



(12)发明专利

(10)授权公告号 CN 104683243 B

(45)授权公告日 2019.02.01

(21)申请号 201310611749.5

(22)申请日 2013.11.26

(65)同一申请的已公布的文献号  
申请公布号 CN 104683243 A

(43)申请公布日 2015.06.03

(73)专利权人 华为技术有限公司  
地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

(72)发明人 郭文 马朋

(74)专利代理机构 广州三环专利商标代理有限公司 44202

代理人 郝传鑫 熊永强

(51)Int.Cl.

H04L 12/757(2013.01)

H04L 1/18(2006.01)

(56)对比文件

WO 2013108761 A1,2013.07.25,  
CN 103023826 A,2013.04.03,  
CN 103155497 A,2013.06.12,  
CN 102859952 A,2013.01.02,  
CN 101741605 A,2010.06.16,  
OPEN NETWORKING FOUNDATION.openflow-spec-v1.3.2.《https://www.cs.princeton.edu/courses/archive/fall13/cos597E/papers/openflow-spec-v1.3.2.pdf》.2013,全文.

审查员 孙丽

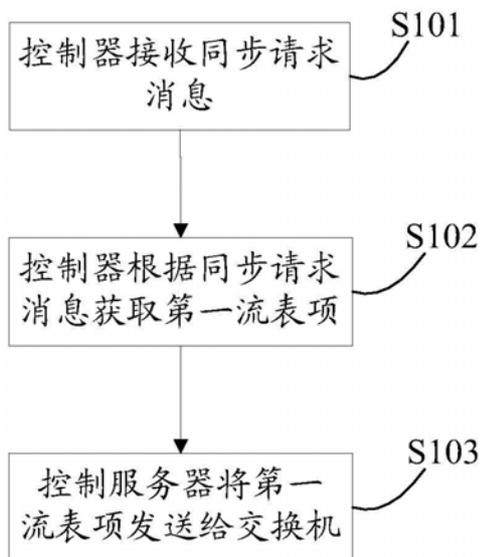
权利要求书4页 说明书16页 附图7页

(54)发明名称

一种数据处理方法、装置及系统

(57)摘要

本发明实施例公开了一种数据处理方法、装置及系统,其中方法包括:控制器接收同步请求消息,所述同步请求消息为交换机请求将所述控制器上的第一流表项同步到所述交换机上的消息,所述第一流表项是所述控制器在接收到所述同步请求消息前向所述交换机发送的流表项;所述控制器根据所述同步请求消息获取所述第一流表项;所述控制器将所述第一流表项发送给所述交换机。采用该方法及系统,可实现控制器和交换机之间的流表项同步,避免因交换机丢失流表项而造成控制器和交换机之间的流表项不同步的现象,增加了数据传输过程中的自动纠错功能,提高了网络设备信息交互的安全性。



1. 一种数据处理方法,其特征在于,包括:

控制器接收同步请求消息,所述同步请求消息为交换机请求将所述控制器上的第一流表项同步到所述交换机上的消息,所述第一流表项是所述控制器在接收到所述同步请求消息前向所述交换机发送的流表项;

所述控制器根据所述同步请求消息获取所述第一流表项;

所述控制器将所述第一流表项发送给所述交换机。

2. 如权利要求1所述的数据处理方法,其特征在于,所述同步请求消息中携带第一时间点;相应地,所述控制器根据所述同步请求消息获取所述第一流表项,包括:

所述控制器解析所述同步请求消息,获得所述第一时间点;

所述控制器将所述第一时间点前所述控制器向所述交换机发送的流表项确定为所述第一流表项;

所述控制器获取所述第一流表项。

3. 如权利要求1所述的数据处理方法,其特征在于,在所述控制器将所述第一流表项发送给所述交换机之前,还包括:

所述控制器向所述交换机发送同步开始消息,所述同步开始消息用于通知所述交换机所述控制器开始发送所述第一流表项;

在所述控制器将所述第一流表项发送给交换机之后,还包括:

所述控制器向所述交换机发送同步结束消息,所述同步结束消息用于通知所述交换机所述控制器结束发送所述第一流表项。

4. 如权利要求2所述的数据处理方法,其特征在于,在所述控制器将所述第一流表项发送给所述交换机之前,还包括:

所述控制器向所述交换机发送同步开始消息,所述同步开始消息用于通知所述交换机所述控制器开始发送所述第一流表项;

在所述控制器将所述第一流表项发送给交换机之后,还包括:

所述控制器向所述交换机发送同步结束消息,所述同步结束消息用于通知所述交换机所述控制器结束发送所述第一流表项。

5. 如权利要求1-4中任一项所述的数据处理方法,其特征在于,在所述控制器将所述第一流表项发送给所述交换机之前,还包括:

所述控制器对所述第一流表项设置标记,所述标记用于指示所述第一流表项为所述交换机所请求同步的流表项。

6. 一种数据处理方法,其特征在于,包括:

交换机向控制器发送同步请求消息,所述同步请求消息为所述交换机请求将所述控制器上的第一流表项同步到所述交换机上的消息,所述第一流表项是所述控制器在接收到所述同步请求消息前向所述交换机发送的流表项;

所述交换机接收所述控制器发送的所述第一流表项。

7. 如权利要求6所述的数据处理方法,其特征在于,所述同步请求消息中携带第一时间点,所述第一时间点用于指示所述控制器将所述第一时间点前所述控制器向所述交换机发送的流表项确定为所述第一流表项。

8. 如权利要求6所述的数据处理方法,其特征在于,所述交换机向控制器发送同步请求

消息,包括:

交换机对已存储的流表项进行检验;

根据检验结果向所述控制器发送所述同步请求消息。

9. 如权利要求7所述的数据处理方法,其特征在于,所述交换机向控制器发送同步请求消息,包括:

交换机对已存储的流表项进行检验;

根据检验结果向所述控制器发送所述同步请求消息。

10. 如权利要求8所述的数据处理方法,其特征在于,所述根据检验结果向所述控制器发送所述同步请求消息,包括:

当检验到所述已存储的流表项的序列号不连续或者所述已存储的流表项丢失时,向所述控制器发送所述同步请求消息。

11. 如权利要求9所述的数据处理方法,其特征在于,所述根据检验结果向所述控制器发送所述同步请求消息,包括:

当检验到所述已存储的流表项的序列号不连续或者所述已存储的流表项丢失时,向所述控制器发送所述同步请求消息。

12. 如权利要求6-11中任一项所述的数据处理方法,其特征在于,在所述交换机向所述控制器发送同步请求消息之后,以及所述交换机接收所述控制器发送的所述第一流表项之前,还包括:

所述交换机接收所述控制器发送的同步开始消息,所述同步开始消息用于通知所述交换机所述控制器开始发送所述第一流表项;

在所述交换机接收所述控制器发送的所述第一流表项之后,还包括:

所述交换机接收所述控制器发送的同步结束消息,所述同步结束消息用于通知所述交换机所述控制器结束发送所述第一流表项;

所述交换机根据所述同步开始消息和所述同步结束消息对所述第一流表项设置标记,所述标记用于指示所述第一流表项为所述交换机所请求同步的流表项。

13. 如权利要求6-11中任一项所述的数据处理方法,其特征在于,所述第一流表项携带标记,所述标记用于指示所述第一流表项为所述交换机所请求同步的流表项。

14. 一种控制器,其特征在于,包括:

消息接收模块,用于接收同步请求消息,所述同步请求消息为交换机请求将所述控制器上的第一流表项同步到所述交换机上的消息,所述第一流表项是所述控制器在接收到所述同步请求消息前向所述交换机发送的流表项;

流表项获取模块,用于根据所述同步请求消息获取所述第一流表项;

流表项发送模块,用于将所述第一流表项发送给所述交换机。

15. 如权利要求14所述的控制器,其特征在于,所述同步请求消息中携带第一时间点;所述流表项获取模块包括:

解析单元,用于解析所述同步请求消息,获得所述第一时间点;

流表项确定单元,用于将所述第一时间点前所述控制器已发送过的流表项确定为所述第一流表项;

流表项获取单元,用于获取所述第一流表项。

16. 如权利要求14所述的控制器,其特征在于,还包括:

同步开始消息发送模块,用于在所述流表项发送模块将所述第一流表项发送给所述交换机之前,向所述交换机发送同步开始消息,所述同步开始消息用于通知所述交换机所述控制器开始发送所述第一流表项;

同步结束消息发送模块,用于在所述流表项发送模块将所述第一流表项发送给所述交换机之后,向所述交换机发送同步结束消息,所述同步结束消息用于通知所述交换机所述控制器结束发送所述第一流表项。

17. 如权利要求15所述的控制器,其特征在于,还包括:

同步开始消息发送模块,用于在所述流表项发送模块将所述第一流表项发送给所述交换机之前,向所述交换机发送同步开始消息,所述同步开始消息用于通知所述交换机所述控制器开始发送所述第一流表项;

同步结束消息发送模块,用于在所述流表项发送模块将所述第一流表项发送给所述交换机之后,向所述交换机发送同步结束消息,所述同步结束消息用于通知所述交换机所述控制器结束发送所述第一流表项。

18. 如权利要求14-17中任一项所述的控制器,其特征在于,还包括:

第一标记模块,用于在所述流表项发送模块将所述第一流表项发送给所述交换机之前,对所述第一流表项设置标记,所述标记用于指示所述第一流表项为所述交换机所请求同步的流表项。

