



# (12) 发明专利

(10) 授权公告号 CN 113709201 B

(45) 授权公告日 2023. 05. 23

(21) 申请号 202010438782.2

H04L 41/16 (2022.01)

(22) 申请日 2020.05.22

H04W 28/08 (2023.01)

H04W 28/086 (2023.01)

(65) 同一申请的已公布的文献号

申请公布号 CN 113709201 A

(43) 申请公布日 2021.11.26

(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 刘志成 宋金铎 赵明宇 严学强

(74) 专利代理机构 北京龙双利达知识产权代理有限公司 11329

专利代理师 时林 王君

(56) 对比文件

CN 102724741 A, 2012.10.10

CN 105450684 A, 2016.03.30

CN 109240818 A, 2019.01.18

US 2017337091 A1, 2017.11.23

US 2020145337 A1, 2020.05.07

余萌迪;唐俊华;李建华.一种基于强化学习的多节点MEC计算资源分配方案.通信技术.(第12期),全文.

郭雪莹;冷冰洁;吴健;牛志升.超蜂窝网络能效延时理论关系及最优基站休眠控制.中国科学:信息科学.(第06期),全文.

审查员 倪坤晟

(51) Int. Cl.

H04L 67/10 (2022.01)

H04L 67/1008 (2022.01)

H04L 67/101 (2022.01)

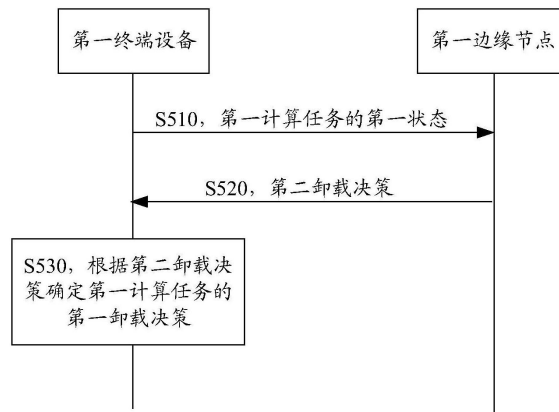
权利要求书6页 说明书26页 附图9页

(54) 发明名称

计算卸载的方法和通信装置

(57) 摘要

本申请提供了一种计算卸载的方法和装置,在网络边缘计算卸载中,边缘节点接收所服务的一个或多个终端设备发送的计算任务的状态,并根据接收到的一个或多个计算任务的状态确定计算资源的分配情况,然后边缘节点向所服务的终端设备广播计算资源的分配情况,终端设备根据该资源分配情况确定是否将计算任务卸载至边缘节点进行计算,从而使得边缘节点和终端设备能够在实际决策中对于环境有更广泛的感知能力,有效地提升两者的决策收益。



1. 一种计算卸载的方法,其特征在于,包括:

第一终端设备向第一边缘节点发送第一计算任务的第一状态,其中,所述第一边缘节点为所述第一终端设备获取计算资源的边缘节点,所述第一状态包括传输所述第一计算任务的数据流的长度、计算所述第一计算任务所需耗费的时钟周期数、所述第一计算任务的惩罚值中的至少一个;

所述第一终端设备接收所述第一边缘节点发送的第二卸载决策,所述第二卸载决策是由所述第一状态确定的,所述第二卸载决策包括一个或多个第二终端设备的计算资源分配信息,所述第二终端设备为从所述第一边缘节点获取计算资源的终端设备且所述第一终端设备为所述一个或多个第二终端设备中的一个终端设备;

所述第一终端设备根据所述第二卸载决策确定所述第一计算任务的第一卸载决策,所述第一卸载决策用于指示所述第一终端设备是否卸载所述第一计算任务至所述第一边缘节点进行计算;

所述第一终端设备根据所述第二卸载决策确定所述第一计算任务的第一卸载决策,包括:

所述第一终端设备根据所述第二卸载决策,更新所述第一计算任务的第一状态中的参数得到所述第一计算任务的第二状态;

所述第一终端设备根据所述第二状态计算所述第一计算任务的代价值,所述第一计算任务的代价值包括所述第一计算任务的本地开销和卸载开销;

所述第一终端设备根据所述第一计算任务的代价值,确定所述第一计算任务的所述第一卸载决策;

所述第一计算任务的本地开销包括第二能耗开销和第二时延开销,其中,

所述第二能耗开销包括所述第一终端设备本地计算所述第一计算任务消耗的能量和所述第一终端设备从休眠功率 $P_{\text{sleep}}$ 切换到第一工作功率 $P_1$ 消耗的能量;

所述第二时延开销包括所述第一终端设备本地计算所述第一计算任务的时延和所述第一终端设备从所述休眠功率切换到所述第一工作功率 $P_1$ 的时延;

所述第一工作功率 $P_1$ 为所述第一终端设备本地计算任务的工作功率。

2. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

当所述第一卸载决策指示所述第一终端设备卸载所述第一计算任务至所述第一边缘节点进行计算时,所述第一终端设备向所述第一边缘节点发送所述第一计算任务;

所述第一终端设备接收所述第一边缘节点发送的所述第一计算任务的计算结果;或者

当所述第一卸载决策指示所述第一终端设备不卸载所述第一计算任务时,所述第一终端设备在本地确定所述第一计算任务的计算结果。

3. 根据权利要求1所述的方法,其特征在于,所述第一终端设备根据所述第二状态计算所述第一计算任务的代价值,包括:

所述第一终端设备根据所述第二状态,使用多智能体深度强化学习MADRL算法中的第一代价值函数确定所述第一计算任务的代价值,

所述第一代价值函数包括卸载开销函数和本地计算开销函数,其中,所述卸载开销函数用于确定所述第一计算任务的卸载开销,所述本地计算开销函数用于确定所述第一计算任务的本地开销;以及

所述第一终端设备根据所述第一计算任务的代价值确定所述第一计算任务的第一卸载决策,包括:

所述第一终端设备根据所述MADRL算法迭代更新所述第一终端设备的第一计算任务的状态和所述第一计算任务的代价值;

当所述MADRL算法达到终止条件时,所述第一终端设备根据所述第一计算任务的最小代价值确定所述第一计算任务的第一卸载决策。

4. 根据权利要求3所述的方法,其特征在于,

所述第一计算任务的卸载开销包括第一能耗开销和第一时延开销,其中,

所述第一能耗开销包括所述第一终端设备将所述第一计算任务卸载至所述第一边缘节点消耗的能量;

所述第一时延开销包括所述第一终端设备将所述第一计算任务卸载至所述第一边缘节点的时延以及所述第一边缘节点计算所述第一计算任务的计算结果的时延。

5. 根据权利要求4所述的方法,其特征在于,所述第一卸载决策还包括第二工作功率,所述第二工作功率为所述MADRL算法达到终止条件时,所述第一计算任务的最小代价值对应的工作功率。

