



# (12) 发明专利申请

(10) 申请公布号 CN 114915595 A

(43) 申请公布日 2022. 08. 16

(21) 申请号 202210265963.9

H04L 43/0882 (2022.01)

(22) 申请日 2022.03.11

H04L 41/147 (2022.01)

(71) 申请人 北京邮电大学

地址 100876 北京市海淀区西土城路10号

申请人 国网上海市电力公司

国网智能电网研究院有限公司

(72) 发明人 张会彬 陶静 陈晓露 刘世栋

肖云杰 卜宪德 吴佳伟 翟欢

孙小惠 张雪影

(74) 专利代理机构 北京风雅颂专利代理有限公司

11403

专利代理师 安凯

(51) Int. Cl.

H04L 47/127 (2022.01)

H04L 47/125 (2022.01)

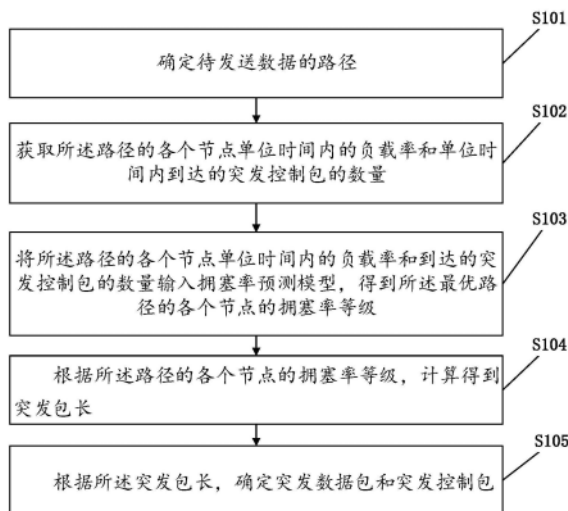
权利要求书2页 说明书10页 附图3页

(54) 发明名称

突发包装配方法和电子设备

(57) 摘要

本申请提供一种突发包装配方法和电子设备。所述方法包括：确定待发送数据的路径；获取所述路径的各个节点单位时间内的负载率和单位时间内到达的突发控制包的数量；将所述路径的各个节点单位时间内的负载率和到达的突发控制包的数量输入拥塞率预测模型，得到所述最优路径的各个节点的拥塞率等级；根据所述路径的各个节点的拥塞率等级，计算得到突发包长；根据所述突发包长，确定突发数据包和突发控制包。达到针对不同拥塞率的路径生成不同长度的突发数据包和对应的突发控制包的效果。



1. 一种突发包装配方法,其特征在于,包括:

确定待发送数据的路径;

获取所述路径的各个节点单位时间内的负载率和单位时间内到达的突发控制包的数量;

将所述路径的各个节点单位时间内的负载率和到达的突发控制包的数量输入拥塞率预测模型,得到所述路径的各个节点的拥塞率等级;

根据所述路径的各个节点的拥塞率等级,计算得到突发包长;

根据所述突发包长,确定突发数据包和突发控制包。

2. 根据权利要求1所述的方法,其特征在于,所述确定待发送数据的路径为最优路径;

所述确定待发送数据的最优路径,包括:

获取网络拓扑信息;

根据所述网络拓扑信息,得到通信代价矩阵;

根据所述通信代价矩阵,利用K最短路径算法,得到K条最短路径;

分别计算每一所述K条最短路径的平均拥塞率,得到K个平均拥塞率;

将所述K个平均拥塞率中的最小值对应的最短路径作为最优路径。

3. 根据权利要求1所述的方法,其特征在于,所述获取所述路径的各个节点单位时间内的负载率和单位时间内到达的突发控制包的数量,包括:

所述路径的每个节点定时统计自己的所述单位时间内的负载率和单位时间内到达的突发控制包的数量;

所述路径的每个节点定时向第一相邻节点发送自身与除所述第一相邻节点的所述单位时间内的负载率和所述单位时间内到达的突发控制包的数量;

在所述路径的任意节点获取所述路径的各个节点单位时间内的负载率和单位时间内到达的突发控制包的数量。

4. 根据权利要求1所述的方法,其特征在于,所述拥塞率预测模型的训练过程,包括:

建立反向传播神经网络;

获取训练用各个节点单位时间内的负载率、训练用单位时间内到达的突发控制包的数量和各个节点的真拥塞率等级;

将所述训练用各个节点单位时间内的负载率和所述训练用单位时间内到达的突发控制包的数量输入所述反向传播神经网络,得到训练用各个节点的拥塞率等级;

根据上述训练用各个节点的拥塞率等级和所述各个节点的真拥塞率等级计算误差函数;

利用所述误差函数训练所述反向传播神经网络,得到拥塞率预测模型。

5. 根据权利要求4所述的方法,其特征在于,通过以下公式得到所述误差函数:

$$E = \frac{1}{2} \sum_{i=1}^e (Y_i - O_i)^2$$

其中,E表示所述误差函数,e表示所述路径上所有节点的个数, $Y_i$ 表示所述各个节点的真拥塞率等级, $O_i$ 表示所述训练用各个节点的拥塞率等级。

6. 根据权利要求1所述的方法,其特征在于,所述根据所述路径的各个节点的拥塞率等

级,计算得到突发包长,包括:

根据所述路径的各个节点的拥塞率等级,得到路径的拥塞率;

根据所述路径的拥塞率,得到突发包长。

7. 根据权利要求6所述的方法,其特征在于,通过以下公式计算所述路径的拥塞率:

$$M = \frac{\sum_{i=1}^n (G_i)}{3n}$$

其中,M表示所述路径的拥塞率,G<sub>i</sub>表示所述路径的某个节点的拥塞率等级,n表示所述路径的节点个数。

8. 根据权利要求1所述的方法,其特征在于,所述根据所述突发包长,得到突发数据包,包括:

设定时间门限;

缓存所述待发送数据,得到缓存待发送数据;