19. 一种交换机,其特征在于,包括:

消息发送模块,用于向控制器发送同步请求消息,所述同步请求消息为所述交换机请求将所述控制器上的第一流表项同步到所述交换机上的消息,所述第一流表项是所述控制器在接受到所述同步请求消息前向所述交换机发送的流表项;

流表项接收模块,用于接收所述控制器下发的所述第一流表项。

20. 如权利要求19所述的交换机,其特征在于,所述同步请求消息中携带第一时间点,所述第一时间点用于指示所述控制器将所述第一时间点前所述控制器向所述交换机发送的流表项确定为所述第一流表项。

21. 如权利要求19所述的交换机,其特征在于,所述消息发送模块包括:

流表项检验单元,用于对已存储的流表项进行检验;

消息发送单元,用于根据检验结果向所述控制器发送所述同步请求消息。

22. 如权利要求20所述的交换机,其特征在于,所述消息发送模块包括:

流表项检验单元,用于对已存储的流表项进行检验;

消息发送单元,用于根据检验结果向所述控制器发送所述同步请求消息。

23. 如权利要求21所述的交换机,其特征在于,所述消息发送单元包括:

消息发送子单元,用于当检验到所述已存储的流表项的序列号不连续或者所述已存储的流表项丢失时,向所述控制器发送所述同步请求消息。

24. 如权利要求22所述的交换机,其特征在于,所述消息发送单元包括:

消息发送子单元,用于当检验到所述已存储的流表项的序列号不连续或者所述已存储的流表项丢失时,向所述控制器发送所述同步请求消息。

25. 如权利要求19-24中任一项所述的交换机,其特征在于,还包括:

同步开始消息接收模块,用于在所述消息发送模块向所述控制器发送同步请求消息之后,以及所述流表项接收模块接收所述控制器下发的所述第一流表项之前,接收同步开始消息,所述同步开始消息用于通知所述交换机所述控制器开始发送所述第一流表项;

同步结束消息接收模块,用于在所述流表项接收模块接收所述控制器下发的所述第一流表项之后,接收同步结束消息,所述同步结束消息用于通知所述交换机所述控制器结束发送所述第一流表项;

第二标记模块,用于根据所述同步开始消息和所述同步结束消息对所述第一流表项设置标记,所述标记用于指示所述第一流表项为所述交换机所请求同步的流表项。

26. 如权利要求19-24中任一项所述的交换机,其特征在于,所述第一流表项携带标记,所述标记用于指示所述第一流表项为所述交换机所请求同步的流表项。

27. 一种数据处理系统,其特征在于,包括如权利要求14-18任一项所述的控制器和如权利要求19-26任一项所述的交换机。

## 一种数据处理方法、装置及系统

### 技术领域

[0001] 本发明涉及通信技术领域,尤其涉及一种数据处理方法、装置及系统。

### 背景技术

[0002] 随着互联网的迅猛发展和互联网新业务的需求不断增大,OpenFlow网络作为一种新型的网络交换模型,对网络的创新发展起到的巨大的推动作用使其越来越受关注和支持。

[0003] OpenFlow网络包括交换机、控制器Controller和网络虚拟层FlowVisor。其中,交换机是整个OpenFlow网络的核心部件,主要管理数据层的转发,其由流表、安全通道和OpenFlow协议三部分组成,交换机包括专用的OpenFlow交换机和支持OpenFlow的交换机。控制器实现控制层的功能,通过OpenFlow协议管理和控制OpenFlow交换机,安全通道是连接OpenFlow交换机到控制器的接口。FlowVisor是位于硬件结构元件和软件之间的网络虚拟层。

[0004] 当控制器通过现有的OpenFlow协议向交换机下发流表项后,交换机内部在处理流表项时,由于交换机内部处理OpenFlow协议报文的进程和处理流表项的进程不是同一进程,而进程间通信有可能出现数据丢包的现象,导致流表项丢失,从而使得交换机的流表项和控制器的流表项不同步。

### 发明内容

[0005] 本发明实施例所要解决的技术问题在于,提供一种数据处理方法、装置及系统,可实现控制器和交换机之间的流表项同步,增加流表项丢失后的自动纠错功能。

[0006] 为了解决上述技术问题,本发明第一方面提供了一种数据处理方法,可包括:

[0007] 控制器接收同步请求消息,所述同步请求消息为交换机请求将所述控制器上的第一流表项同步到所述交换机上的消息,所述第一流表项是所述控制器在接收到所述同步请求消息前向所述交换机发送的流表项;

[0008] 所述控制器根据所述同步请求消息获取所述第一流表项;

[0009] 所述控制器将所述第一流表项发送给所述交换机。

[0010] 基于第一方面,在第一方面的第一种可行的实施方式中,所述同步请求消息中携带第一时间点;相应地,所述控制器根据所述同步请求消息获取所述第一流表项,包括:

[0011] 所述控制器解析所述同步请求消息,获得所述第一时间点;

[0012] 所述控制器将所述第一时间点前所述控制器向所述交换机发送的流表项确定为所述第一流表项;

[0013] 所述控制器获取所述第一流表项。

[0014] 基于第一方面或第一方面的第一种可行的实施方式,在第一方面的第二种可行的实施方式中,在所述控制器将所述第一流表项发送给所述交换机之前,还包括:

[0015] 所述控制器向所述交换机发送同步开始消息,所述同步开始消息用于通知所述交

交换机所述控制器开始发送所述第一流表项；

[0016] 在所述控制器将所述第一流表项发送给交换机之后,还包括:

[0017] 所述控制器向所述交换机发送同步结束消息,所述同步结束消息用于通知所述交换机所述控制器结束发送所述第一流表项。

[0018] 基于第一方面或第一方面的第一种可行的实施方式第一方面的第二种可行的实施方式,在第一方面的第三种可行的实施方式中,在所述控制器将所述流表项发送给所述交换机之前,还包括:

[0019] 所述控制器对所述第一流表项设置标记,所述标记用于指示所述第一流表项为所述交换机所请求同步的流表项。

[0020] 本发明第二方面提供了一种数据处理方法,可包括:

[0021] 交换机向控制器发送同步请求消息,所述同步请求消息为所述交换机请求将所述控制器上的第一流表项同步到所述交换机上的消息,所述第一流表项是所述控制器在接受到所述同步请求消息前向所述交换机发送的流表项;

[0022] 所述交换机接收所述控制器发送的所述第一流表项。

[0023] 基于第二方面,在第二方面的第一种可行的实施方式中,所述同步请求消息中携带第一时间点,所述第一时间点用于指示所述控制器将所述第一时间点前所述控制器向所述交换机发送的流表项确定为所述第一流表项。

[0024] 基于第二方面或第二方面的第一种可行的实施方式,在第二方面的第二种可行的实施方式中,所述交换机向控制器发送同步请求消息,包括:

[0025] 交换机对已存储的流表项进行检验;

[0026] 根据检验结果向所述控制器发送所述同步请求消息。

[0027] 基于第二方面的第二种可行的实施方式,在第二方面的第三种可行的实施方式中,所述根据检验结果向所述控制器发送所述同步请求消息,包括:

[0028] 当检验到所述已存储的流表项的序列号不连续或者所述已存储的流表项丢失时,向所述控制器发送所述同步请求消息。

[0029] 基于第二方面或第二方面的第一种可行的实施方式或第二方面的第二种可行的实施方式或第二方面的第三种可行的实施方式,在第二方面的第四种可行的实施方式中,在所述交换机向所述控制器发送同步请求消息之后,以及所述交换机接收所述控制器发送的所述第一流表项之前,还包括:

[0030] 所述交换机接收所述控制器发送的同步开始消息,所述同步开始消息用于通知所述交换机所述控制器开始发送所述第一流表项;

[0031] 在所述交换机接收所述控制器发送的所述第一流表项之后,还包括:

[0032] 所述交换机接收所述控制器发送的同步结束消息,所述同步结束消息用于通知所述交换机所述控制器结束发送所述第一流表项;

[0033] 所述交换机根据所述同步开始消息和所述同步结束消息对所述第一流表项设置标记,所述标记用于指示所述第一流表项为所述交换机所请求同步的流表项。

[0034] 基于第二方面或第二方面的第一种可行的实施方式或第二方面的第二种可行的实施方式或第二方面的第三种可行的实施方式或第二方面的第四种可行的实施方式,在第二方面的第五种可行的实施方式中,所述第一流表项携带标记,所述标记用于指示所述第

一流表项为所述交换机所请求同步的流表项。

[0035] 本发明第三方面提供了一种控制器,可包括:

[0036] 消息接收模块,用于接收同步请求消息,所述同步请求消息为交换机请求将所述控制器上的第一流表项同步到所述交换机上的消息,所述第一流表项是所述控制器在接收到所述同步请求消息前向所述交换机发送的流表项;

[0037] 流表项获取模块,用于根据所述同步请求消息获取所述第一流表项;

[0038] 流表项发送模块,用于将所述第一流表项发送给所述交换机。

[0039] 基于第三方面,在第三方面的第一种可行的实施方式中,所述同步请求消息中携带第一时间点;所述流表项获取模块包括:

[0040] 解析单元,用于解析所述同步请求消息,获得所述第一时间点;

[0041] 流表项确定子单元,用于将所述第一时间点前所述控制器已发送过的流表项确定为所述第一流表项;