6. 根据权利要求5所述的方法,其特征在于,

当所述第一卸载决策指示所述第一终端设备卸载所述第一计算任务至所述第一边缘节点进行计算时,所述第一终端设备以所述休眠功率工作。

7. 根据权利要求4所述的方法,其特征在于,所述方法还包括:

所述第一终端设备使用第一参数对所述第一时延开销进行动态调节,所述第一参数用于表示所述第一终端设备与所述第一边缘节点处理计算任务的技术差异。

8. 根据权利要求4至7中任一项所述的方法,其特征在于,所述方法还包括:

所述第一终端设备使用第二参数对所述第一能耗开销和所述第二能耗开销进行动态调节,所述第二参数用于表示所述第一终端设备对于能耗开销的敏感程度。

9. 一种计算卸载的方法,其特征在于,包括:

第一边缘节点接收一个或多个任务的状态,所述一个或多个任务的状态包括第一终端设备发送的第一计算任务的第一状态,所述第一边缘节点是为一个或多个第二终端设备提供计算资源的边缘节点,且所述第一终端设备为所述一个或多个第二终端设备中的一个终端设备;

所述第一边缘节点根据所述一个或多个任务状态,确定第二卸载决策,所述第二卸载决策包括所述第一边缘节点对所述一个或多个第二终端设备的计算资源分配信息;

所述第一边缘节点向所述一个或多个第二终端设备广播所述第二卸载决策,所述第二卸载决策用于所述一个或多个第二终端设备确定所述第一计算任务的第一卸载决策,所述第一卸载决策用于指示所述第一终端设备是否卸载所述第一计算任务至所述第一边缘节点进行计算;

所述第一边缘节点根据所述一个或多个任务的状态确定所述第二卸载决策,包括:

所述第一边缘节点根据所述一个或多个任务的状态,更新所述第一边缘节点的第三状态得到所述第一边缘节点的第四状态,其中,

所述第三状态是所述第一边缘节点接收到所述一个或多个任务状态之前的状态;

所述第一边缘节点根据所述第四状态确定所述第一边缘节点的代价值,所述第一边缘节点的代价值为所述第一边缘节点为所述一个或多个计算任务分配计算资源的开销;

所述第一边缘节点根据所述第一边缘节点的代价值确定所述第二卸载决策。

10. 根据权利要求9所述的方法,其特征在于,所述方法还包括:

所述第一边缘节点接收所述第一终端设备发送的所述第一计算任务;

所述第一边缘节点确定所述第一计算任务的计算结果;

所述第一边缘节点向所述第一终端设备发送所述第一计算任务的计算结果。

11. 根据权利要求9所述的方法,其特征在于,所述第一边缘节点根据所述第四状态确定所述第一边缘节点的代价值,包括:

所述第一边缘节点根据所述第四状态,使用多智能体深度强化学习MADRL算法中的第一代价值函数和第二代价值函数确定所述第一边缘节点的代价值;

所述第一价值函数包括卸载开销函数和本地计算开销函数,其中,所述卸载开销函数用于确定所述一个或多个任务的卸载开销,所述本地计算开销函数用于计算所述一个或多个任务的本地开销;

所述第二价值函数包括平均价值函数和公平价值函数,其中,所述平均价值函数用于根据所述一个或多个任务的卸载开销和本地开销确定所述一个或多个任务的平均开销,所述公平价值函数用于根据使用所述第一边缘节点计算资源的第二终端设备的数量确定所述第一边缘节点的公平代价;

所述第一边缘节点根据所述一个或多个任务的平均开销与所述第一边缘节点的公平代价确定所述第一边缘节点的代价值。

12. 根据权利要求11所述的方法,其特征在于,所述第一边缘节点根据所述第一边缘节点的代价值确定所述第二卸载决策,包括:

所述第一边缘节点根据所述MADRL算法迭代更新所述第一边缘节点的状态和所述第一边缘节点的代价值;

在所述MADRL算法达到终止条件的情况下,所述第一边缘节点根据所述第一边缘节点的最小代价值确定所述第二卸载决策。

13. 一种通信装置,其特征在于,包括:

发送单元,用于向第一边缘节点发送第一计算任务的第一状态,其中,所述第一边缘节点为所述通信装置获取计算资源的边缘节点;

接收单元,用于接收所述第一边缘节点发送的第二卸载决策,所述第二卸载决策是由所述第一状态确定的,所述第二卸载决策包括一个或多个第二终端设备的计算资源分配信息,所述第二终端设备为从所述第一边缘节点获取计算资源的终端设备且所述通信装置为所述一个或多个第二终端设备中的一个终端设备;

处理单元,用于根据所述第二卸载决策确定所述第一计算任务的第一卸载决策,所述第一卸载决策用于指示是否卸载所述第一计算任务至所述第一边缘节点进行计算;

所述处理单元,具体用于根据所述第二卸载决策更新所述第一计算任务的第一状态得到所述第一计算任务的第二状态;

所述处理单元,还用于根据所述第二状态计算所述第一计算任务的代价值,所述第一计算任务的代价值包括所述第一计算任务的本地开销和卸载开销;

所述处理单元,还用于根据所述第一计算任务的代价值确定所述第一计算任务的第一卸载决策,

所述第一计算任务的本地开销包括第二能耗开销和第二时延开销,其中,

所述第二能耗开销包括所述通信装置本地计算所述第一计算任务消耗的能量和所述通信装置从休眠功率 $P_{\text{sleep}}$ 切换到第一工作功率 $P_1$ 消耗的能量;

所述第二时延开销包括所述通信装置本地计算所述第一计算任务的时延和所述通信装置从所述休眠功率切换到所述第一工作功率 $P_1$ 的时延;

所述第一工作功率 $P_1$ 为所述通信装置本地计算任务的工作功率。

14. 根据权利要求13所述的通信装置,其特征在于,

当所述第一卸载决策指示所述通信装置卸载所述第一计算任务至所述第一边缘节点进行计算时,所述发送单元,用于向所述第一边缘节点发送所述第一计算任务;

所述接收单元,用于接收所述第一边缘节点发送的所述第一计算任务的计算结果;或者

当所述第一卸载决策指示所述通信装置不卸载所述第一计算任务时,所述处理单元,用于在本地确定所述第一计算任务的计算结果。

15. 根据权利要求13所述的通信装置,其特征在于,

所述处理单元,具体用于根据所述第二状态,使用多智能体深度强化学习MADRL算法中的第一代价值函数确定所述第一计算任务的代价值,所述第一代价值函数包括卸载开销函数和本地计算开销函数,其中,所述卸载开销函数用于确定所述第一计算任务的卸载开销,所述本地计算开销函数用于确定所述第一计算任务的本地开销;以及

