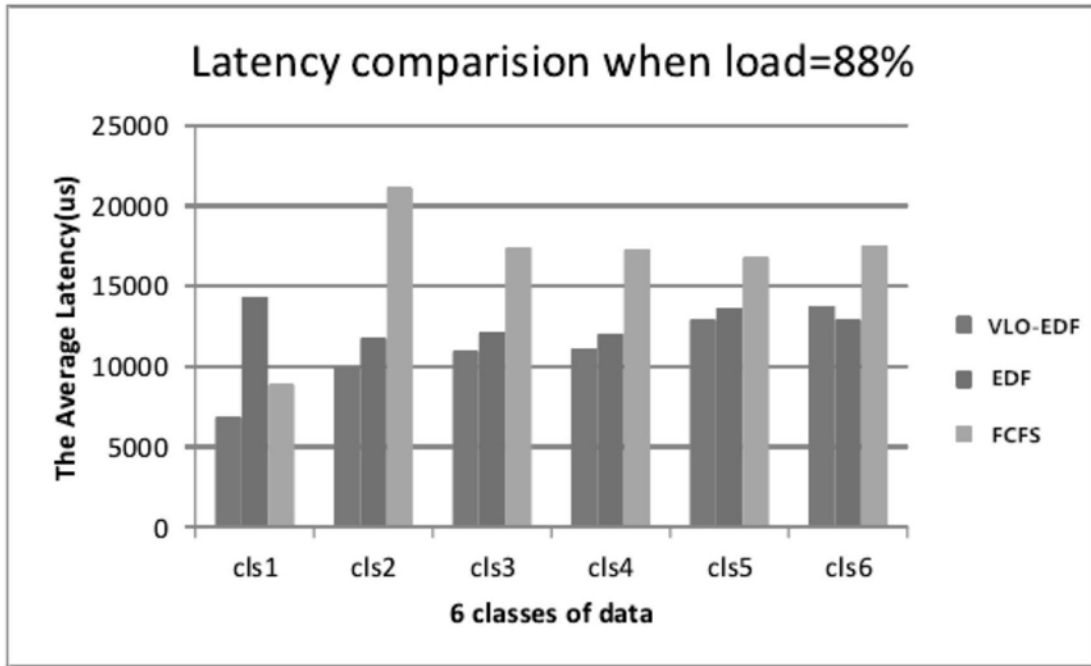


图5

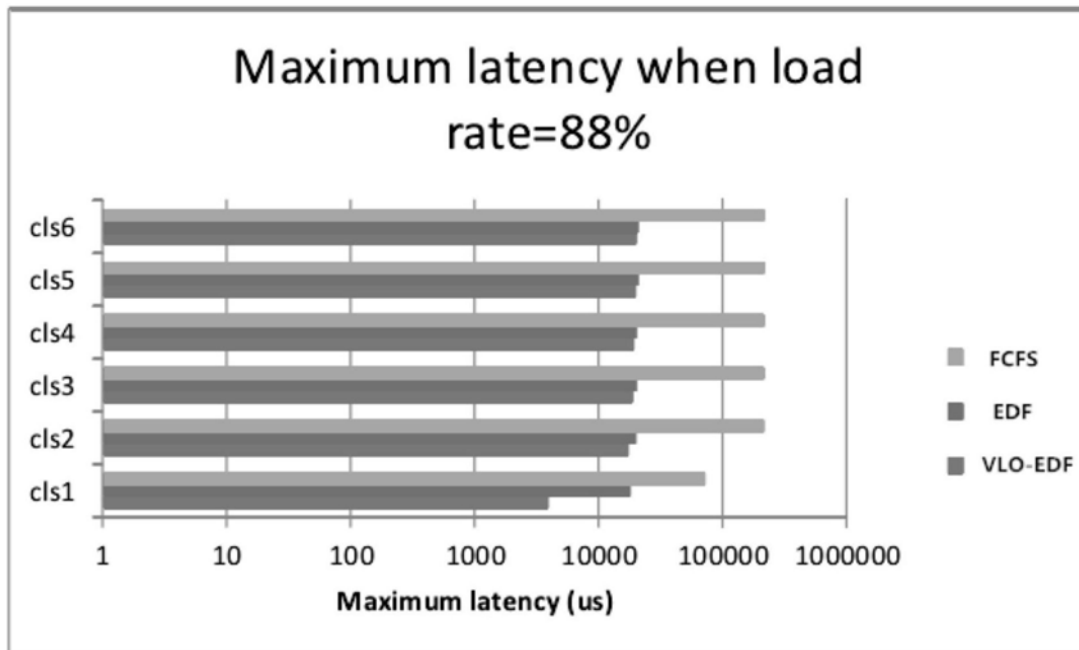