[0042] 流表项获取单元,用于获取所述第一流表项。

[0043] 基于第三方面或第三方面的第一种可行的实施方式,在第三方面的第二种可行的实施方式中,还包括:

[0044] 同步开始消息发送模块,用于在所述流表项发送模块将所述第一流表项发送给所述交换机之前,向所述交换机发送同步开始消息,所述同步开始消息用于通知所述交换机所述控制器开始发送所述第一流表项;

[0045] 同步结束消息发送模块,用于在所述流表项发送模块将所述第一流表项发送给所述交换机之后,向所述交换机发送同步结束消息,所述同步结束消息用于通知所述交换机所述控制器结束发送所述第一流表项。

[0046] 基于第三方面或第三方面的第一种可行的实施方式或第三方面的第二种可行的实施方式,在第三方面的第三种可行的实施方式中,还包括:

[0047] 第一标记模块,用于在所述流表项发送模块将所述第一流表项发送给所述交换机之前,对所述第一流表项设置标记,所述标记用于指示所述第一流表项为所述交换机所请求同步的流表项。

[0048] 本发明第四方面提供了一种交换机,可包括:

[0049] 消息发送模块,用于向控制器发送同步请求消息,所述同步请求消息为所述交换机请求将所述控制器上的第一流表项同步到所述交换机上的消息,所述第一流表项是所述控制器在接收到所述同步请求消息前向所述交换机发送的流表项;

[0050] 流表项接收模块,用于接收所述控制器发送的所述第一流表项。

[0051] 基于第四方面,在第四方面的第一种可行的实施方式中,所述同步请求消息中携带第一时间点,所述第一时间点用于指示所述控制器将所述第一时间点前所述控制器向所述交换机发送的流表项确定为所述第一流表项。

[0052] 基于第四方面或第四方面的第一种可行的实施方式,在第四方面的第二种可行的实施方式中,所述消息发送模块包括:

[0053] 流表项检验单元,用于对已存储的流表项进行检验;

[0054] 消息发送单元,用于根据检验结果向所述控制器发送所述同步请求消息。

[0055] 基于第四方面的第二种可行的实施方式,在第四方面的第三种可行的实施方式

中,所述消息发送单元包括:

[0056] 消息发送子单元,用于当检验到所述已存储的流表项的序列号不连续或者所述已存储的流表项丢失时,向所述控制器发送所述同步请求消息。

[0057] 基于第四方面或第四方面的第一种可行的实施方式或第四方面的第二种可行的实施方式或第四方面的第三种可行的实施方式,在第四方面的第四种可行的实施方式中,所述交换机还包括:

[0058] 同步开始消息接收模块,用于在所述消息发送模块向所述控制器发送同步请求消息之后,以及所述流表项接收模块接收所述控制器下发的所述第一流表项之前,接收同步开始消息,所述同步开始消息用于通知所述交换机所述控制器开始发送所述第一流表项;

[0059] 同步结束消息接收模块,用于在所述流表项接收模块接收所述控制器下发的所述第一流表项之后,接收同步结束消息,所述同步结束消息用于通知所述交换机所述控制器结束发送所述第一流表项;

[0060] 第二标记模块,用于根据所述同步开始消息和所述同步结束消息对所述第一流表项设置标记,所述标记用于指示所述第一流表项为所述交换机所请求同步的流表项。

[0061] 基于第四方面或第四方面的第一种可行的实施方式或第四方面的第二种可行的实施方式或第四方面的第三种可行的实施方式或第四方面的第四种可行的实施方式,在第四方面的第五种可行的实施方式中,所述第一流表项携带标记,所述标记用于指示所述第一流表项为所述交换机所请求同步的流表项。

[0062] 本发明第五方面提供了一种数据处理系统,包括如本发明第三方面所述的控制器和本发明第四方面所述的交换机。

[0063] 实施本发明实施例,具有如下有益效果:控制器通过接收交换机请求同步流表项的同步请求消息,将其请求的流表项重新发送给交换机,从而实现了控制器和交换机之间的流表项同步,避免因交换机丢失流表项而造成控制器和交换机之间的流表项不同步的现象,增加了数据传输过程中的自动纠错功能,提高了网络设备信息交互的安全性。

## 附图说明

[0064] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0065] 图1是本发明实施例提供的一种数据处理方法的流程示意图;

[0066] 图2是本发明实施例提供的另一种数据处理方法的流程示意图;

[0067] 图3是本发明实施例提供的又一种数据处理方法的流程示意图;

[0068] 图4是本发明实施例提供的再一种数据处理方法的流程示意图;

[0069] 图5是本发明实施例提供的一种控制器的结构示意图;

[0070] 图6是本发明实施例的流表项获取模块的结构示意图;

[0071] 图7是本发明实施例提供的一种交换机的结构示意图;

[0072] 图8是本发明实施例的消息发送模块的结构示意图;

[0073] 图9是是本发明实施例的消息发送单元的结构示意图;

[0074] 图10是本发明实施例提供的一种数据处理装置的结构示意图；

[0075] 图11是本发明实施例提供的另一种数据处理装置的结构示意图。

### 具体实施方式

[0076] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0077] 本发明实施例提供的数据处理方法主要应用于OpenFlow网络环境中,OpenFlow网络主要由交换机、控制器和网络虚拟层FlowVisor组成。其中,交换机接收到数据包后,首先在本地流表查找可转发的端口,若没有匹配的端口,则将数据包转发给控制器,由控制器决定转发端口。控制器通过OpenFlow协议向交换机下发流表项,交换机对OpenFlow协议报文进行解析处理后,通过接收到的流表项获取数据包的转发路径,由此确定目标端口。

[0078] 请参见图1,为本发明实施例提供的一种数据处理方法的流程示意图。本实施例从控制器一侧阐述数据处理方法的具体流程;所述方法可以包括以下步骤S101-步骤S103。

[0079] S101,控制器接收同步请求消息,所述同步请求消息为交换机请求将所述控制器上的第一流表项同步到所述交换机上的消息,所述第一流表项是所述控制器在接收到所述同步请求消息前向所述交换机发送的流表项。

[0080] 需要说明的是,交换机包括流表、安全通道和OpenFlow协议,其中,安全通道是连接交换机和控制器的接口,控制器通过这个接口控制和管理交换机。控制器通过安全通道接收来自交换机的同步请求消息。OpenFlow协议是用来描述控制器和交换机之间交互所用信息的标准,以及控制器和交换机的接口标准。同步请求消息是交换机检测到需要同步流表项的时候向控制器发送的请求消息,发送与接收同步请求消息需对OpenFlow协议进行扩展,举例来说同步请求消息的结构如下:

```

/* Smooth Request. */
struct ofp_switch_smooth_request {
[0081] struct ofp_header header;
};

```

[0082] 其中,OpenFlow协议消息头部的ofp\_header.ofp\_type表示消息的类型,其值为OFPT\_FLOW\_SMOOTH\_REQUEST,当控制器收到OFPT\_FLOW\_SMOOTH\_REQUEST消息时,接受交换机同步流表项的请求。以上举例中涉及的具体的名称仅为示例。

[0083] 举例来说,控制器可将接收到同步请求消息之前收到的流表项均确定为交换机所请求同步的第一流表项,目的在于:交换机在发送同步请求消息后到控制器接收到同步请求消息之前的这段时间内,交换机一端若再次出现其他需要同步的流表项,则无需再次发送同步请求消息,因为控制器重新发送的第一流表项已包含了控制器接收到同步请求消息之前存储的所有(包括最新)历史数据。又举例来说,控制器将从接收到同步请求消息之前向交换机发送的所有的流表项中的一部分流表项确定为交换机所请求同步的第一流表项。

[0084] S102,控制器根据同步请求消息获取第一流表项;

[0085] 控制器根据同步请求消息,确定接收到同步请求消息前的流表项为交换机请求交换的第一流表项,获取第一流表项,并根据OpenFlow协议进行打包处理(可能产生多个数据包)。

[0086] S103,控制器将第一流表项发送给交换机。

[0087] 控制器通过安全通道将获取并打包后的第一流表项发送给交换机,交换机重新接收流表项,以此实现控制器和交换机的流表项同步。

[0088] 通过本发明实施例,控制器通过接收交换机请求同步流表项的同步请求消息,将其请求的流表项重新发送给交换机,从而实现了控制器和交换机之间的流表项同步,避免因交换机丢失流表项而造成控制器和交换机之间的流表项不同步的现象,增加了数据传输过程中的自动纠错功能,提高了网络设备信息交互的安全性。

[0089] 请参见图2,为本发明实施例提供了另一种数据处理方法的流程示意图。本实施例从控制器一侧阐述数据处理方法的具体流程;所述方法可以包括以下步骤S201-步骤S208。

[0090] S201,控制器接收同步请求消息,所述同步请求消息为交换机请求将所述控制器上的第一流表项同步到所述交换机上的消息,所述第一流表项是所述控制器在接收到所述同步请求消息前向所述交换机发送的流表项。

[0091] 具体执行方法,请参考图1所示的实施例中的步骤S101中描述,在此不再赘述。

[0092] 控制器接收同步请求消息后,控制器根据同步请求消息获取第一流表项,具体的,可包括步骤S202-步骤S203。

[0093] S202,控制器解析同步请求消息,获得所述第一时间点。

[0094] 需要说明的是,同步请求消息中携带第一时间点,当控制器接收到来自交换机的同步请求消息时,对携带该同步请求消息的报文进行解析处理,获得其所携带的第一时间点。举例来说,第一时间点可以定义为交换机发出同步请求消息的时间点往前推移某个时间段的时间点。