根据所述突发包长和所述时间门限,读取对应长度的所述缓存待发送数据,得到所述突发数据包。

9. 根据权利要求1所述的方法,其特征在于,所述突发控制包包括偏置时间;

通过以下公式计算所述偏置时间:

$$T_o = k \times T_{bcp} + T_{sw}$$

其中,T<sub>o</sub>表示所述偏置时间,k表示中间节点的个数,T<sub>bcp</sub>表示所述中间节点处突发控制包的处理时间,T<sub>sw</sub>表示所述中间节点处交换建立的时间;

所述偏置时间为所述突发控制包的发送完成时间和所述突发数据包的发送开始时间的时间间隔。

10. 一种电子设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,其特征在于,所述处理器执行所述程序时实现如权利要求1至9任意一项所述的方法。

## 突发包装配方法和电子设备

### 技术领域

[0001] 本申请涉及光突发交换技术领域,尤其涉及一种突发包装配方法和电子设备。

### 背景技术

[0002] 目前,光网络的数据交换方式主要有OCS (Optical Circuit Switching,光线路交换)、OPS (Optical Packet Switching,光分组交换) 和OBS (Optical Burst Switching,光突发包交换) 三种方式。由于光缓存器件还不成熟,因此在全光网络中最可行的全光交换技术只有OBS。在OBS网络中,实现了控制信号和数据信号的分离传输,边缘节点首先发送BCP (Burst Control Packet,突发控制包) 到核心节点进行资源的预留,经过一个偏置时间后再发送BDP (Burst Data Packet,突发数据包) 到核心节点,直接在光域实现路由交换。

[0003] 在OBS网络中,从源节点到目的节点的路径可能只有一条,也可能有多条。因此对于不同时刻,或者相同时刻的不同路径上从源节点到目的节点的路径拥塞率是不同的。而在当前的OBS资源预留协议中,无论是JIT还是JET,对BDP的汇聚都采用了固定包长或固定时间的汇聚,并没有根据网络的拥塞率进行汇聚,无法适应不同网络拥塞率的路径。若在低拥塞率和高拥塞率的路径中汇聚的BDP长度相同,对于低拥塞率的路由则不能充分利用路由的带宽,而对于高拥塞率的路由则会增加数据包冲突的概率,造成数据包丢失。

### 发明内容

[0004] 有鉴于此,本申请的目的在于提出一种突发包装配方法和电子设备。基于上述目的,本申请提供了一种突发包装配方法,其特征在于,包括:

[0005] 确定待发送数据的路径;

[0006] 获取所述路径的各个节点单位时间内的负载率和单位时间内到达的突发控制包的数量;

[0007] 将所述路径的各个节点单位时间内的负载率和到达的突发控制包的数量输入拥塞率预测模型,得到所述路径的各个节点的拥塞率等级;

[0008] 根据所述路径的各个节点的拥塞率等级,计算得到突发包长;

[0009] 根据所述突发包长,确定突发数据包和突发控制包。

[0010] 可选的,所述确定待发送数据的路径为最优路径;

[0011] 所述确定待发送数据的最优路径,包括:

[0012] 获取网络拓扑信息;

[0013] 根据所述网络拓扑信息,得到通信代价矩阵;

[0014] 根据所述通信代价矩阵,利用K最短路径算法,得到K条最短路径;

[0015] 分别计算每一所述K条最短路径的平均拥塞率,得到K个平均拥塞率;

[0016] 将所述K个平均拥塞率中的最小值对应的最短路径作为最优路径。

[0017] 可选的,所述获取所述路径的各个节点单位时间内的负载率和单位时间内到达的突发控制包的数量,包括:

[0018] 所述路径的每个节点定时统计自己的所述单位时间内的负载率和单位时间内到达的突发控制包的数量；

[0019] 所述路径的每个节点定时向第一相邻节点发送自身与除所述第一相邻节点的所述单位时间内的负载率和所述单位时间内到达的突发控制包的数量；

[0020] 在所述路径的任意节点获取所述路径的各个节点单位时间内的负载率和单位时间内到达的突发控制包的数量。

[0021] 可选的,所述拥塞率预测模型的训练过程,包括:

[0022] 建立反向传播神经网络;

[0023] 获取训练用各个节点单位时间内的负载率、训练用单位时间内到达的突发控制包的数量和各个节点的真拥塞率等级;

[0024] 将所述训练用各个节点单位时间内的负载率和所述训练用单位时间内到达的突发控制包的数量输入所述反向传播神经网络,得到训练用各个节点的拥塞率等级;

[0025] 根据上述训练用各个节点的拥塞率等级和所述各个节点的真拥塞率等级计算误差函数;

[0026] 利用所述误差函数训练所述反向传播神经网络,得到拥塞率预测模型。

[0027] 可选的,通过以下公式得到所述误差函数:

$$[0028] \quad E = \frac{1}{2} \sum_{i=1}^e (Y_i - O_i)^2$$

[0029] 其中,E表示所述误差函数,e表示所述路径上所有节点的个数, $Y_i$ 表示所述各个节点的真拥塞率等级, $O_i$ 表示所述训练用各个节点的拥塞率等级。

[0030] 可选的,所述根据所述路径的各个节点的拥塞率等级,计算得到突发包长,包括:

[0031] 根据所述路径的各个节点的拥塞率等级,得到路径的拥塞率;

[0032] 根据所述路径的拥塞率,得到突发包长。

[0033] 可选的,通过以下公式计算所述路径的拥塞率:

$$[0034] \quad M = \frac{\sum_{i=1}^n (G_i)}{3n}$$

[0035] 其中,M表示所述路径的拥塞率, $G_i$ 表示所述路径的某个节点的拥塞率等级,n表示所述路径的节点个数。

[0036] 可选的,所述根据所述突发包长,得到突发数据包,包括:

[0037] 设定时间门限;

[0038] 缓存所述待发送数据,得到缓存待发送数据;