所述处理单元,还具体用于根据所述MADRL算法迭代更新所述通信装置的第一计算任务的状态和所述第一计算任务的代价值;

当所述MADRL算法达到终止条件时,所述处理单元,还用于根据所述第一计算任务的最小代价值确定所述第一计算任务的第一卸载决策。

16. 根据权利要求15所述的通信装置,其特征在于,

所述第一计算任务的卸载开销包括第一能耗开销和第一时延开销,其中,所述第一能耗开销包括所述通信装置将所述第一计算任务卸载至所述第一边缘节点消耗的能量,所述第一时延开销包括所述通信装置将所述第一计算任务卸载至所述第一边缘节点的时延以及所述第一边缘节点确定所述第一计算任务的计算结果的时延。

17. 根据权利要求16所述的通信装置,其特征在于,所述第一卸载决策还包括第二工作功率,所述第二工作功率为所述MADRL算法达到终止条件时,所述第一计算任务的最小代价值对应的工作功率。

18. 根据权利要求17所述的通信装置,其特征在于,

当所述第一卸载决策指示所述通信装置卸载所述第一计算任务至所述第一边缘节点进行计算时,所述处理单元,还用于以所述休眠功率工作。

19. 根据权利要求16所述的通信装置,其特征在于,

所述处理单元,还用于使用第一参数对所述第一时延开销进行动态调节,所述第一参数用于表示所述处理单元与所述第一边缘节点处理计算任务的技术差异。

20. 根据权利要求16至19中任一项所述的通信装置,其特征在于,

所述处理单元,还用于使用第二参数对所述第一能耗开销和所述第二能耗开销进行动态调节,所述第二参数用于表示所述处理单元对于能耗开销的敏感程度。

21.一种计算卸载的通信装置,其特征在于,包括:

接收单元,用于接收一个或多个任务的状态,所述一个或多个任务的状态包括第一终端设备发送的第一计算任务的第一状态,所述通信装置是为一个或多个第二终端设备获取计算资源的边缘节点,且所述第一终端设备为所述一个或多个第二终端设备中的一个终端设备;

处理单元,用于根据所述一个或多个任务状态,确定第二卸载决策,所述第二卸载决策包括所述处理单元对所述一个或多个第二终端设备的计算资源分配信息;

发送单元,用于向所述一个或多个第二终端设备广播所述第二卸载决策,所述第二卸载决策用于所述一个或多个第二终端设备确定所述第一计算任务的第一卸载决策,所述第一卸载决策用于指示所述第一终端设备是否卸载所述第一计算任务至所述通信装置进行计算;

所述处理单元,具体用于根据所述一个或多个任务的状态更新第一边缘节点的第三状态得到所述第一边缘节点的第四状态,其中,所述第三状态是所述第一边缘节点接收到所述一个或多个任务状态之前的状态;

所述处理单元,还用于根据所述第四状态确定所述通信装置的代价值,所述代价值为所述处理单元为所述一个或多个计算任务分配计算资源的开销;

所述处理单元,根据所述代价值确定所述第二卸载决策。

22.根据权利要求21所述的通信装置,其特征在于,

所述接收单元,还用于接收所述第一终端设备发送的所述第一计算任务;

所述处理单元,还用于确定所述第一计算任务的计算结果;

所述发送单元,还用于向所述第一终端设备发送所述第一计算任务的计算结果。

23.根据权利要求21所述的通信装置,其特征在于,

所述处理单元,具体用于根据所述第四状态,使用多智能体深度强化学习MADRL算法中的第一代价值函数和第二代价值函数确定所述代价值;

所述第一代价值函数包括卸载开销函数和本地计算开销函数,其中,所述卸载开销函数用于确定所述一个或多个任务的卸载开销,所述本地计算开销函数用于计算所述一个或多个任务的本地开销;

所述第二代价值函数包括平均代价函数和公平代价函数,其中,所述平均代价函数用于根据所述一个或多个任务的卸载开销和本地开销确定所述一个或多个任务的平均开销,所述公平代价函数用于根据使用所述通信装置计算资源的第二终端设备的数量确定所述通信装置的公平代价;

所述处理单元,具体用于根据所述一个或多个任务的平均开销与所述公平代价确定所述通信装置的代价值。

24.根据权利要求23所述的通信装置,其特征在于,

所述处理单元,具体用于根据所述MADRL算法迭代更新所述第一边缘节点的状态和所述第一边缘节点的代价值;

在所述MADRL算法达到终止条件的情况下,所述处理单元,还用于根据所述第一边缘节

点的最小代价值确定所述第二卸载决策。

25. 一种通信装置,其特征在于,包括处理器和接口电路,其中,所述接口电路用于接收计算机代码或指令并将其传输至所述处理器,所述处理器运行所述计算机代码或指令,使得如权利要求1-8中任一项所述的方法被执行,或者,使得如权利要求9-12中任一项所述的方法被执行。

26. 一种通信装置,其特征在于,包括至少一个处理器,所述至少一个处理器与至少一个存储器耦合,所述至少一个处理器用于执行所述至少一个存储器中存储的计算机程序或指令,当所述计算机程序或指令被执行时,如权利要求1-8中任一项所述的方法被执行,或者,如权利要求9-12中任一项所述的方法被执行。

27. 一种计算机可读存储介质,其特征在于,所述计算机可读存储介质中存储有计算机指令,当所述计算机指令被运行时,如权利要求1-8中任一项所述的方法被执行,或者,如权利要求9-12中任一项所述的方法被执行。

## 计算卸载的方法和通信装置

### 技术领域

[0001] 本申请涉及通信领域,具体涉及一种计算卸载的方法和通信装置。

### 背景技术

[0002] 随着互联网方兴未艾,终端设备更新换代,新型应用不断涌现。与此同时,终端设备的计算能力逐渐难以满足用户业务的性能需求。尽管云计算的出现一定程度上缓解了终端的资源限制,然而在业务延迟、传输速率与数据安全等方面仍尚待提升,特别是对于诸如自动驾驶、实时视频监控等高实时高安全的应用上,这促使着云计算模式产生新的变革。

[0003] 作为云计算的一种概念演进,边缘计算提出在网络边缘处提供计算与存储资源,具有延迟更低、能耗更低、节省带宽、隐私性强、更智能化等特点,并产生了诸如微云(Cloudlet)、雾计算(fog computing)、移动边缘计算(mobile edge computing)、多接入边缘计算(multi-access edge computing,MEC)等衍生概念。作为实现物联网与第五代移动通信技术(5th generation,5G)愿景的关键技术,边缘计算(及其衍生概念)处于计算模式与网络通信的交叉点,对满足种类繁多的业务需求与精益求精的用户体验具有不可替代的作用。