[0095] S203,控制器将第一时间点前控制器向所述交换机发送的流表项确定为第一流表项。

[0096] 控制器获得第一时间点后,将第一时间点前控制器已发送过的流表项确定为第一流表项。

[0097] 在图2所示的实施例中,通过在同步请求消息中获得第一时间点,相当于设置了时间维度的过滤条件。举例来说,同步请求消息中除了携带第一时间点,还可以携带另一个时间点,以表示时间维度的过滤条件为“在另一个时间点之后到第一时间点之前”,即将另一个时间点后且第一时间点前控制器向所述交换机发送的流表项确定为第一流表项。又举例来说,同步请求消息中的时间维度的过滤条件,还可以包括多个不连续的时间区间。除了时间维度的过滤掉条件,交换机发送的同步请求消息中还可以携带其他维度的过滤条件,如需要同步的流表项的具体标识(如流表项序列号等),控制器根据该标识确定对应的流表项为第一流表项。

[0098] 在本实施例中,通过在同步请求消息中设置过滤条件,由此可避免发送无需同步的流表项,节省对安全通道的资源占用和减少控制器、交换机的数据处理负担。

[0099] S204,控制器获取第一流表项。

[0100] 具体的,控制器获取根据第一时间点确定的第一流表项,即获取第一时间点前控

制器已发送过的流表项,并对所有的流表项进行打包,准备再次发送给交换机。

[0101] S205,控制器向交换机发送同步开始消息,同步开始消息用于通知交换机控制器开始发送第一流表项。

[0102] 控制器将流表项发送给交换机之前,还向交换机发送同步开始消息,同步开始消息用于通知交换机控制器开始发送第一流表项。具体的,若要实现同步开始消息的发送与接收,同样需对OpenFlow协议进行扩展,具体的,同步开始消息的结构如下:

```

/* Smooth Start. */
struct ofp_switch_smooth_start {
[0103] struct ofp_header header;
};

```

[0104] 其中,OpenFlow协议消息头部的ofp\_header.ofp\_type表示消息的类型,其值为OFPT\_FLOW\_SMOOTH\_START,当控制器向交换机发送OFPT\_FLOW\_SMOOTH\_START消息时,表示开始发送交换机所请求的流表项。

[0105] S206,控制器对第一流表项设置标记,该标记用于指示第一流表项为所述交换机请求同步的流表项。

[0106] 需要说明的是,控制器对流表项设置标记,该标记用于指示被标记的流表项为交换机请求同步的流表项,即表示这些流表项是控制器最新下发的,而对于交换机中没有同步标记的流表项,交换机则可以优先删除。在其他实施方式中,由于控制器向交换机发送了同步开始消息,因此在交换机一侧,也可以由交换机对接收到同步开始消息后接收到的流表项设置标记,以表示这些流表项是控制器最新下发的,而无需由控制器标记流表项,即控制器不需要执行S206,对于交换机中没有同步标记的流表项,交换机则可以优先删除。

[0107] S207,控制器将第一流表项发送给所述交换机。

[0108] 具体的,控制器通过安全通道将获取并打包后的第一流表项发送给交换机,交换机重新接收流表项,以此实现控制器和交换机的流表项同步。

[0109] S208,控制器向交换机发送同步结束消息,所述同步结束消息用于通知所述交换机所述控制器结束发送请求同步的流表项。

[0110] 控制器将流表项发送完毕后,向交换机发送同步结束消息,同步结束消息用于通知交换机控制器结束发送请求同步的流表项。若要实现同步结束消息的发送与接收,同样需对OpenFlow协议进行扩展,具体的,同步结束消息的结构如下:

```

/* Smooth Finish. */
struct ofp_switch_smooth_finish {
[0111] struct ofp_header header;
};

```

[0112] 其中,OpenFlow协议消息头部的ofp\_header.ofp\_type表示消息的类型,其值为OFPT\_FLOW\_SMOOTH\_FINISH,当控制器向交换机发送OFPT\_FLOW\_SMOOTH\_FINISH消息时,表示交换机所请求的流表项已发送完毕。

[0113] 通过本发明实施例,控制器通过接收交换机请求同步流表项的同步请求消息,将其请求的流表项重新发送给交换机,从而实现了控制器和交换机之间的流表项同步,避免因交换机丢失流表项而造成控制器和交换机之间的流表项不同步的现象,增加了数据传输过程中的自动纠错功能,提高了网络设备信息交互的安全性。

[0114] 请参见图3,为本发明实施例提供的又一种数据处理方法的流程示意图。本实施例从交换机一侧阐述数据处理方法的具体流程;所述方法可以包括以下步骤S301-步骤S302。

[0115] S301,交换机向控制器发送同步请求消息,所述同步请求消息为所述交换机请求将所述控制器上的第一流表项同步到所述交换机上的消息,所述第一流表项是所述控制器在接受到所述同步请求消息前向所述交换机发送的流表项。

[0116] 举例来说,当交换机检测到本地存储的流表项需要同步(如流表项序列号不连续)时,交换机需要控制器重新下发流表项以使得两者之间的流表项同步,通过安全通道向控制器发送同步请求消息。同步请求消息是交换机检测到需要同步流表项的时候向控制器发送的请求消息,发送与接收同步请求消息需对OpenFlow协议进行扩展,同步请求消息的具体实现请参考图1所示的实施例中的步骤S101中描述,在此不再赘述。

[0117] 举例来说,控制器可将接收到同步请求消息之前收到的流表项均确定为交换机所请求同步的第一流表项,目的在于:交换机在发送同步请求消息后到控制器接收到同步请求消息之前的这段时间内,交换机一端若再次出现其他需要同步的流表项,则无需再次发送同步请求消息,因为控制器重新发送的第一流表项已包含了控制器接收到同步请求消息之前存储的所有(包括最新)历史数据。又举例来说,控制器将从接收到同步请求消息之前向交换机发送的所有的流表项中的一部分流表项确定为交换机所请求同步的第一流表项。

[0118] S302,交换机接收控制器发送的第一流表项。

[0119] 交换机接收控制器重新打包并下发的流表项,完成流表项同步。

[0120] 请参见图4,为本发明实施例提供的再一种数据处理方法的流程示意图。本实施例从交换机一侧阐述数据处理方法的具体流程;所述方法可以包括以下步骤S401-步骤S407。

[0121] S401,交换机对已存储的流表项进行检验。

[0122] 交换机对本地已存储的流表项进行检验,在本发明一些实施例中,可对每个流表项各自对应的序列号进行检验,具体的,可对对流表项的序列号连续性进行检验。

[0123] S402,根据检验结果向控制器发送同步请求消息。

[0124] 具体的,当检验到已存储的流表项的序列号不连续或者已存储的流表项丢失时,向控制器发送同步请求消息。

[0125] 需要说明的是,造成交换机内部存储的流表项序列号不连续或流表项丢失等情况的原因通常在于,交换机重启时流表项全部丢失,或者由于交换机中内部处理OpenFlow协议报文的进程和用于处理流表项的FES(Forward EntryService)进程不一致,进程间通信可能发生数据丢包的情况。

[0126] 当交换机检测到本地存储的流表项序号不连续时,交换机需要控制器重新下发流表项以使得两者之间的流表项同步,通过安全通道向控制器发送同步请求消息,同步请求消息为交换机请求同步流表项的消息。同步请求消息是交换机检测到需要同步流表项的时候向控制器发送的请求消息,发送与接收同步请求消息需对OpenFlow协议进行扩展,同步请求消息的具体实现请参考图1所示的实施例中的步骤S101中描述,在此不再赘述。

[0127] 在本发明一些实施例中,同步请求消息中携带第一时间点,第一时间点用于指示所述控制器将所述第一时间点前所述控制器向所述交换机发送的流表项确定为所述第一流表项。举例来说,第一时间点可以定义为交换机发出同步请求消息的时间点往前推移某个时间段的时间点。

[0128] S403,交换机接收控制器发送的同步开始消息,同步开始消息用于通知交换机所述控制器开始发送第一流表项。

[0129] 需要说明的是,同步开始消息为对OpenFlow协议进行扩展后控制器向交换机发送的消息,该消息用于通知交换机控制器开始发送请求同步的流表项。同步开始消息的结构如下:

```
/* Smooth Start. */  
  
[0130] struct ofp_switch_smooth_start {  
    struct ofp_header header;  
};
```

[0131] S404,交换机接收控制器发送的第一流表项。

[0132] 具体的,交换机通过安全通道接收控制器重新打包并下发的第一流表项。

[0133] S405,交换机接收控制器发送的同步结束消息,同步结束消息用于通知交换机控制器结束发送流表项。

[0134] 需要说明的是,同步结束消息为对OpenFlow协议进行扩展后控制器向交换机发送的消息,该消息用于通知交换机控制器结束发送流表项。同步结束消息的结构如下:

```
/* Smooth Finish. */  
  
[0135] struct ofp_switch_smooth_finish {  
    struct ofp_header header;  
};
```

[0136] 其中,OpenFlow协议消息头部的ofp\_header.ofp\_type表示消息的类型,其值为OFPT\_FLOW\_SMOOTH\_FINISH,当控制器向交换机发送OFPT\_FLOW\_SMOOTH\_FINISH消息时,表示交换机所请求的流表项已发送完毕。