[0039] 根据所述突发包长和所述时间门限,读取对应长度的所述缓存待发送数据,得到所述突发数据包。

[0040] 可选的,所述突发控制包包括偏置时间;

[0041] 通过以下公式计算所述偏置时间:

$$[0042] \quad T_o = k \times T_{bcp} + T_{sw}$$

[0043] 其中, $T_o$ 表示所述偏置时间,k表示中间节点的个数, $T_{bcp}$ 表示所述中间节点处突发控制包的处理时间, $T_{sw}$ 表示所述中间节点处交换建立的时间;

[0044] 所述偏置时间为所述突发控制包的发送完成时间和所述突发数据包的发送开始时间的的时间间隔。

[0045] 基于同一发明构思,本申请还提供了一种电子设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,其特征在于,所述处理器执行所述程序时实现如上述任意一项所述的突发包装配方法。

[0046] 从上面所述可以看出,本申请提供的突发包装配方法和电子设备,通过确定待发送数据的路径,获取该路径的各个节点单位时间内的负载率和单位时间内到达的突发控制包的数量,并将其输入拥塞率预测模型,得到路径的各个节点的拥塞率等级,根据拥塞率等级计算得到突发包长,根据突发包长来确定突发数据包和突发控制包。对于光突发交换网络来说,对于不同时刻,或者相同时刻的不同路径上从源节点到目的节点的路径拥塞率是不同的,通过拥塞率预测模型实时预测网络的拥塞率,并根据拥塞率来相应的调整突发数据包的长度,对于低拥塞率的路由而言,能够充分利用路由的带宽,而对于高拥塞率的路由而言,有效降低了数据包冲突的概率,避免了数据包的丢失。

## 附图说明

[0047] 为了更清楚地说明本申请或相关技术中的技术方案,下面将对实施例或相关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申请的实施例,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。

[0048] 图1为本申请实施例的突发包装配方法流程图;

[0049] 图2为本申请实施例的网络拓扑结构示意图;

[0050] 图3为本申请实施例的通信代价矩阵示意图;

[0051] 图4为本申请实施例的神经网络模型示意图;

[0052] 图5为本申请实施例的神经网络训练过程示意图;

[0053] 图6为本申请实施例的电子设备示意图。

## 具体实施方式

[0054] 为使本申请的目的、技术方案和优点更加清楚明白,以下结合具体实施例,并参照附图,对本申请进一步详细说明。

[0055] 需要说明的是,除非另外定义,本申请实施例使用的技术术语或者科学术语应当为本申请所属领域内具有一般技能的人士所理解的通常意义。本申请实施例中使用的“第一”、“第二”以及类似的词语并不表示任何顺序、数量或者重要性,而只是用来区分不同的组成部分。“包括”或者“包含”等类似的词语意指出现该词前面的元件或者物件涵盖出现在该词后面列举的元件或者物件及其等同,而不排除其他元件或者物件。“连接”或者“相连”等类似的词语并非限定于物理的或者机械的连接,而是可以包括电性的连接,不管是直接的还是间接的。“上”、“下”、“左”、“右”等仅用于表示相对位置关系,当被描述对象的绝对位置改变后,则该相对位置关系也可能相应地改变。

[0056] 如背景技术部分所述,相关技术中在低拥塞率和高拥塞率的路径中汇聚出的突发数据包的长度相同,然而,这种不根据网络拥塞率对突发数据包进行汇聚的技术方案,使得

汇聚出的突发数据包无法适应不同网络拥塞率的路径,使得对于低拥塞率的路由不能充分利用该路由的带宽,而对于高拥塞率的路由则会增加数据包冲突的概率,造成数据包的丢失。

[0057] 综合上述考虑,本申请提出一种突发包装配方法,基于预先训练的拥塞率预测模型,利用路径的各个节点单位时间内的负载率和到达的突发控制包的数量来预测路径的各个节点的拥塞率等级,根据拥塞率等级来决定汇聚的突发数据包的长度,从而能够使得汇聚出的突发数据包有效适应不同网络拥塞率的路径,对于低拥塞率的路由能够充分利用该路由的带宽,而对于高拥塞率的路由则会降低数据包冲突的概率,避免数据包的丢失。

[0058] 以下,通过具体的实施例来详细说明本申请的技术方案。

[0059] 参考图1,本申请实施例的突发包装配方法,包括以下步骤:

[0060] 步骤S101,确定待发送数据的路径。

[0061] 可选的,所述确定待发送数据的路径为最优路径;

[0062] 在本实施例中,确定待发送数据的路径选取为最优路径,在本申请中,因为确定的待发送数据的路径为预测路径,若随意选取路径,则可能选定的路径中存在拥塞率极高的节点,导致数据丢包率上升,而最优路径上的拥塞率最小,丢包的可能性也最小,因此本实施例选取最优路径传输,但需注意的是,本申请可应用于所有路径。

[0063] 所述确定待发送数据的最优路径,包括:

[0064] 获取网络拓扑信息;

[0065] 在本实施例中,为了模拟真实的网络拓扑结构,选择NSFNET的14点结构作为通信网络模型。参考图2,为本申请实施例的网络拓扑结构,其中包括14个网络节点,图中方形的为边缘节点,圆形的为核心节点,直线上的数字为直线连接的两个节点间的通信代价,节点附近的数字为该节点的拥塞率等级,此处节点的拥塞率等级是经拥塞率预测模型预测后得到的。在本申请中,拥塞率模型实时获取所有路径上节点的信息并对其拥塞率等级进行预测。

[0066] 根据所述网络拓扑信息,得到通信代价矩阵;