[0004] 与云计算中的计算卸载方法类似,边缘计算是指终端设备将计算密集的任务卸载(offload)到网络边缘节点进行计算,并接受对应的任务的运行结果以达到计算卸载的目的。具体来说,面向边缘计算有两方面需要具体考虑:一方面,终端设备存在着运算能力较低、能量敏感度高、支持功能较少等不足;另一方面,网络边缘节点也暴露出计算资源分散、供电方式不同、系统架构多样等现象。因此,如何更好地管理与分配边缘节点的计算能力成为亟需关注的重要问题。

### 发明内容

[0005] 本申请提供一种计算卸载的方法和通信装置,能够使边缘节点和终端设备在实际决策中对于环境有更广泛的感知能力,从而有效地提升两者的决策收益。

[0006] 第一方面,提供了一种方法,该方法包括:第一终端设备向第一边缘节点发送第一计算任务的第一状态,其中,第一边缘节点为第一终端设备获取计算资源的边缘节点,所述第一状态包括传输所述第一计算任务的数据流的长度、计算所述第一计算任务所需耗费的时钟周期数、所述第一计算任务的惩罚值中的至少一个;第一终端设备接收第一边缘节点发送的第二卸载决策,第二卸载决策是由第一状态确定的,第二卸载决策包括一个或多个第二终端设备的计算资源分配信息,第二终端设备为从第一边缘节点获取计算资源的终端设备且第一终端设备为一个或多个第二终端设备中的一个终端设备;第一终端设备根据第二卸载决策确定第一计算任务的第一卸载决策,第一卸载决策用于指示第一终端设备是否卸载第一计算任务至第一边缘节点进行计算。

[0007] 上述技术方案中,终端设备向边缘节点发送计算任务,边缘节点根据接收到的一个或多个计算任务的状态确定计算资源的分配情况,然后边缘节点向所服务的终端设备发



送计算资源的分配情况,终端设备根据该资源分配情况确定是否将计算任务卸载至边缘节点进行计算,能够使终端设备能够在实际的决策中对于环境有更广泛的感知能力,从而有效地提升两者的决策收益。

[0008] 结合第一方面,在第一方面的某些实现方式中,当第一卸载决策指示第一终端设备卸载第一计算任务至第一边缘节点进行计算时,第一终端设备向第一边缘节点发送第一计算任务;第一终端设备接收第一边缘节点发送的第一计算任务的计算结果;或者当第一卸载决策指示第一终端设备不卸载所述第一计算任务时,第一终端设备在本地确定第一计算任务的计算结果。

[0009] 结合第一方面,在第一方面的某些实现方式中,第一终端设备根据第二卸载决策确定第一计算任务的第一卸载决策,包括:第一终端设备根据第二卸载决策,更新第一计算任务的第一状态中的参数得到第一计算任务的第二状态;第一终端设备根据第二状态计算第一计算任务的代价值,第一计算任务的代价值包括第一计算任务的本地开销和卸载开销;第一终端设备根据第一计算任务的代价值确定第一计算任务的第一卸载决策。

[0010] 上述技术方案中,终端设备根据边缘节点的资源分配情况确定是否将计算任务卸载至边缘节点进行计算,能够使终端设备能够在实际的决策中对于环境有更广泛的感知能力,从而有效地提升两者的决策收益。

[0011] 结合第一方面,在第一方面的某些实现方式中,第一终端设备根据第二状态计算第一计算任务的代价值,包括:第一终端设备根据第二状态,使用多智能体深度强化学习MADRL算法中的第一代价值函数确定第一计算任务的代价值,第一代价值函数包括卸载开销函数和本地计算开销函数,其中,卸载开销函数用于确定第一计算任务的卸载开销,本地计算开销函数用于确定第一计算任务的本地开销;以及第一终端设备根据第一计算任务的代价值确定第一计算任务的第一卸载决策,包括:第一终端设备根据MADRL算法迭代更新第一终端设备的第一计算任务的状态和第一计算任务的代价值;当MADRL算法达到终止条件时,第一终端设备根据第一计算任务的最小代价值确定第一计算任务的第一卸载决策。

[0012] 上述技术方案中,终端设备根据边缘节点的资源分配情况使用MADRL算法确定是否将计算任务卸载至边缘节点进行计算,能够使终端设备能够在实际的决策中对于环境有更广泛的感知能力,从而有效地提升两者的决策收益。

[0013] 结合第一方面,在第一方面的某些实现方式中,第一计算任务的卸载开销包括第一能耗开销和第一时延开销,其中,第一能耗开销包括第一终端设备将第一计算任务卸载至第一边缘消耗的能量,第一时延开销包括第一终端设备将第一计算任务卸载至第一边缘节点的时延以及第一边缘节点确定第一计算任务的计算结果的时延。

[0014] 结合第一方面,在第一方面的某些实现方式中,第一计算任务的本地开销包括第二能耗开销和第二时延开销,其中,第二能耗开销包括第一终端设备本地计算第一计算任务消耗的能量和第一终端设备从休眠功率 $P_{\text{sleep}}$ 切换到第一工作功率 $P_1$ 消耗的能量,第一时延开销包括第一终端设备本地计算第一计算任务的时延和第一终端设备从休眠功率切换到第一工作功率 $P_1$ 的时延,第一工作功率 $P_1$ 为第一终端设备本地计算任务的工作功率。

[0015] 上述技术方案中,在终端设备本地处理计算任务的代价值计算中,引入功率控制因素,在终端设备代价值计算中增加了功率切换时的代价损失,这有助于更为精细地描述终端设备的计算卸载中的决策内容,进而有效地提高用户体验、减少模型误差。

[0016] 结合第一方面,在第一方面的某些实现方式中,第一卸载决策还包括第二工作功率,第二工作功率为MADRL算法达到终止条件时,第一计算任务的最小代价值对应的工作功率。

[0017] 结合第一方面,在第一方面的某些实现方式中,当第一卸载决策指示第一终端设备卸载第一计算任务至第一边缘节点进行计算时,第一终端设备以休眠功率工作。

[0018] 结合第一方面,在第一方面的某些实现方式中,方法还包括:第一终端设备使用第一参数对第一时延开销进行动态调节,第一参数用于表示第一终端设备与第一边缘节点处理计算任务的技术差异。

[0019] 上述技术方案中,引入技术差异参数,能够更为精细地描述计算卸载中的决策内容。

[0020] 结合第一方面,在第一方面的某些实现方式中,方法还包括:第一终端设备使用第二参数对第一能耗开销和第二能耗开销进行动态调节,第二参数用于表示第一终端设备对于能耗开销的敏感程度。