[0137] 进一步的,交换机可根据同步开始消息和同步结束消息对第一流表项设置标记,标记用于指示所述第一流表项为所述交换机所请求同步的流表项。

[0138] 在本发明一些实施例中,同步开始消息携带第二时间点,第二时间点为交换机接收到同步开始消息的时间点,同步结束消息携带第三时间点,第三时间点为交换机接收到同步结束消息的时间点。交换机可以在接收第一流表项的过程中,对第二时间点之后以及第三时间点之前接收到的第一流表项逐条设置标记,以表示这些流表项是控制器最新下发的,用于区别本地早已存在的流表项和重新接收的流表项。对于交换机中没有同步标记的流表项,交换机可以优先删除。

[0139] 当然,在其他实施方式中,交换机可对第二时间点和第三时间点进行保存,在接收第一流表项的过程中不对第一流表项设置标记,而在第一流表项接收完毕后,才根据保存

的第二时间点和第三时间点对这两个时间点内接收的第一流表项进行批量的同步标记。

[0140] 在本发明一些实施例中,控制器发送的第一流表项携带标记,标记用于指示所述第一流表项为所述交换机所请求同步的流表项,即控制器已对第一流表项进行同步标记,因此交换机无需自己设置标记即可获知第一流表项为其所请求的流表项。

[0141] 通过本发明实施例,控制器通过接收交换机请求同步流表项的同步请求消息,将其请求的流表项重新发送给交换机,从而实现了控制器和交换机之间的流表项同步,避免因交换机丢失流表项而造成控制器和交换机之间的流表项不同步的现象,增加了数据传输过程中的自动纠错功能,提高了网络设备信息交互的安全性。

[0142] 请一并参见图5,为本发明实施例提供的一种控制器的结构示意图,包括消息接收模块51、流表项获取模块52、流表项发送模块53、同步开始消息发送模块54、同步结束消息发送模块55和第一标记模块56,各模块工作过程具体如下。

[0143] 消息接收模块51,用于接收同步请求消息,所述同步请求消息为交换机请求将所述控制器上的第一流表项同步到所述交换机上的消息,所述第一流表项是所述控制器在接收到所述同步请求消息前向所述交换机发送的流表项。

[0144] 需要说明的是,交换机包括流表、安全通道和OpenFlow协议,其中,安全通道是连接交换机和控制器的接口,控制器的消息接收模块51通过这个接口控制和管理交换机,即消息接收模块51通过安全通道接收来自交换机的同步请求消息。OpenFlow协议用来描述控制器和交换机之间交互所用信息的标准,以及控制器和交换机的接口标准。同步请求消息是交换机检测到需要同步流表项的时候向控制器发送的请求消息,若要完成同步请求消息的发送与接收,需对OpenFlow协议进行扩展,具体的,同步请求消息的结构如下:

```

/* Smooth Request. */
struct ofp_switch_smooth_request {
[0145] struct ofp_header header;
};

```

[0146] 其中,OpenFlow协议消息头部的ofp\_header.ofp\_type表示消息的类型,其值为OFPT\_FLOW\_SMOOTH\_REQUEST,当控制器收到OFPT\_FLOW\_SMOOTH\_REQUEST消息时,接受交换机同步流表项的请求。

[0147] 请参见图6,为本发明实施例的流表项获取模块52的结构示意图,流表项获取模块52用于根据所述同步请求消息获取所述第一流表项;具体的,流表项获取模块52包括解析单元521、流表项确定单元522和流表项获取单元523。

[0148] 其中,解析单元521,用于解析同步请求消息,获得第一时间点。

[0149] 需要说明的是,同步请求消息中携带第一时间点,当消息接收模块51接收到来自交换机的同步请求消息时,解析单元521对携带该同步请求消息的报文进行解析处理,获得其所携带的第一时间点。举例来说,第一时间点可以定义为交换机发出同步请求消息的时间点往前推移某个时间段的时间点。

[0150] 其中,流表项确定单元522,用于将所述第一时间点前所述控制器已发送过的流表项确定为第一流表项。

[0151] 需要说明的是,解析单元521获得第一时间点后,流表项确定单元522将第一时间

点前控制器已发送过的流表项确定为第一流表项。

[0152] 在本发明实施实施例中,通过在同步请求消息中第一时间点,相当于设置了时间维度的过滤条件。举例来说,同步请求消息中除了携带第一时间点,还可以携带另一个时间点,以表示时间维度的过滤条件为“在另一个时间点之后到第一时间点之前”,即将另一个时间点后且第一时间点前控制器向所述交换机发送的流表项确定为第一流表项。又举例来说,同步请求消息中的时间维度的过滤条件,还可以包括多个不连续的时间区间。除了时间维度的过滤掉条件,交换机发送的同步请求消息中还可以携带其他维度的过滤条件,如需要同步的流表项的具体标识(如流表项序列号等),控制器根据该标识确定对应的流表项为第一流表项。

[0153] 在本实施例中,通过在同步请求消息中设置过滤条件,由此可避免发送无需同步的流表项,节省对安全通道的资源占用和减少控制器、交换机的数据处理负担。

[0154] 其中,流表项获取单元523,用于获取所述第一流表项。

[0155] 具体的,流表项获取单元523获取根据第一时间点确定的第一流表项,即获取第一时间点前控制器已发送过的流表项,并对所有的流表项进行打包,准备再次发送给交换机。

[0156] 流表项发送模块53,用于将第一流表项发送给所述交换机。

[0157] 具体的,流表项发送模块53通过安全通道将获取并打包后的第一流表项发送给交换机,交换机重新接收流表项,以此实现控制器和交换机的流表项同步。

[0158] 同步开始消息发送模块54,用于在所述流表项发送模块523将所述第一流表项发送给所述交换机之前,向交换机发送同步开始消息,所述同步开始消息用于通知所述交换机所述控制器开始发送第一流表项。

[0159] 需要说明的是,在流表项发送模块53将流表项发送给交换机之前,同步开始消息发送模块54还向交换机发送同步开始消息,同步开始消息用于通知交换机控制器开始发送第一流表项。具体的,若要实现同步开始消息的发送与接收,同样需对OpenFlow协议进行扩展,具体的,同步开始消息的结构如下:

```

/* Smooth Start. */
struct ofp_switch_smooth_start {
[0160] struct ofp_header header;
};

```

[0161] 其中,OpenFlow协议消息头部的ofp\_header.ofp\_type表示消息的类型,其值为OFPT\_FLOW\_SMOOTH\_START,当控制器向交换机发送OFPT\_FLOW\_SMOOTH\_START消息时,表示开始发送交换机所请求的流表项。

[0162] 同步结束消息发送模块55,用于在所述流表项发送模块523将所述第一流表项发送给所述交换机之后,向交换机发送同步结束消息,所述同步结束消息用于通知所述交换机所述控制器结束发送第一流表项。

[0163] 需要说明的是,流表项发送模块53将流表项发送完毕后,同步结束消息发送模块55向交换机发送同步结束消息,同步结束消息用于通知交换机控制器结束发送请求同步的流表项。若要实现同步结束消息的发送与接收,同样需对OpenFlow协议进行扩展,具体的,同步结束消息的结构如下:

```
/* Smooth Finish. */  
[0164] struct ofp_switch_smooth_finish {  
    struct ofp_header header;  
};
```

[0165] 其中,OpenFlow协议消息头部的ofp\_header.ofp\_type表示消息的类型,其值为OFPT\_FLOW\_SMOOTH\_FINISH,当控制器向交换机发送OFPT\_FLOW\_SMOOTH\_FINISH消息时,表示交换机所请求的流表项已发送完毕。

[0166] 第一标记模块56,用于在所述流表项发送模块523将所述第一流表项发送给所述交换机之前,对第一流表项设置标记,所述标记用于指示第一流表项为所述交换机请求同步的流表项。

[0167] 需要说明的是,在流表项发送模块53将流表项发送给交换机之前,第一标记模块56还可对即将发送的流表项设置标记,该标记用于指示被标记的流表项为交换机请求同步的流表项,即表示这些流表项是控制器最新下发的,而对于交换机中没有同步标记的流表项,交换机则可以优先删除。在其他实施方式中,由于控制器向交换机发送了同步开始消息,因此在交换机一侧,也可以由交换机对接收到同步开始消息后接收到的流表项设置标记,以表示这些流表项是控制器最新下发的,而无需由控制器标记流表项,对于交换机中没有同步标记的流表项,交换机则可以优先删除。

[0168] 通过本发明实施例,控制器通过接收交换机请求同步流表项的同步请求消息,将其请求的流表项重新发送给交换机,从而实现了控制器和交换机之间的流表项同步,避免因交换机丢失流表项而造成控制器和交换机之间的流表项不同步的现象,增加了数据传输过程中的自动纠错功能,提高了网络设备信息交互的安全性。

[0169] 请参见图7,为本发明实施例提供的一种交换机的结构示意图,该交换机包括消息发送模块61、同步开始消息接收模块62、流表项接收模块63、同步结束消息接收模块64和第二标记模块65,各模块工作过程具体如下:

[0170] 消息发送模块61用于向控制器发送同步请求消息,所述同步请求消息为所述交换机请求将所述控制器上的第一流表项同步到所述交换机上的消息,所述第一流表项是所述控制器在接收到所述同步请求消息前向所述交换机发送的流表项。