[0067] 在本步骤中,两个节点之间的通信代价可由网络带宽、服务优先级等信息确定。

[0068] 在本实施例中,参考图3,为本申请实施例的通信代价矩阵。用 $14 \times 14$ 通信代价矩阵P表示网络拓扑中相邻节点的连接关系:若两个节点 $i, j$ 之间存在链路,则将 $p_{ij}$ 设置为两个节点之间的通信代价;若两个节点之间不存在链路,则将 $p_{ij}$ 设置为无穷大,用inf表示;同一个节点之间的通信代价 $p_{ii}$ 设置为0。

[0069] 根据所述通信代价矩阵,利用K最短路径算法,得到K条最短路径;

[0070] 在本步骤中,K最短路径(k-shortest paths)算法基于Dijkstra最短路径算法,通过不断进行Dijkstra算法计算出K条最短路径。此算法可以分为两部分,通过Dijkstra算法计算出第一条最短路径P[1],然后在此基础上依次计算出其他K-1条最短路径。在求P[i+1]时,将P[i]上除了终止节点外的所有节点都视为偏离节点,并计算每个偏离节点到终止节点的最短路径,再与之前的P[i]上起始节点到偏离节点的路径拼接,构成候选路径,进而求得最短偏离路径。

[0071] 在本实施例中,根据K最短路径算法计算出从节点1到节点14的5条最短路径及其通信代价,如表1所示。

[0072] 表1五条最短路径及通信代价

序号	最短路径	通信代价
1	1-8-9-14	26
2	1-2-4-13-14	29
3	1-8-9-12-13-14	34
4	1-2-4-5-7-8-9-14	35
5	1-8-9-12-11-14	35

[0074] 分别计算每一所述K条最短路径的平均拥塞率,得到K个平均拥塞率;

[0075] 在本步骤中,每条路径的拥塞率为路径中所有节点的拥塞率等级之和的归一化值。

[0076] 可选的,通过以下公式计算所述路径的拥塞率:

$$[0077] \quad M = \frac{\sum_{i=1}^n (G_i)}{3n}$$

[0078] 其中,M表示所述路径的拥塞率,G<sub>i</sub>表示所述路径的某个节点的拥塞率等级,n表示所述路径的节点个数。

[0079] 五条最短路径的拥塞率如表2所示:

[0080] 表2五条最短路径的拥塞率

序号	最短路径	通信代价	路径拥塞率M
1	1-2-4-13-14	29	0.67
2	1-2-4-5-7-8-9-14	35	0.71
3	1-8-9-12-13-14	34	0.89
4	1-8-9-12-11-14	35	0.89
5	1-8-9-14	26	0.92

[0082] 将所述K个平均拥塞率中的最小值对应的最短路径作为最优路径。

[0083] 在本实施例中,对表2中得到的五条最短路径的拥塞率进行排序,可见1号路径的路径拥塞率最小,因此可以将1-2-4-13-14作为本次通信的最优路径。

[0084] 步骤S102,获取所述路径的各个节点单位时间内的负载率和单位时间内到达的突发控制包的数量。

[0085] 可选的,所述获取所述路径的各个节点单位时间内的负载率和单位时间内到达的突发控制包的数量,包括:

[0086] 所述路径的每个节点定时统计自己的所述单位时间内的负载率和单位时间内到达的突发控制包的数量;

[0087] 所述路径的每个节点定时向第一相邻节点发送自身与除所述第一相邻节点的所述单位时间内的负载率和所述单位时间内到达的突发控制包的数量;

[0088] 在所述路径的任意节点获取所述路径的各个节点单位时间内的负载率和单位时间内到达的突发控制包的数量。

[0089] 在本实施例中,光网络中的各个节点会定时统计单位时间内的负载率、到达的突发控制包数量、端口利用率等信息,并通过控制信道 (BCP传输的信道) 定时向相邻节点发送



自己以及其他相邻节点的数据,类似于动态路由维护各个节点路由表的过程。以图2中的2号节点为例,1、3、4号节点每隔2S向2号节点发送包含统计数据的数据包,2号节点根据数据包的数据更新自己的数据库。同时,2号节点会每隔2S向1号节点发送自己的统计数据以及3、4号节点的数据,也会向3、4号节点发送自己的统计数据以及其他节点的数据。这样,4号节点就会拥有1、2、3号节点的数据,又会向5、13号节点发送。一定时间之后,每个节点都能够拥有其他节点的单位时间内的负载率、到达的BCP数量、端口利用率等信息,因此,此时从任意节点均可获取所有节点的信息。

[0090] 步骤S103,将所述路径的各个节点单位时间内的负载率和到达的突发控制包的数量输入拥塞率预测模型,得到所述路径的各个节点的拥塞率等级。

[0091] 在本步骤中,将前述步骤中获取的所述路径的各个节点单位时间内的负载率和到达的突发控制包的数量输入拥塞率预测模型,得到路径的各个节点的拥塞率等级,上述步骤是实时进行的。

[0092] 可选的,所述拥塞率预测模型的训练过程,包括:

[0093] 建立反向传播神经网络;

[0094] 在本步骤中,反向传播神经网络从结构上可以分为输入层、隐含层和输出层三部分,隐含层的层数可以根据实际应用进行选择。在光网络中,为了减少计算量,降低传输延迟,可以选用一层隐含层的反向传播神经网络,提高预测的准确度可以通过增加隐含层神经元个数来实现。输入层神经元数由输入矩阵的特征决定,输出层神经元数由预期样本类别决定,而隐含层神经元数由  $l = \sqrt{n + m} + a$  (n为输入层神经元个数,m为输出层神经元个数,a为[1,10]之间的常数)确定。

[0095] 在本实施例中,参考图4,为本申请实施例的反向传播神经网络模型。本次预测中反向传播神经网络输入层采用2个神经元,隐含层采用10个神经元,输出层采用3个神经元,X1、X2表示输入矩阵的特征,在本实施例中为路径的各个节点单位时间内的负载率和到达的突发控制包的数量,H1-H10为神经元的个数,01-03为预期样本类别,在本实施例中为3个拥塞率等级。反向传播神经网络模型中隐含层和输出层分别选择logsig(对数S型传递函数)和purelin(线性传递函数)作为传递函数,反向传播的训练函数选择traindx(学习率可变的动量BP算法)学习算法。