[0021] 上述技术方案中,引入能量权衡参数,能够更为精细地描述计算卸载中的决策内容。

[0022] 第二方面,提供了一种方法,该方法包括:第一边缘节点接收一个或多个任务的状态,一个或多个任务的状态包括第一终端设备发送的第一计算任务的第一状态,第一边缘节点是为一个或多个第二终端设备提供计算资源的边缘节点,且第一终端设备为一个或多个第二终端设备中的一个终端设备;第一边缘节点根据一个或多个任务状态,确定第二卸载决策,第二卸载决策包括第一边缘节点对一个或多个第二终端设备的计算资源分配信息;第一边缘节点向一个或多个第二终端设备广播第二卸载决策。

[0023] 上述技术方案中,边缘节点接收所服务的一个或多个终端设备发送的计算任务的状态,并根据接收到的一个或多个计算任务的状态确定计算资源的分配情况,然后边缘节点向所服务的终端设备广播计算资源的分配情况,终端设备根据该资源分配情况确定是否将计算任务卸载至边缘节点进行计算,从而使得边缘节点和终端设备能够在实际决策中对于环境有更广泛的感知能力,从而有效地提升两者的决策收益。

[0024] 结合第二方面,在第二方面的某些实现方式中,第一边缘节点接收第一终端设备发送的第一计算任务;第一边缘节点确定第一计算任务的计算结果;第一边缘节点向第一终端设备发送第一计算任务的计算结果。

[0025] 结合第二方面,在第二方面的某些实现方式中,第一边缘节点根据一个或多个任务的状态确定第二卸载决策,包括:第一边缘节点根据一个或多个任务的状态更新第一边缘节点的第三状态得到第一边缘节点的第四状态,其中,第三状态是第一边缘节点接收到一个或多个任务状态之前的状态;第一边缘节点根据第四状态确定第一边缘节点的代价值,第一边缘节点的代价值为第一边缘节点为一个或多个计算任务分配计算资源的开销;第一边缘节点根据第一边缘节点的代价值确定第二卸载决策。

[0026] 结合第二方面,在第二方面的某些实现方式中,第一边缘节点根据第四状态确定第一边缘节点的代价值,包括:第一边缘节点根据第四状态,使用多智能体深度强化学习MADRL算法中的第一代价值函数和第二代价值函数确定第一边缘节点的代价值;第一代价值函数包括卸载开销函数和本地计算开销函数,其中,卸载开销函数用于确定一个或多个任务的

卸载开销,本地计算开销函数用于计算一个或多个任务的本地开销;第二代价函数包括平均代价函数和公平代价函数,其中,平均代价函数用于根据一个或多个任务的卸载开销和本地开销确定一个或多个任务的平均开销,公平代价函数用于根据使用第一边缘节点计算资源的第二终端设备的数量确定第一边缘节点的公平代价;第一边缘节点根据一个或多个任务的平均开销与第一边缘节点的公平代价确定第一边缘节点的代价值。

[0027] 上述技术方案中,边缘节点的代价值权衡了多种因素,边缘节点决策项保证了所服务的终端设备的平均体验,同时提升了终端设备在资源利用上的公平性,即在保证资源高效分配的同时也避免了服务用户的数量过少,使边缘节点的代价值反映的现实环境更为全面。

[0028] 结合第二方面,在第二方面的某些实现方式中,第一边缘节点根据第一边缘节点的代价值确定第二卸载决策,包括:第一边缘节点根据MADRL算法迭代更新第一边缘节点的状态和第一边缘节点的代价值;在MADRL算法达到终止条件的情况下,第一边缘节点根据第一边缘节点的最小代价值确定第二卸载决策。

[0029] 第三方面,本申请提供了一种通信装置,所述通信装置具有实现第一方面或其任意可能的实现方式中的方法的功能。所述功能可以通过硬件实现,也可以通过硬件执行相应的软件实现。所述硬件或软件包括一个或多个与上述功能相对应的单元。

[0030] 第四方面,本申请提供了一种通信装置,所述通信装置具有实现第二方面或其任意可能的实现方式中的方法的功能。所述功能可以通过硬件实现,也可以通过硬件执行相应的软件实现。所述硬件或软件包括一个或多个与上述功能相对应的单元。

[0031] 第五方面,本申请提供一种通信装置,包括接口电路和处理器,所述接口电路用于接收计算机代码或指令,并传输至所述处理器,所述处理器运行所述计算机代码或指令,第一方面或其任意实现方式中的方法被实现。

[0032] 第六方面,本申请提供一种通信装置,包括接口电路和处理器,所述接口电路用于接收计算机代码或指令,并传输至所述处理器,所述处理器运行所述计算机代码或指令,第二方面或其任意实现方式中的方法被实现。

[0033] 第七方面,本申请提供一种通信设备,包括至少一个处理器,所述至少一个处理器与至少一个存储器耦合,所述至少一个存储器用于存储计算机程序或指令,所述至少一个处理器用于从所述至少一个存储器中调用并运行该计算机程序或指令,使得通信设备执行第一方面或其任意可能的实现方式中的方法。

[0034] 在一个示例中,所述通信设备可以为终端设备。

[0035] 第八方面,本申请提供一种通信设备,包括至少一个处理器,所述至少一个处理器与至少一个存储器耦合,所述至少一个存储器用于存储计算机程序或指令,所述至少一个处理器用于从所述至少一个存储器中调用并运行该计算机程序或指令,使得通信设备执行第二方面或其任意可能的实现方式中的方法。

[0036] 在一个示例中,所述通信设备可以为边缘节点。

[0037] 第九方面,本申请提供一种计算机可读存储介质,所述计算机可读存储介质中存储有计算机指令,当计算机指令在计算机上运行时,所述第一方面或其任意可能的实现方式中的方法被实现。

[0038] 第十方面,本申请提供一种计算机可读存储介质,所述计算机可读存储介质中存

储有计算机指令,当计算机指令在计算机上运行时,所述第二方面或其任意可能的实现方式中的方法被实现。

[0039] 第十一方面,本申请提供一种计算机程序产品,所述计算机程序产品包括计算机程序代码,当所述计算机程序代码在计算机上运行时,所述第一方面或其任意可能的实现方式中的方法被实现。

[0040] 第十二方面,本申请提供一种计算机程序产品,所述计算机程序产品包括计算机程序代码,当所述计算机程序代码在计算机上运行时,所述第二方面或其任意可能的实现方式中的方法被实现。