图6

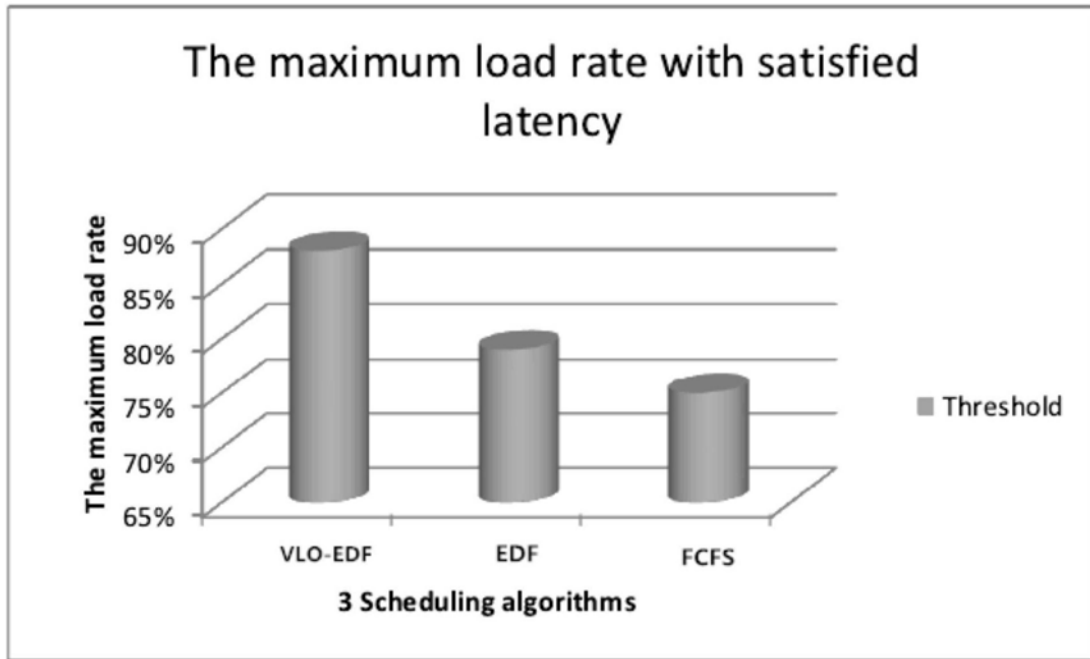


图7