[0096] 其中,logsig函数的计算公式为:

$$[0097] \quad f(x) = \frac{1}{1 + e^{-x}}$$

[0098] 获取训练用各个节点单位时间内的负载率、训练用单位时间内到达的突发控制包的数量和各个节点的真拥塞率等级;

[0099] 在本步骤中,节点拥塞率等级可以自行定义,对节点拥塞率等级的定义取决于该节点的性能。

[0100] 在本实施例中,将节点拥塞率定义为三个等级,如表3所示:

[0101] 表3节点拥塞率等级定义

拥塞率等级	定义
1	单位时间内BCP到达的数量小于40并且节点负载率低于60%

2	介于1、3之间
3	单位时间内BCP到达的数量大于40并且节点负载率高于60%

[0103] 按照以上原则,将节点数据按单位时间t进行分组,计算t时间内到达该节点的突发控制包数量及负载率,获得100组数据用于神经网络训练

[0104] 将所述训练用各个节点单位时间内的负载率和所述训练用单位时间内到达的突发控制包的数量输入所述反向传播神经网络,得到训练用各个节点的拥塞率等级;

[0105] 根据上述训练用各个节点的拥塞率等级和所述各个节点的真拥塞率等级计算误差函数;

[0106] 利用所述误差函数训练所述反向传播神经网络,得到拥塞率预测模型。可选的,通过以下公式得到所述误差函数:

$$[0107] \quad E = \frac{1}{2} \sum_{i=1}^e (Y_i - O_i)^2$$

[0108] 其中,E表示所述误差函数,e表示所述路径上所有节点的个数, $Y_i$ 表示所述各个节点的真拥塞率等级, $O_i$ 表示所述训练用各个节点的拥塞率等级。

[0109] 确定各层之间的传递函数后,设定网络迭代次数epochs为500次,期望误差goal为0.01,学习速率lr为0.01,之后对节点的拥塞率数据进行学习,训练神经网络模型。其中,将60组数据作为训练数据,40组作为测试数据。

[0110] 反向传播神经网络通过误差函数不断调节各层之间的连接权值和阈值使误差函数E达到极小,从而满足期望误差的要求。

[0111] 在本实施例中,参考图5,为本申请实施例的神经网络训练过程示意图。其中,纵坐标表示均方误差(误差函数),横坐标为训练轮数,从图中可以看出,最佳训练性能是第492轮,即神经网络在进行492次重复学习后达到最佳性能,均方误差为0.029211。训练完成后对剩余40组数据进行测试,神经网络的预测正确率高达97.5%。并且随着训练数据的不断增加,神经网络的预测正确率还会进一步提高,能够正确的预测出光交换节点下一时刻的拥塞率。

[0112] 步骤S104,根据所述路径的各个节点的拥塞率等级,计算得到突发包长。

[0113] 可选的,所述根据所述路径的各个节点的拥塞率等级,计算得到突发包长,包括:

[0114] 根据所述路径的各个节点的拥塞率等级,得到路径的拥塞率;

[0115] 根据所述路径的拥塞率,得到突发包长。

[0116] 在本步骤中,光突发交换网络中会事先设置一个基本长度1,突发包长的计算原则为:所述路径的拥塞率越高,所述突发包长越短;所述路径的拥塞率越低,所述突发包长越长。如表4突发包长计算规则所示:

[0117] 表4突发包长计算规则

拥塞程度	定义	突发包长 L
严重拥塞	路径拥塞率超过 90%	1
一般拥塞	路径拥塞率超过 80%	2l
高负荷	路径拥塞率超过 70%	3l
正常负荷	路径拥塞率低于 70%	4l
l: 突发包基本长度		

[0119] 需要注意的是,上述突发包长计算规则可以根据实际情况自行调节。

[0120] 在本实施例中,路径拥塞率为67%,通信链路为1-2-4-13-14,令突发包长长度为L=1024Byte。根据表4可得,本次通信的突发数据包包长设置为4096Byte。

[0121] 步骤S105,根据所述突发包长,确定突发数据包和突发控制包。

[0122] 可选的,所述根据所述突发包长,得到突发数据包,包括:

[0123] 设定时间门限;

[0124] 缓存所述待发送数据,得到缓存待发送数据;

[0125] 根据所述突发包长和所述时间门限,读取对应长度的所述缓存待发送数据,得到所述突发数据包。

[0126] 在本步骤中,突发数据包的生成过程中需要结合MSMAP算法,通过定时器设定一个时间门限T,防止存储器中的数据量不够造成死锁。按照上述步骤中得到的突发包长读取相应长度的待发送数据,添加同步帧和包头包尾构成完整的突发数据包。

[0127] 可选的,所述突发控制包包括偏置时间;

[0128] 通过以下公式计算所述偏置时间:

$$[0129] \quad T_o = k \times T_{bcp} + T_{sw}$$

[0130] 其中, $T_o$ 表示所述偏置时间,k表示中间节点的个数, $T_{bcp}$ 表示所述中间节点处突发控制包的处理时间, $T_{sw}$ 表示所述中间节点处交换建立的时间;

[0131] 所述偏置时间为所述突发控制包的发送完成时间和所述突发数据包的发送开始时间的时间间隔。

[0132] 在本步骤中,所述突发控制包至少包括源地址、目的地址、服务优先级、传输路径、偏置时间和突发包长等基本信息。在发送时,突发控制包和突发数据包均会进入发送队列,根据突发控制包中标记的偏置时间来发送突发控制包和突发数据包到网络中。