[0041] 第十三方面,本申请提供一种无线通信系统,包括如第七方面的通信设备和第八方面所述通信设备。

### 附图说明

[0042] 图1示出了适用于本申请实施例的系统架构的示意图。

[0043] 图2是强化学习的训练过程示意图。

[0044] 图3是一种基于DRL的计算卸载方法的示意图。

[0045] 图4是MADRL中智能体与环境的交互过程的示意图。

[0046] 图5为本申请提供的计算卸载的方法的示意性流程图。

[0047] 图6为本申请提供的计算卸载的方法的示意性流程图。

[0048] 图7是本申请提供的基于MADDPG的数据采集的示意性流程图。

[0049] 图8是本申请提供的基于MADDPG的参数模型训练的示意性流程图。

[0050] 图9是本申请提供的基于MADDPG的计算卸载的方法的示意性流程图。

[0051] 图10为本申请提供的通信装置1000的示意性框图。

[0052] 图11为本申请提供的通信装置2000的示意性框图。

[0053] 图12为本申请提供的通信装置10的示意性结构图。

[0054] 图13为本申请提供的通信装置20的示意性结构图。

### 具体实施方式

[0055] 下面将结合附图,对本申请中的技术方案进行描述。

[0056] 本申请实施例的技术方案可以应用于各种通信系统,例如:长期演进(long term evolution,LTE)系统、LTE频分双工(frequency division duplex,FDD)系统、LTE时分双工(time division duplex,TDD)、通用移动通信系统(universal mobile telecommunication system,UMTS)、新无线(new radio,NR)系统等第五代(5th generation,5G)系统,卫星通信系统,以及其它未来演进的通信系统,车到其它设备(vehicle-to-X V2X),其中V2X可以包括车到互联网(vehicle to network,V2N)、车到车(vehicle to-vehicle,V2V)、车到基础设施(vehicle to infrastructure,V2I)、车到行人(vehicle to pedestrian,V2P)等、车间通信长期演进技术(long term evolution-vehicle,LTE-V)、车联网、机器类通信(machine type communication,MTC)、物联网(internet of things,IoT)、机器间通信长期演进技术(long term evolution-machine,LTE-M)、机器到机器(machine to machine,M2M)等。

[0057] 参见图1,图1示出了适用于本申请实施例的系统架构的示意图。如图1所示,该系统架构中包括至少一个终端设备、至少一个无线接入节点和至少一个边缘节点。终端设备通过多种无线接入方式(包括但不限于蜂窝与Wi-Fi)与对应的无线接入点,访问到位于互联网边缘(即网络边缘)节点组成的资源池(即边缘节点构成的资源池)。

[0058] 边缘节点,即表示在靠近终端设备的网络边缘侧构建的业务平台,提供存储、计算、网络等资源,将部分关键业务应用下沉到接入网络边缘,以减少网络传输和多级转发带来的宽度和时延损耗。本申请将网络边缘中具有计算能力的设备统一抽象为边缘节点。但本申请本身不涉及边缘节点内计算资源的抽象、管理与分配功能,仅指出边缘节点以统一的整体向终端设备分配计算资源的策略。

[0059] 应理解,图1所示的系统架构中的终端设备、边缘节点与无线信道,根据现实情况有如下规定:在每个时刻,单个终端设备仅能接入一条无线信道,并通过该信道获取到一个边缘节点的计算资源;单个无线接入点能够以此访问到其中的一个(或多个)边缘节点,并固定提供一条无线信道给终端设备接入;单个边缘节点能够同时利用多个无线接入点向终端设备提供计算资源,并可以同时为多个终端设备提供计算资源。为了方便下文详细说明,有如下定义:终端设备 $D$ 、边缘节点 $N$ 、无线信道 $Z$ 可分别表示为 $D = \{1, 2, \dots, d\}$ ,  $i \in D$ ,  $N = \{1, 2, \dots, n\}$ ,  $j \in N$ ,  $Z = \{1, 2, \dots, z\}$ ,  $k \in Z$ ;终端设备连接到的边缘节点与通信的无线信道分别为: $\text{conn}_i \in \{0\} \cup N$ 和 $\text{link}_i \in \{0\} \cup Z$ 。

[0060] 本申请实施例中的终端设备也可以称为:用户设备(user equipment, UE)、移动台(mobile station, MS)、移动终端(mobile terminal, MT)、接入终端、用户单元、用户站、移动站、移动台、远方站、远程终端、移动设备、用户终端、终端、无线通信设备、用户代理或用户装置等。

[0061] 终端设备可以是一种向用户提供语音/数据连通性的设备,例如,具有无线连接功能的手持式设备、车载设备等。目前,一些终端设备的举例为:手机(mobile phone)、平板电脑、笔记本电脑、掌上电脑、移动互联网设备(mobile internet device, MID)、可穿戴设备,虚拟现实(virtual reality, VR)设备、增强现实(augmented reality, AR)设备、工业控制(industrial control)中的无线终端、无人驾驶(self driving)中的无线终端、远程手术(remote medical surgery)中的无线终端、智能电网(smart grid)中的无线终端、运输安全(transportation safety)中的无线终端、智慧城市(smart city)中的无线终端、智慧家庭(smart home)中的无线终端、蜂窝电话、无绳电话、会话启动协议(session initiation protocol, SIP)电话、无线本地环路(wireless local loop, WLL)站、个人数字助理(personal digital assistant, PDA)、具有无线通信功能的手持设备、计算设备或连接到无线调制解调器的其它处理设备、车载设备、可穿戴设备,未来6G网络中的终端设备或者未来演进的公用陆地移动通信网络(public land mobile network, PLMN)中的终端设备和/或用于在无线通信系统上通信的任意其它适合设备,本申请实施例对此并不限定。

[0062] 其中,可穿戴设备也可以称为穿戴式智能设备,是应用穿戴式技术对日常穿戴进行智能化设计、开发出可以穿戴的设备的总称,如眼镜、手套、手表、服饰及鞋等。可穿戴设备即直接穿在身上,或是整合到用户的衣服或配件的一种便携式设备。可穿戴设备不仅仅是一种硬件设备,更是通过软件支持以及数据交互、云端交互来实现强大的功能。广义穿戴式智能设备包括功能全、尺寸大、可不依赖智能手机实现完整或者部分的功能,例如:智能

手表或智能眼镜等,以及只专注于某一类应用功能,需要和其它设备如智能手机配合使用,如各类进行体征监测的智能手环、智能首饰等。

[0063] 此外,在本申请实施例中,终端设备还可以是物联网系统中的终端设备,IoT是未来信息技术发展的重要组成部分,其主要技术特点是将物品通过通信技术与网络连接,从而实现人机互连,物物互连的智能化网络。

[0064] 此外,在本申请实施例中,终端设备还可以包括智能打印机、火车探测器、加油站等传感器,主要功能包括收集数据(部分终端设备)、接收网络设备的控制信息与下行数据,并发送电磁波,向网络设备传输上行数据。