[0171] 需要说明的是,控制器可将接收到同步请求消息之前收到的流表项均确定为交换机所请求同步的第一流表项,目的在于:交换机在发送同步请求消息后到控制器接收到同步请求消息之前的这段时间内,交换机一端若再次出现其他需要同步的流表项,则无需再次发送同步请求消息,因为控制器重新发送的第一流表项已包含了控制器接收到同步请求消息之前存储的所有(包括最新)历史数据。

[0172] 具体的,请参见图8,为本发明实施例的消息发送模块61的结构示意图。消息发送模块61包括流表项检验单元611和消息发送单元612。

[0173] 其中,流表项检验单元611用于对已存储的流表项进行检验。

[0174] 需要说明的是,流表项检验单元611对本地已存储的流表项进行检验,在本发明一些实施例中,可对每个流表项各自对应的序列号进行检验,具体的,可对对流表项的序列号连续性进行检验。

[0175] 请参见图9,为本发明实施例的消息发送单元612的结构示意图。消息发送单元612用于根据检验结果向控制器发送同步请求消息。具体的,消息发送单元612包括消息发送子单元6121。

[0176] 当检验到已存储的流表项的序列号不连续或者所述已存储的流表项丢失时,消息发送子单元6121向控制器发送同步请求消息。

[0177] 需要说明的是,造成交换机内部存储的流表项序列号不连续或流表项丢失等情况的原因通常在于,交换机重启时流表项全部丢失,或者由于交换机中内部处理OpenFlow协议报文的进程和用于处理流表项的FES(Forward Entry Service)进程不一致,进程间通信可能发生数据丢包的情况。

[0178] 当消息发送子单元6121检测到本地存储的流表项序号不连续时,交换机需要控制器重新下发流表项以使得两者之间的流表项同步,通过安全通道向控制器发送同步请求消息,同步请求消息为交换机请求同步流表项的消息。同步请求消息是交换机检测到需要同步流表项的时候向控制器发送的请求消息,发送与接收同步请求消息需对OpenFlow协议进行扩展,同步请求消息的具体实现请参考图1所示的实施例中的步骤S101中描述,在此不再赘述。

[0179] 在本发明一些实施例中,同步请求消息中携带第一时间点,第一时间点用于指示所述控制器将所述第一时间点前所述控制器向所述交换机发送的流表项确定为所述第一流表项。举例来说,第一时间点可以定义为交换机发出同步请求消息的时间点往前推移某个时间段的时间点。

[0180] 同步开始消息接收模块62,用于在所述消息发送模块61向所述控制器发送同步请求消息之后,以及所述流表项接收模块63接收所述控制器下发的所述第一流表项之前,接收控制器发送的同步开始消息,所述同步开始消息用于通知所述交换机所述控制器开始发送第一流表项。

[0181] 需要说明的是,同步开始消息为对OpenFlow协议进行扩展后控制器向交换机发送的消息,该消息用于通知交换机控制器开始发送请求同步的流表项。同步开始消息的结构如下:

```
/* Smooth Start. */  
struct ofp_switch_smooth_start {  
[0182] struct ofp_header header;  
};
```

[0183] 流表项接收模块63,用于接收控制器发送的第一流表项。

[0184] 具体的,流表项接收模块63通过安全通道接收控制器打包并下发的第一流表项。

[0185] 同步结束消息接收模块64,用于在所述流表项接收模块63接收所述控制器下发的所述第一流表项之后,接收控制器发送的同步结束消息,所述同步结束消息用于通知所述交换机所述控制器结束发送所述第一流表项。

[0186] 需要说明的是,同步结束消息为对OpenFlow协议进行扩展后控制器向交换机发送的消息,该消息用于通知交换机控制器结束发送流表项。同步结束消息的结构如下:

```
/* Smooth Finish. */  
[0187] struct ofp_switch_smooth_finish {  
    struct ofp_header header;  
};
```

[0188] 其中,OpenFlow协议消息头部的ofp\_header.ofp\_type表示消息的类型,其值为OFPT\_FLOW\_SMOOTH\_FINISH,当控制器向交换机发送OFPT\_FLOW\_SMOOTH\_FINISH消息时,表示交换机所请求的流表项已发送完毕。

[0189] 第二标记模块65,用于根据所述同步开始消息和所述同步结束消息对所述第一流表项设置标记,所述标记用于指示所述第一流表项为所述交换机所请求同步的流表项。在本发明实施例中,同步开始消息携带第二时间点,第二时间点为交换机接收到同步开始消息的时间点,同步结束消息携带第三时间点,第三时间点为交换机接收到同步结束消息的时间点。交换机可以在接收第一流表项的过程中,对第二时间点之后以及第三时间点之前接收到的第一流表项逐条设置标记,以表示这些流表项是控制器最新下发的,用于区别本地早已存在的流表项和重新接收的流表项。对于交换机中没有同步标记的流表项,交换机可以优先删除。

[0190] 当然,在其他实施方式中,交换机可对第二时间点和第三时间点进行保存,在接收第一流表项的过程中不对第一流表项设置标记,而在第一流表项接收完毕后,才根据保存的第二时间点和第三时间点对这两个时间点内接收的第一流表项进行批量的同步标记。

[0191] 在本发明一些实施例中,控制器发送的第一流表项携带标记,标记用于指示所述第一流表项为所述交换机所请求同步的流表项,即控制器已对第一流表项进行同步标记,因此交换机无需自己设置标记即可获知第一流表项为其所请求的流表项。

[0192] 通过本发明实施例,控制器通过接收交换机请求同步流表项的同步请求消息,将其请求的流表项重新发送给交换机,从而实现了控制器和交换机之间的流表项同步,避免因交换机丢失流表项而造成控制器和交换机之间的流表项不同步的现象,增加了数据传输过程中的自动纠错功能,提高了网络设备信息交互的安全性。

[0193] 本发明实施例还提供了一种数据处理系统,包括如图5至图7所阐述的控制器和如图8至图10所阐述的交换机,其具体实施方式请参见上述实施例,在此不再赘述。

[0194] 本发明实施例还提供了一种数据处理装置7,如图10所示,该装置7包括:总线74,以及连接到总线74的处理器71、存储器72和接口73,其中该接口73用于接收和发送消息;该存储器72用于存储指令,该处理器71执行该指令用于控制该接口73接收同步请求消息,所述同步请求消息为交换机请求将所述控制器上的第一流表项同步到所述交换机上的消息,所述第一流表项是所述控制器在接收到所述同步请求消息前向所述交换机发送的流表项;该处理器71执行该指令还用于根据所述同步请求消息获取所述第一流表项;该处理器71执行该指令还用于控制该接口73将所述第一流表项发送给所述交换机。

[0195] 在本发明实施例中,可选地,所述同步请求消息中携带第一时间点;该处理器71执行该指令用于根据所述同步请求消息获取所述第一流表项,包括用于:

[0196] 解析所述同步请求消息,获得所述第一时间点;

[0197] 将所述第一时间点前处理器71向所述交换机发送的流表项确定为所述第一流表

项；

[0198] 获取所述第一流表项。

[0199] 在本发明实施例中,可选地,在处理器71执行该指令用于控制该接口73将所述第一流表项发送给所述交换机之前,该处理器71执行该指令还用于:

[0200] 向所述交换机发送同步开始消息,所述同步开始消息用于通知所述交换机所述控制器开始发送所述第一流表项;

[0201] 在处理器71执行该指令用于控制该接口73将所述第一流表项发送给所述交换机之后,该处理器71执行该指令还用于:

[0202] 向所述交换机发送同步结束消息,所述同步结束消息用于通知所述交换机所述控制器结束发送所述第一流表项。

[0203] 在本发明实施例中,可选地,在处理器71执行该指令用于控制该接口73将所述第一流表项发送给所述交换机之前,该处理器71执行该指令还用于:

[0204] 对所述第一流表项设置标记,所述标记用于指示所述第一流表项为所述交换机所请求同步的流表项。

[0205] 因此,本发明实施例的数据处理装置,通过接收交换机请求同步流表项的同步请求消息,将其请求的流表项重新发送给交换机,从而实现了数据处理装置和交换机之间的流表项同步,避免因交换机丢失流表项而造成数据处理装置和交换机之间的流表项不同步的现象,增加了数据传输过程中的自动纠错功能,提高了网络设备信息交互的安全性。

[0206] 本发明实施例还提供了另一种数据处理装置8,如图11所示,该装置8包括:总线84,以及连接到总线84的处理器81、存储器82和接口83,其中该接口83用于接收和发送消息;该存储器82用于存储指令,该处理器81执行该指令用于控制该接口83向控制器发送同步请求消息,所述同步请求消息为所述交换机请求将所述控制器上的第一流表项同步到所述交换机上的消息,所述第一流表项是所述控制器在接受到所述同步请求消息前向所述交换机发送的流表项;该处理器81执行该指令还用于接收所述控制器发送的所述第一流表项。

[0207] 在本发明实施例中,可选地,所述同步请求消息中携带第一时间点,所述第一时间点用于指示所述控制器将所述第一时间点前所述控制器向所述交换机发送的流表项确定为所述第一流表项。

[0208] 在本发明实施例中,可选地,该处理器81执行该指令用于控制该接口83向控制器发送同步请求消息,包括用于:

[0209] 对已存储的流表项进行检验;

[0210] 根据检验结果控制该接口83向所述控制器发送所述同步请求消息。

[0211] 在本发明实施例中,可选地,该处理器81执行该指令用于根据检验结果控制该接口83向所述控制器发送所述同步请求消息,包括:

[0212] 当检验到所述已存储的流表项的序列号不连续或者所述已存储的流表项丢失时,控制该接口83向所述控制器发送所述同步请求消息。

[0213] 在本发明实施例中,可选地,在该处理器81执行该指令用于控制该接口83向控制器发送同步请求消息之后,以及该处理器81执行该指令用于控制该接口83接收所述控制器发送的所述第一流表项之前,还用于:

[0214] 控制该接口83接收所述控制器发送的同步开始消息,所述同步开始消息用于通知所述交换机所述控制器开始发送所述第一流表项;

[0215] 在该处理器81执行该指令用于控制该接口83接收所述控制器发送的所述第一流表项之后,还用于:

[0216] 控制该接口83接收所述控制器发送的同步结束消息,所述同步结束消息用于通知所述交换机所述控制器结束发送所述第一流表项;

[0217] 根据所述同步开始消息和所述同步结束消息对所述第一流表项设置标记,所述标记用于指示所述第一流表项为所述交换机所请求同步的流表项。

[0218] 在本发明实施例中,可选地,所述第一流表项携带标记,所述标记用于指示所述第一流表项为所述交换机所请求同步的流表项。

[0219] 因此,本发明实施例的数据处理装置,通过向控制器发送请求同步流表项的同步请求消息,并获取所请求的流表项,从而实现了控制器和本数据处理装置之间的流表项同步,避免因数据处理装置丢失流表项而造成控制器和本数据处理装置的流表项不同步的现象,增加了数据传输过程中的自动纠错功能,提高了网络设备信息交互的安全性。

[0220] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程,是可以通过计算机程序来指令相关的硬件来完成,所述的程序可存储于一计算机可读取存储介质中,该程序在执行时,可包括如上述各方法的实施例的流程。其中,所述的存储介质可为磁碟、光盘、只读存储记忆体(Read-Only Memory,ROM)或随机存储记忆体(Random Access Memory,RAM)等。