[0133] 通过上述实施例可以看出,本申请实施例所述的突发包装配方法,通过确定待发送数据的路径,获取该路径的各个节点单位时间内的负载率和单位时间内到达的突发控制包的数量,并将其输入拥塞率预测模型,得到路径的各个节点的拥塞率等级,根据拥塞率等级计算得到突发包长,根据突发包长来确定突发数据包和突发控制包。对于低拥塞率的路由而言,能够充分利用路由的带宽,而对于高拥塞率的路由而言,有效降低了数据包冲突的概率,避免了数据包的丢失。

[0134] 此外,相比于现有技术而言,本申请首先采取确定最优路径,最优路径中的拥塞率是所有路径中最小的,在传输前就有效避免了丢包的发生,结合上述根据路径拥塞率调整突发包长进而控制突发数据包长度的方法,更加有效降低了数据包冲突的概率,避免了数据包的丢失,且结合上述方法后,即便最优路径的拥塞率也很高,后续对突发数据包长度的

调整也能使得突发数据包良好的适应该路径,避免丢包的发生。

[0135] 需要说明的是,本申请实施例的方法可以由单个设备执行,例如一台计算机或服务器等。本实施例的方法也可以应用于分布式场景下,由多台设备相互配合来完成。在这种分布式场景的情况下,这多台设备中的一台设备可以只执行本申请实施例的方法中的某一个或多个步骤,这多台设备相互之间会进行交互以完成所述的方法。

[0136] 需要说明的是,上述对本申请的一些实施例进行了描述。其它实施例在所附权利要求书的范围内。在一些情况下,在权利要求书中记载的动作或步骤可以按照不同于上述实施例中的顺序来执行并且仍然可以实现期望的结果。另外,在附图中描绘的过程不一定要求示出的特定顺序或者连续顺序才能实现期望的结果。在某些实施方式中,多任务处理和并行处理也是可以的或者可能是有利的。

[0137] 基于同一发明构思,与上述任意实施例方法相对应的,本申请还提供了一种电子设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,所述处理器执行所述程序时实现上任意一实施例所述的突发包装配方法。

[0138] 图6示出了本实施例所提供的一种更为具体的电子设备硬件结构示意图,该设备可以包括:处理器1010、存储器1020、输入/输出接口1030、通信接口1040和总线1050。其中处理器1010、存储器1020、输入/输出接口1030和通信接口1040通过总线1050实现彼此之间在设备内部的通信连接。

[0139] 处理器1010可以采用通用的CPU (Central Processing Unit,中央处理器)、微处理器、应用专用集成电路 (Application Specific Integrated Circuit,ASIC)、或者一个或多个集成电路等方式实现,用于执行相关程序,以实现本说明书实施例所提供的技术方案。

[0140] 存储器1020可以采用ROM (Read Only Memory,只读存储器)、RAM (Random Access Memory,随机存取存储器)、静态存储设备,动态存储设备等形式实现。存储器1020可以存储操作系统和其他应用程序,在通过软件或者固件来实现本说明书实施例所提供的技术方案时,相关的程序代码保存在存储器1020中,并由处理器1010来调用执行。

[0141] 输入/输出接口1030用于连接输入/输出模块,以实现信息输入及输出。输入输出/模块可以作为组件配置在设备中(图中未示出),也可以外接于设备以提供相应功能。其中输入设备可以包括键盘、鼠标、触摸屏、麦克风、各类传感器等,输出设备可以包括显示器、扬声器、振动器、指示灯等。

[0142] 通信接口1040用于连接通信模块(图中未示出),以实现本设备与其他设备的通信交互。其中通信模块可以通过有线方式(例如USB、网线等)实现通信,也可以通过无线方式(例如移动网络、WIFI、蓝牙等)实现通信。

[0143] 总线1050包括一通路,在设备的各个组件(例如处理器1010、存储器1020、输入/输出接口1030和通信接口1040)之间传输信息。

[0144] 需要说明的是,尽管上述设备仅示出了处理器1010、存储器1020、输入/输出接口1030、通信接口1040以及总线1050,但是在具体实施过程中,该设备还可以包括实现正常运行所必需的其他组件。此外,本领域的技术人员可以理解的是,上述设备中也可以仅包含实现本说明书实施例方案所必需的组件,而不必包含图中所示的全部组件。

[0145] 上述实施例的电子设备用于实现前述任一实施例中相应的突发包装配方法,并且

具有相应的方法实施例的有益效果,在此不再赘述。

[0146] 所属领域的普通技术人员应当理解:以上任何实施例的讨论仅为示例性的,并非旨在暗示本申请的范围(包括权利要求)被限于这些例子;在本申请的思路下,以上实施例或者不同实施例中的技术特征之间也可以进行组合,步骤可以以任意顺序实现,并存在如上所述的本申请实施例的不同方面的许多其它变化,为了简明它们没有在细节中提供。

[0147] 另外,为简化说明和讨论,并且为了不会使本申请实施例难以理解,在所提供的附图中可以示出或不示出与集成电路(IC)芯片和其它部件的公知的电源/接地连接。此外,可以以框图的形式示出装置,以便避免使本申请实施例难以理解,并且这也考虑了以下事实,即关于这些框图装置的实施方式的细节是高度取决于将要实施本申请实施例的平台(即,这些细节应当完全处于本领域技术人员的理解范围内)。在阐述了具体细节(例如,电路)以描述本申请的示例性实施例的情况下,对本领域技术人员来说显而易见的是,可以在没有这些具体细节的情况下或者这些具体细节有变化的情况下实施本申请实施例。因此,这些描述应被认为是说明性的而不是限制性的。

[0148] 尽管已经结合了本申请的具体实施例对本申请进行了描述,但是根据前面的描述,这些实施例的很多替换、修改和变型对本领域普通技术人员来说将是显而易见的。例如,其它存储器架构(例如,动态RAM(DRAM))可以使用所讨论的实施例。

[0149] 本申请实施例旨在涵盖落入所附权利要求的宽泛范围之内的所有这样的替换、修改和变型。因此,凡在本申请实施例的精神和原则之内,所做的任何省略、修改、等同替换、改进等,均应包含在本申请的保护范围之内。