[0065] 为便于理解本申请实施例,首先对本申请中涉及到的术语作简单说明。

[0066] 1、边缘计算:从“云计算”这一概念中演进而出,是一种新型的分布式计算模式。具体定义为:在互联网边缘向用户提供计算与存储资源。其中,“边缘”被定义为终端设备与云数据中心之间路径上的任何网络位置,这些位置相比云数据中心更靠近于用户。一般来说,“边缘计算”是指无线链路和有线网络的交界附近,即在无线接入点附近部署资源。边缘计算具有延迟更低、能耗更低、节省带宽、隐私性强、更智能化等特点,作为实现物联网与5G愿景的关键使能技术,边缘计算(及其衍生概念)处于计算模式与网络通信的交叉点,对满足种类繁多的业务需求与精益求精的用户体验具有不可替代的作用。

[0067] 2、计算卸载:是指计算设备将资源密集型的计算任务转移到单独的处理器或其他设备的行为。在本申请中,计算设备即是指终端设备,资源密集型的计算任务是是需要花费一定计算才能完成的任务,而转移的位置则是网络边缘处的边缘节点。具体地,其典型过程为:终端设备将计算任务卸载到边缘节点,边缘节点对计算任务进行处理,终端设备接收边缘节点相应任务的计算结果。

[0068] 与云计算中的计算卸载方法类似,具体的,可能的一种利用边缘计算提升终端性能的方式即是:终端设备将计算密集的任务卸载(offload)到网络边缘执行,并接受对应的运行结果以达到计算卸载的目的。具体来说,面向边缘计算两方面需要具体考虑:一方面,终端设备存在着运算能力较低、能量敏感度高、支持功能较少等不足;另一方面,网络边缘也暴露出计算资源分散、供电方式不同、系统架构多样等现象。因此,如何管理与分配网络边缘的计算能力成为亟需关注的重要问题。

[0069] 3、深度强化学习(deep reinforcement learning,DRL):强化学习是机器学习中的一个领域。参见图2,图2是强化学习的训练过程示意图。如图2所示,强化学习主要包含四个元素:智能体(agent)、环境(environment)、状态(state)、动作(action)与奖励(reward),其中,智能体的输入为状态,输出为动作。当前技术中,强化学习的训练过程为:通过智能体与环境进行多次交互,获得每次交互的动作、状态、奖励;将这多组(动作,状态,奖励)作为训练数据,对智能体进行一次训练。采用上述过程,对智能体进行下一轮次训练,直至满足收敛条件。其中,获得一次交互的动作、状态、奖励的过程如图1所示,将环境当前状态 $s(t)$ 输入至智能体,获得智能体输出的动作 $a(t)$ ,根据环境在动作 $a(t)$ 作用下的相关性能指标计算本次交互的奖励 $r(t)$ ,至此,获得本次交互的动作 $a(t)$ 、动作 $a(t)$ 与奖励 $r(t)$ 。记录本次交互的动作 $a(t)$ 、动作 $a(t)$ 与奖励 $r(t)$ ,以备后续用来训练智能体。还记录环境在动作 $a(t)$ 作用下的下一个状态 $s(t+1)$ ,以便实现智能体与环境的下一次交互。将强化学习和深度学习相结合,就得到了深度强化学习。深度强化学习仍然符合强化学习中主体

和环境交互的框架。不同的是,智能体(agent)中使用深度神经网络进行决策。深度强化学习是一种综合了深度学习与强化学习的人工智能方法,在诸如动态决策、实时控制、图像感知等复杂问题中具有广泛的应用,并产生了一些基于DRL的计算卸载方法。其中,针对计算卸载的已有方法中涉及的算法包括且不限于:深度Q网络(deepQ-learning,DQN)、深度确定性策略梯度(deep deterministic policy gradient,DDPG)、Actor-Critic(演员-评委)算法。基于DRL的方法模型中主要包括智能体和与其交互的环境。其中,终端设备被视作决策与执行的实体,其采用深度神经网络来感知环境的当前状态,并通过类似强化学习的方式与环境互动并调整策略,最后经过迭代多次的感知-调整过程逐渐向最优策略靠拢。

[0070] 4、多智能体深度强化学习(multi-agent deep reinforcement learning, MADRL):是一种融合了多智能体系统中深度学习与强化学习的方法,其中,深度学习能够利用深度神经网络从信息中有效地提取特征,强化学习通过与环境的动态交互以不断地强化决策能力,多智能体系统则强调智能体在关注环境因素的同时也关注智能体之间决策的相互影响。因此,MADRL适用于描述复杂环境下多个智能体决策与交互的过程,在诸如机器人协作、分布式控制、协同决策等诸多领域有着较为广泛的应用。具体地,MADRL针对特定问题构建了一个包含多个智能体的系统,其中每个智能体都使用深度强化学习(deep reinforcement learning,DRL)方法描述其决策与交互过程。相比于通常的(单智能体)深度强化学习,多智能体系统中每个智能体的最优解不仅受到环境变量的制约,也同时受到其它智能体行为的约束与影响。

[0071] 5、多智能体深度确定性策略梯度(multi-agent deep deterministic policy gradient,MADDPG)算法:是一种典型的多智能体深度强化学习算法,是深度确定性策略梯度(deep deterministic policy gradient,DDPG)算法在多智能体中的延伸与拓展,其中每个智能体分别运行一个DDPG模型。MADDPG算法通过集中式训练、分布式执行的方法实现多智能体的决策过程,在训练时需要知晓其他智能体的决策信息,而执行时仅需局部信息即可做出决策。

[0072] 6、计算功率控制:是指终端设备在处理计算任务的过程中调整计算功率以权衡业务延迟与能量耗费的行为。由于边缘计算中终端设备是能量受限的,并且在能耗敏感程度上具有差异,因此控制本地处理任务的计算功率能够有效提升用户体验。

[0073] 本申请中边缘计算卸载问题中的决策实体包括两种:终端设备(计算卸载方)与边缘节点(资源提供方)。其中,终端设备在生成计算密集的任务时将决定是在终端设备本地处理该任务(即不执行计算卸载),还是将计算任务打包通过网络传输给边缘节点,并接收边缘节点处理任务后返回的结果。边缘节点则需要根据当前的环境情况与终端状态,动态地统计当前可用资源并将计算资源分配给需要的终端设备。资源受限的终端设备需要根据无线环境、计算能力与能量模式等当前状态,确定终端设备中待处理的计算任务与可用的计算资源之间的对应关系,即决策在何处、以怎样的方式执行哪个计算任务。一方面,本地执行避免了在网络中传输任务而造成的开销;另一方面,计算卸载则减少了执行计算任务的时间与能耗。