[0221] 以上所揭露的仅为本发明较佳实施例而已,当然不能以此来限定本发明之权利范围,因此依本发明权利要求所作的等同变化,仍属本发明所涵盖的范围。

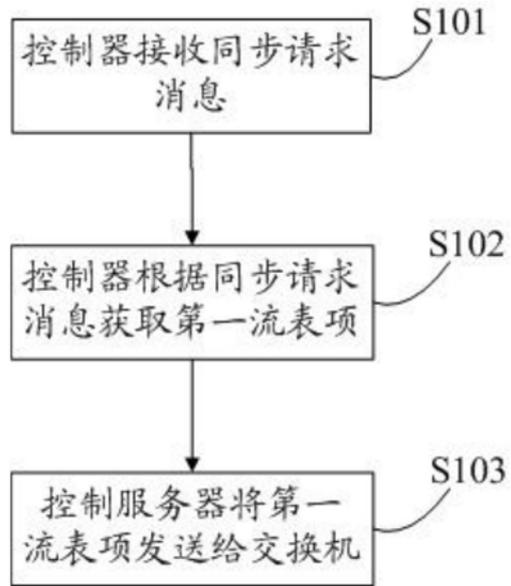


图1

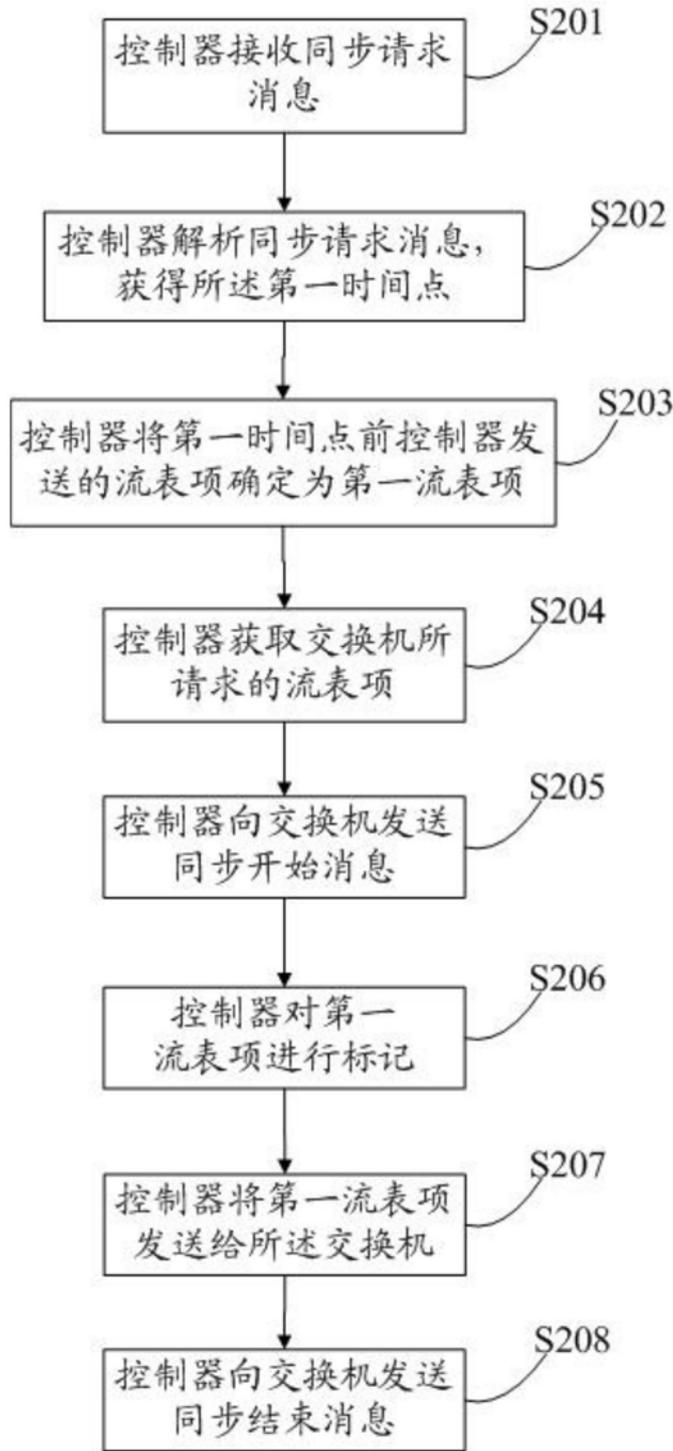


图2

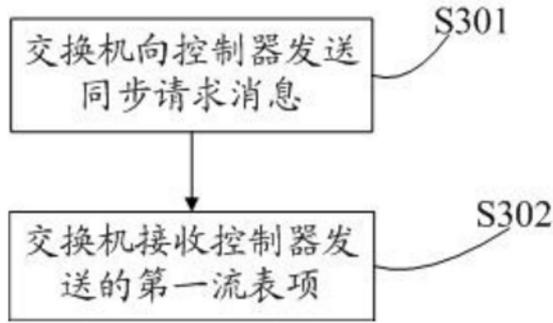


图3

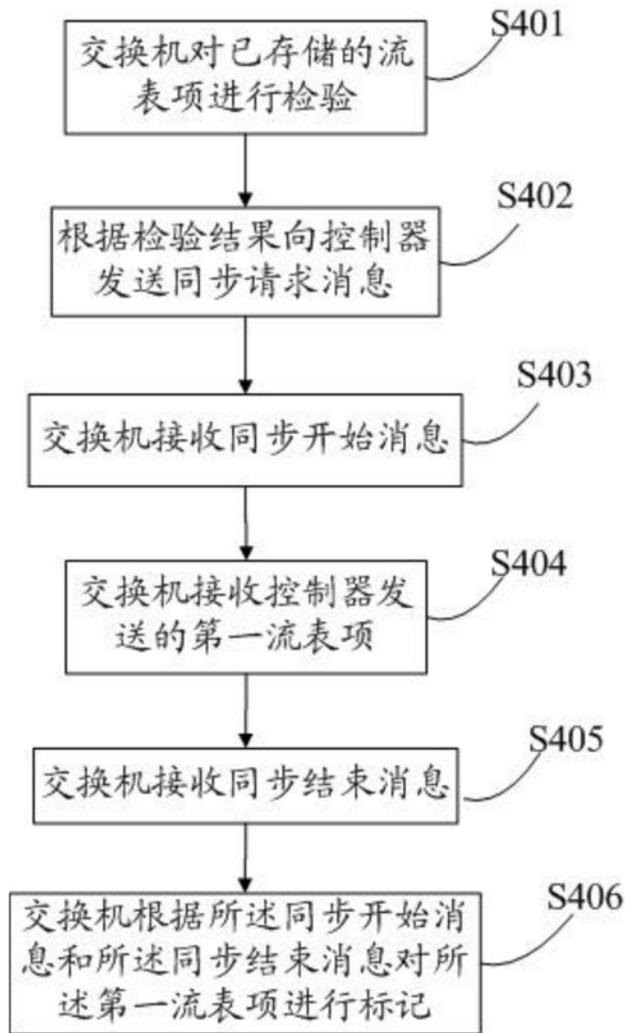


图4

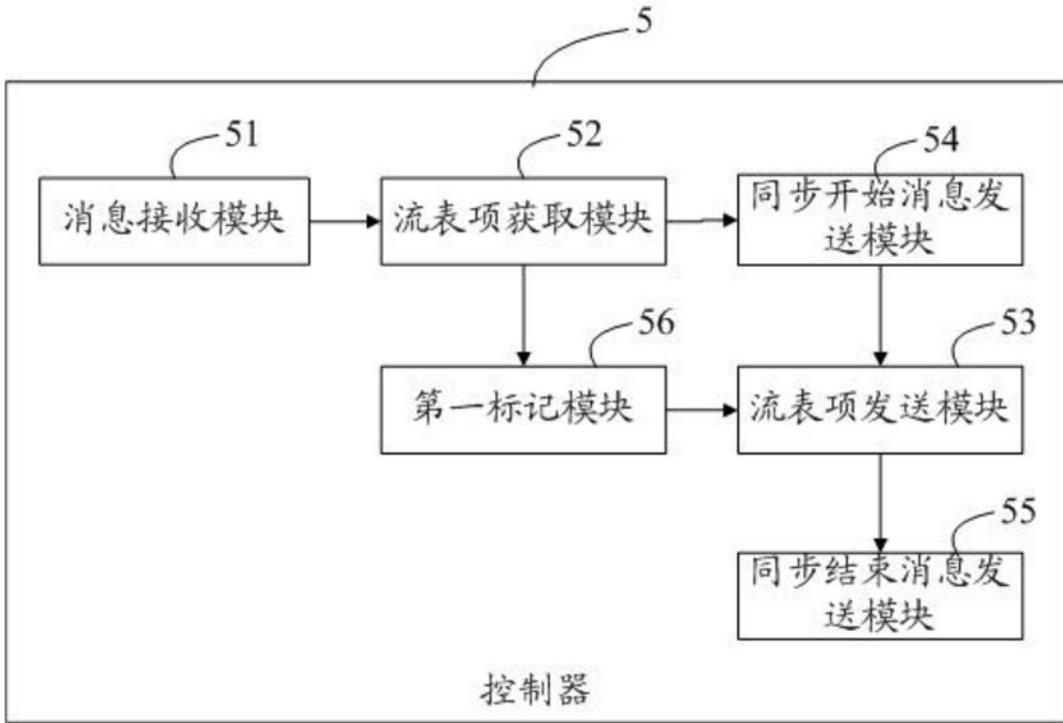


图5

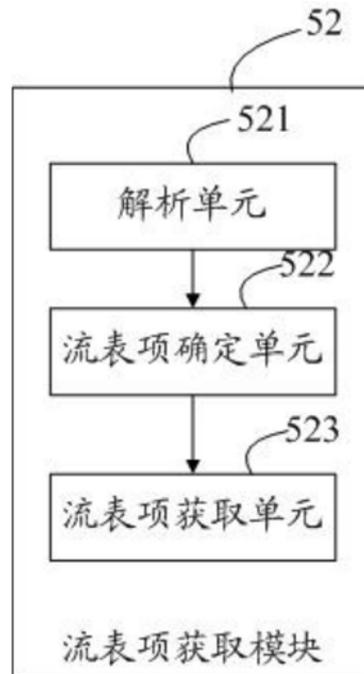


图6

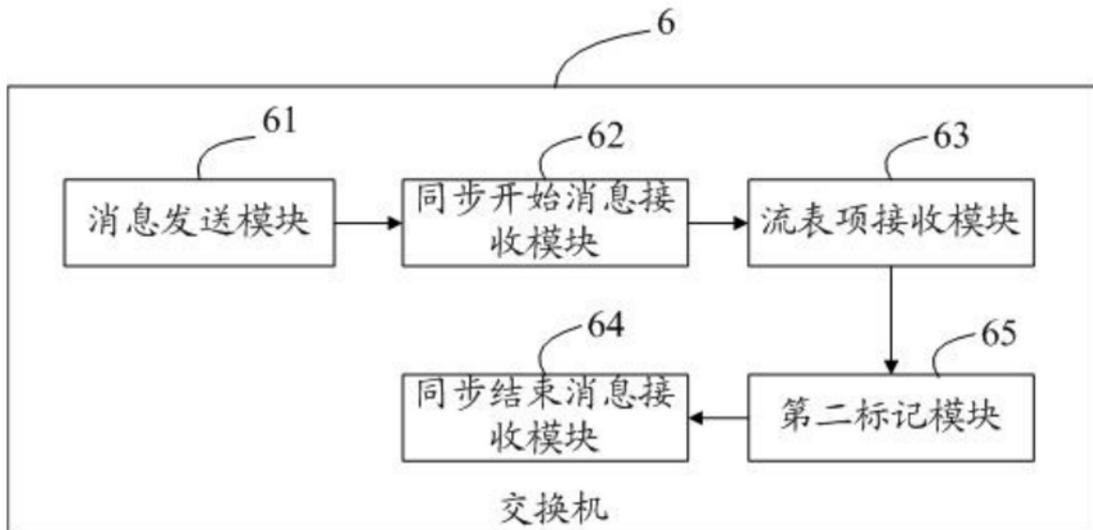


图7

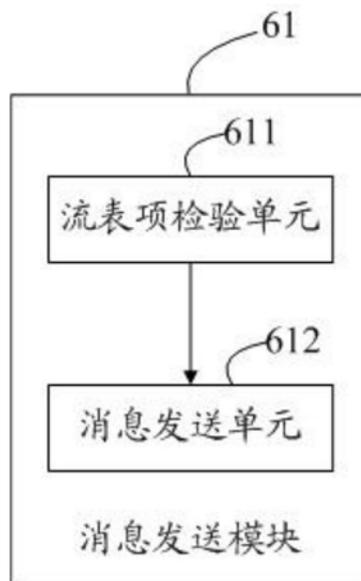


图8

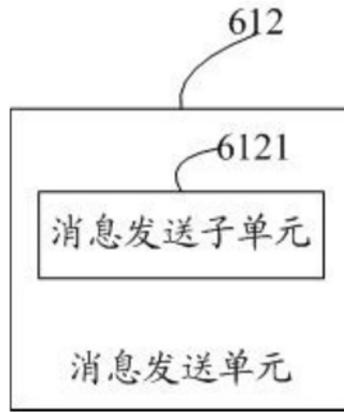


图9

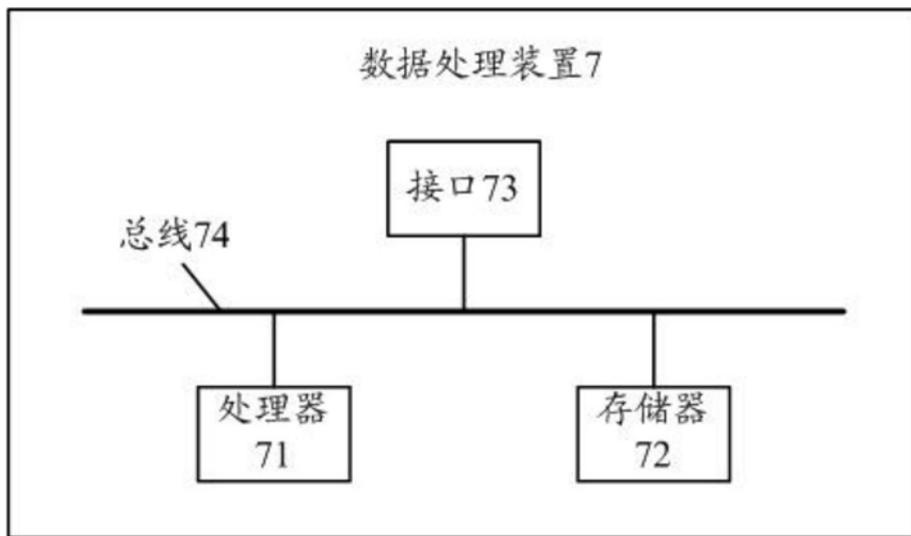


图10

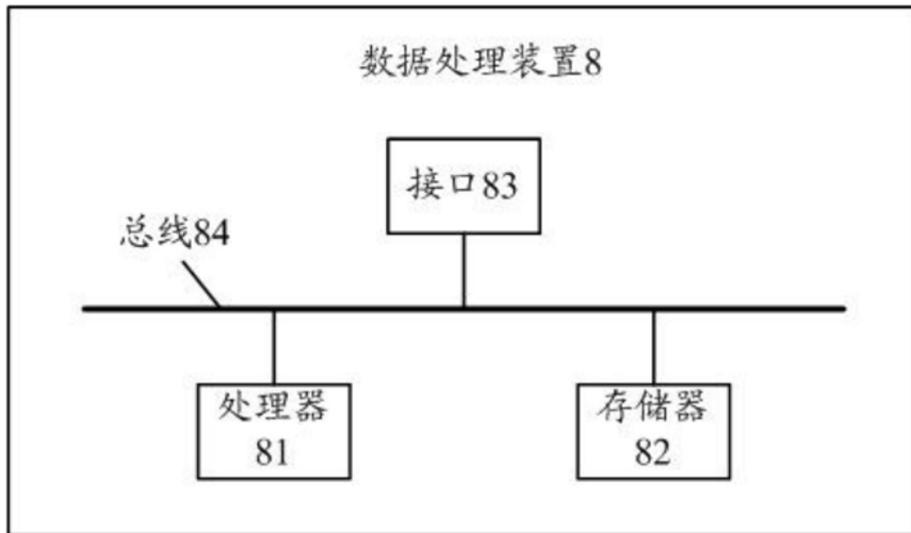


图11