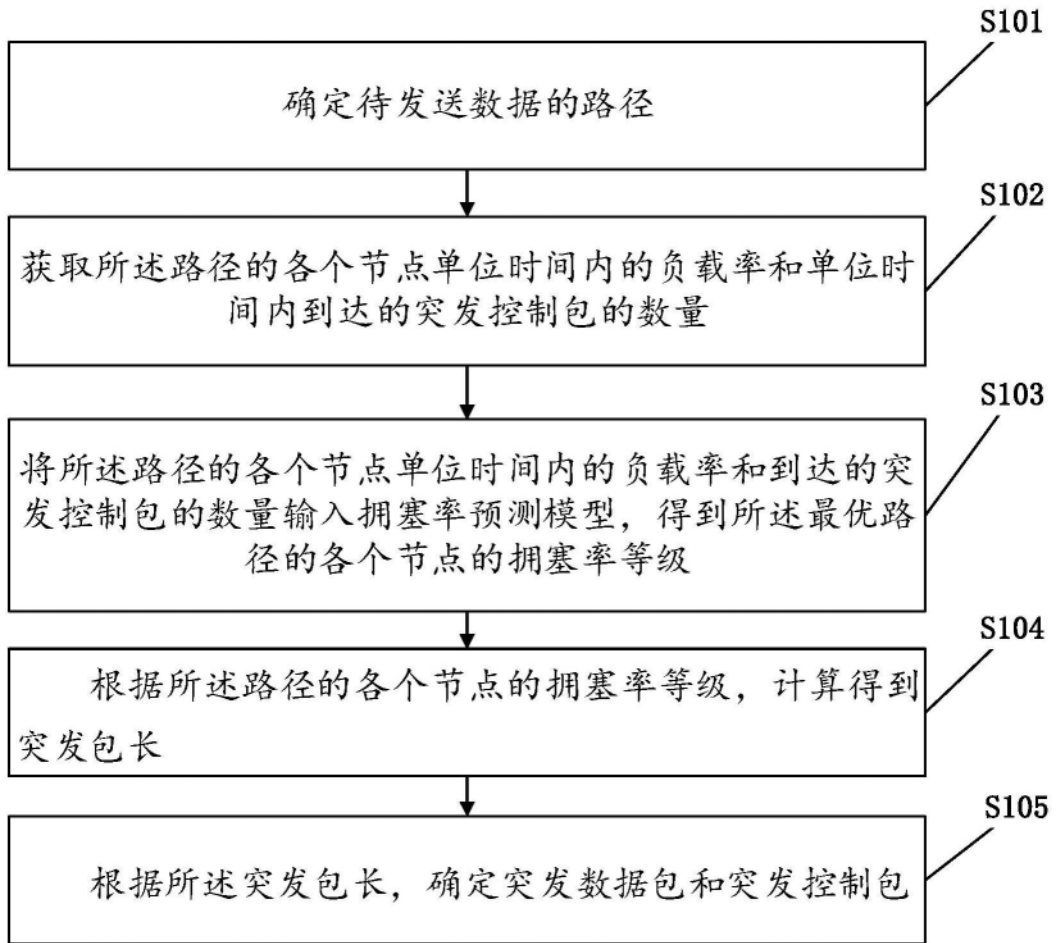


图1

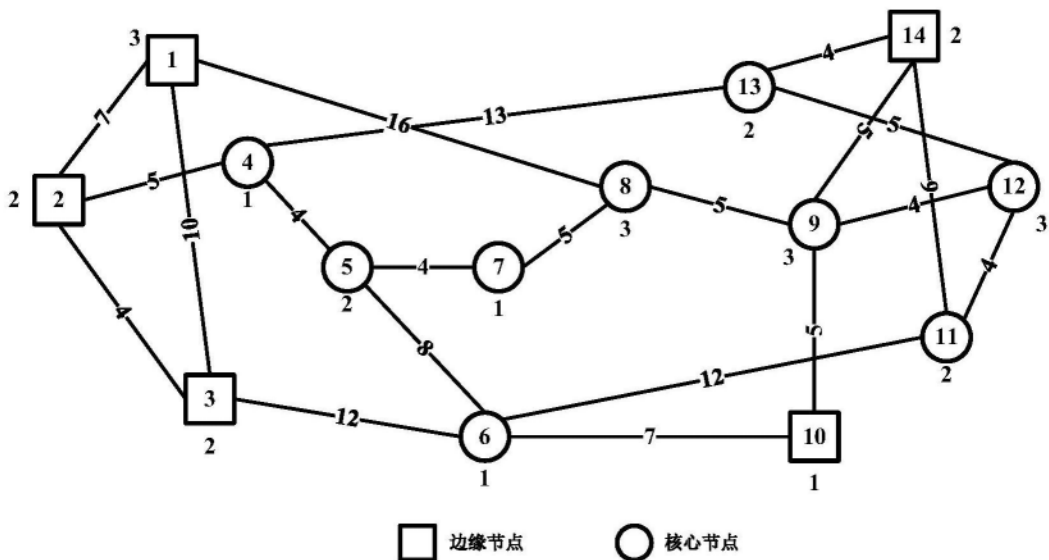


图2

$$P = \begin{bmatrix} 0 & 7 & 10 & \text{inf} & \text{inf} & \text{inf} & \text{inf} & 16 & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} \\ 7 & 0 & 4 & 5 & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} \\ 10 & 4 & 0 & \text{inf} & \text{inf} & 12 & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} \\ \text{inf} & 5 & \text{inf} & 0 & 4 & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & 13 & \text{inf} \\ \text{inf} & \text{inf} & \text{inf} & 4 & 0 & 8 & 4 & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} \\ \text{inf} & \text{inf} & 12 & \text{inf} & 8 & 0 & \text{inf} & \text{inf} & \text{inf} & 7 & 12 & \text{inf} & \text{inf} & \text{inf} \\ \text{inf} & \text{inf} & \text{inf} & \text{inf} & 4 & \text{inf} & 0 & 5 & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} \\ 16 & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & 5 & 0 & 5 & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} \\ \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & 5 & 0 & 5 & \text{inf} & 4 & \text{inf} & 5 \\ \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & 7 & \text{inf} & \text{inf} & 5 & 0 & \text{inf} & \text{inf} & \text{inf} & \text{inf} \\ \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & 12 & \text{inf} & \text{inf} & \text{inf} & \text{inf} & 0 & 4 & \text{inf} & 6 \\ \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & 4 & \text{inf} & 4 & 0 & 5 & \text{inf} \\ \text{inf} & \text{inf} & \text{inf} & 13 & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & 5 & 0 & 4 \\ \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & 5 & \text{inf} & 6 & \text{inf} & 4 & 0 \end{bmatrix}$$

图3

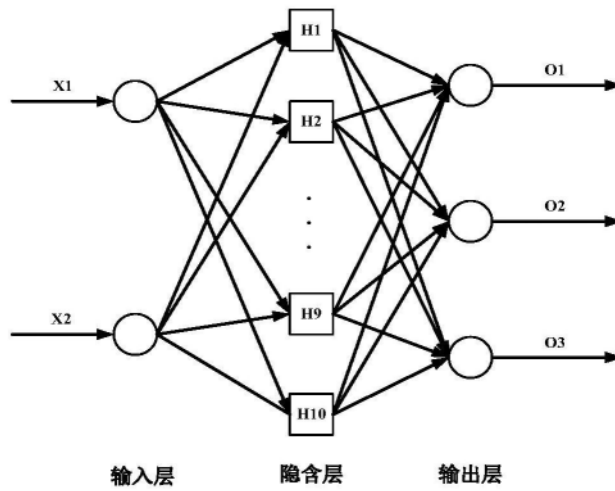


图4

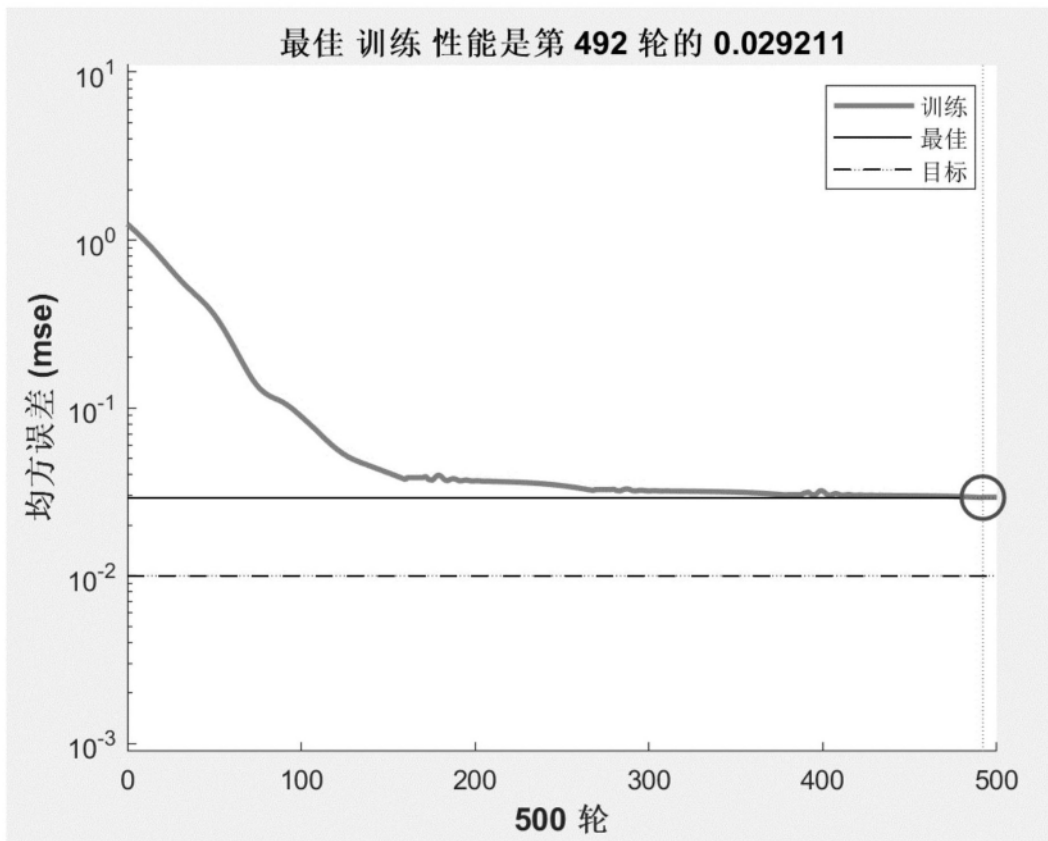


图5

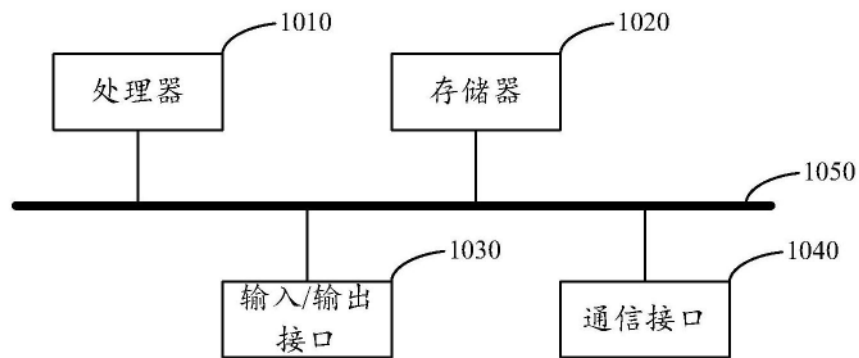


图6