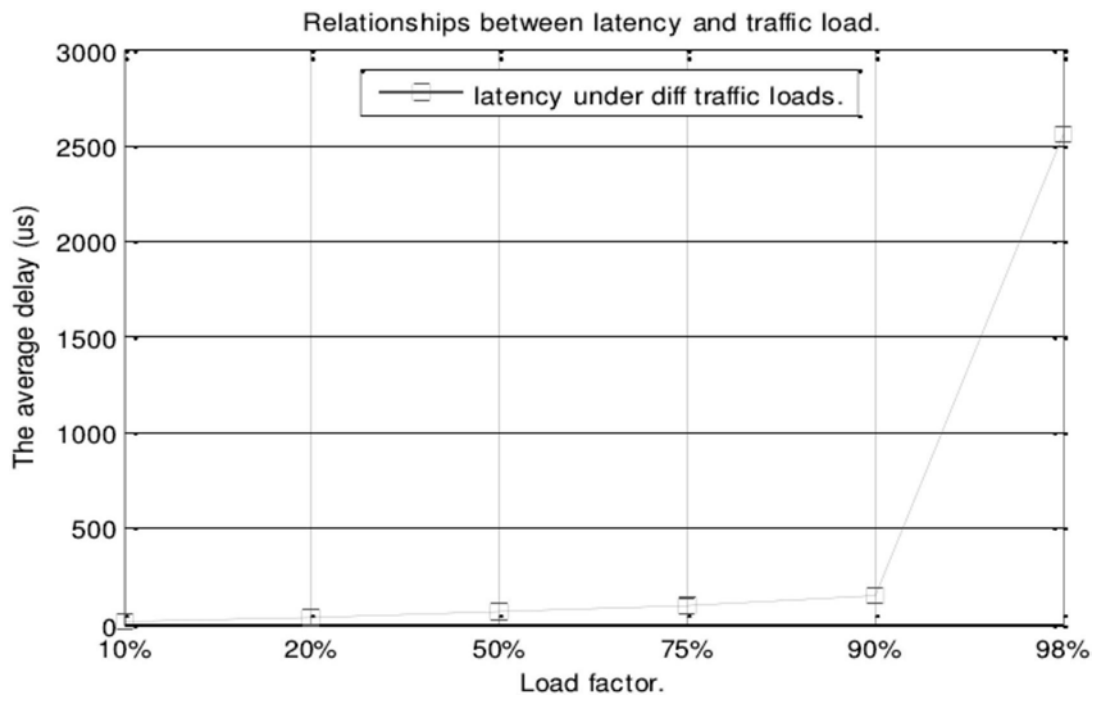


图8

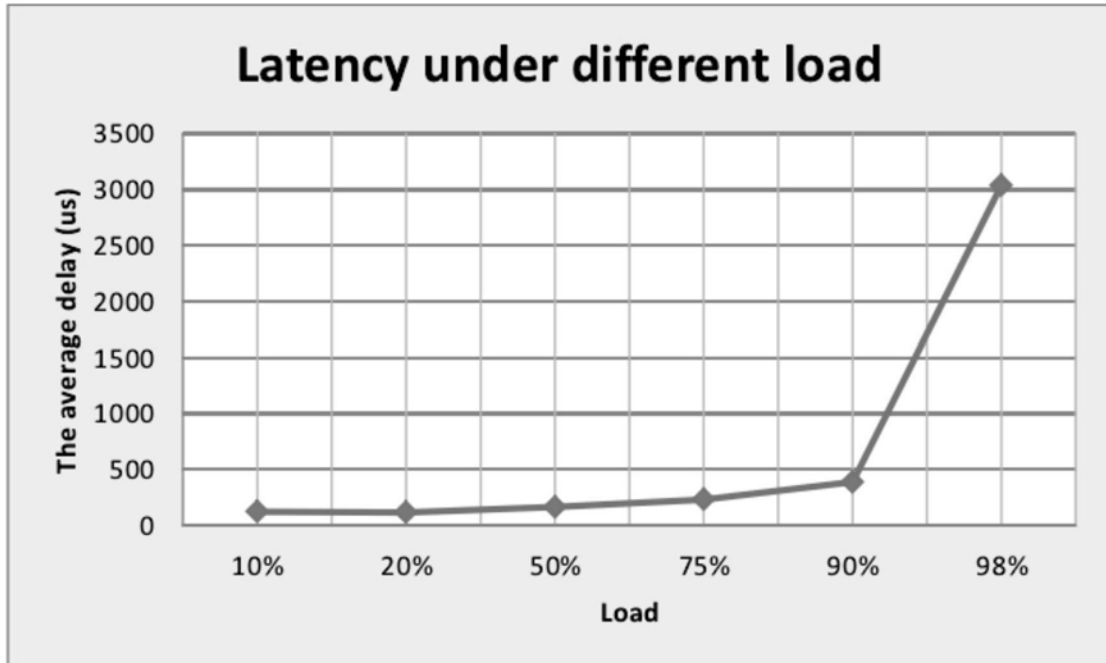


图9