[0074] 参见图3,图3是一种基于DRL的计算卸载方法的示意图。

[0075] 以Actor-Critic算法为例,在一个无线基站的覆盖范围中存在着多个终端设备,并且每个终端设备都存在着一个计算密集且延迟敏感的任务需要完成。终端设备可以选择

将计算任务在本地进行处理,也可以选择通过基站将计算任务卸载到远端。由于计算卸载过程中的无线传输速率受到背景噪声与其它终端设备的通信干扰,因此终端设备需要权衡本机处理与计算卸载的效益大小进行决策。这里简单的介绍一下该Actor-Critic算法的计算流程:当终端设备运行Actor-Critic算法时,将会同时维护两个神经网络Actor与Critic,并分别对应着两套策略参数。其中,Actor的输入为当前状态,输出为当前状态下能够获得最高奖励的策略动作,Critic的输入则为当前状态、环境返回的下一个状态与奖励,输出为环境返回奖励与自身估计的奖励之间的误差(TD\_error)。

[0076] 结合图3,具体地,Actor-Critic算法的训练过程如下所述:(1) Actor首先根据环境当前状态S决策行动A。(2) 环境根据状态S与行动A返回下一个状态S'与奖励R。(3) Critic根据当前状态S、下一个状态S'与奖励R得到估计误差TD\_error,并根据该误差调整自身的策略参数;(4) Actor根据状态S、行动A与误差TD\_error也调整自身的策略参数;(5) 终端将当前状态S赋值为下一个状态S',即 $S=S'$ ,重复步骤(1)到(4)的过程继续训练。相对应地,基于Actor-Critic的求解方法在具体执行时,仅需使用Actor即可根据环境当前状态S决定是否卸载的动作A。

[0077] 上述单智能体DRL假设决策环境是稳定的,不会因为决策本身而发生改变,这在多个终端设备同时决策是否计算卸载时是难以成立的。具体地,是否计算卸载这一决策对应着终端设备在无线信道与外部计算资源上的竞争关系,同时会影响到决策奖励。因此,对于每个终端设备来说,自身的决策过程都可能会导致其它终端设备的决策变化,进而引发DRL中环境的动态变化。而当DRL对于环境稳定的假设不成立时,算法收敛将会不再稳定,进而造成求解方法的性能退化。

[0078] 此外,基于DRL的已有方法着重考虑了智能体与环境间的交互过程,而较少关注智能体与智能体之间的交互机制与信息内容,这使得智能体难以合理地从环境中获取到更多的有效信息,进而限制决策收益的进一步提升。

[0079] 参见图4,图4是MADRL中智能体与环境的交互过程的示意图。与单智能体DRL类似,MADRL中的每个智能体仍旧采用了一套DRL的方法,即包含着一个感知与控制的系统,并由五个部分组成:智能体、决策动作、状态、奖励与环境。智能体是系统中的决策实体,图4中的输入S'是反映当前环境对于智能体的一个状态,输出A是给定状态下该智能体所执行的动作。所有的智能体共享着同一个环境,但动作、状态与奖励往往不同。环境用于与多个智能体进行交互,维护着所有智能体的当前状态集合,向智能体分别展示相应的当前状态,并根据所有智能体的动作产生对应的奖励与下一个状态集合。

[0080] MADRL的训练过程即是求取给定环境中智能体最优策略的过程,该过程通过上述多个智能体与环境的不断交互来实现:每个智能体从环境中获取到一个高维度的观测状态S,利用类似深度学习(deep learning, DL)的方法来感知并分析该观测状态,随后根据预期回报来评价各个可行动作,并根据策略在当前状态下做出决策动作,最后环境对所有智能体的决策动作统一做出反应,从而得到下一个状态集合S'与相应的奖励集合R。通过不断循环以上过程,多智能体最终可以得到给定环境下的最优策略。

[0081] 特别地,与单智能体DRL的不同之处在于,MADRL中的策略学习是一种集中式的过程。每个智能体在决策时都能够知晓所有智能体对于当前观测状态的策略,并可以获知所有智能体的决策奖励集合R与下一个状态集合,这样就能够避免学习过程中智能体决策相



互影响造成的非平稳性。在MADRL的策略推断过程中,每个智能体根据自身对于环境的观测状态独立地做出决策。以上所述的策略学习与推断过程是一种类似于群体行为的人工智能方法。

[0082] 为了使智能体(即终端设备)能从环境中获取到更多的有效信息,更好地决策是否卸载计算任务这一基础决策,本申请提供一种基于多智能体深度强化学习的计算卸载的方法,具体的,通过优化并权衡业务延迟与能量耗费这两个主要指标,从而能够进一步提升计算卸载任务的决策收益,达到增强用户体验与提高资源效用的目的。

[0083] 由上可知,在终端设备执行计算卸载的过程中,终端设备的收益由两个阶段的性能表现所构成:网络传输与任务计算,其中,任务计算的消耗时间受到边缘节点的资源分配所影响,而网络传输的处理代价则由无线链路传输与有线链路传输两部分构成。考虑到端-边(终端设备-边缘节点)交互中的无线网络环境更为复杂,受到环境影响与传输干扰更为严重,因此本申请关注无线链路中的环境噪声与带宽争用现象。其中,环境噪声的范围包括背景噪声与不参与计算卸载决策的其它设备在该信道上造成的通信干扰,带宽争用的范围包括所有使用该信道并参与计算卸载决策的终端设备相互的通信干扰。此外,由于当前的无线环境设计与部署时已经充分考虑到了接入点信号的相互干扰,因此本申请默认接入点的部署是合理的,这使得终端设备彼此之间的通信干扰成为影响网络传输速率的主要因素。在本申请的典型场景中,计算卸载过程的处理时间通常在秒级及以下,在这种时间粒度中终端设备的移动对于边缘节点与无线接入点的连接状态是可以预期的,因此本申请在考虑计算卸载决策时默认计算卸载的决策与执行过程中终端设备与边缘节点、终端设备与无线接入点的链路状态不发生改变。

[0084] 参见图5,图5为本申请提供的计算卸载方法的示意性流程图。

[0085] S510,第一终端设备向第一边缘节点发送第一计算任务的第一状态。

[0086] 对应的,第一边缘节点接收第一终端设备发送的第一状态。

[0087] 需要说明的是,由于一个边缘节点往往会服务一个或多个的终端设备,所以第一边缘节点除了收到第一终端设备发送第一状态信息,还可能会接收到其他服务的终端设备发送的计算任务的状态。例如:第一边缘节点为边缘节点 $j$ ,那么边缘节点 $j$ 服务的终端设备可以表示为: $i \in D$ 且 $conn_i = j$ (即第二终端设备的一例)。

[0088] 本申请中,任意一个终端设备的计算任务的状态可以包括但不限于:计算任务的数据流长度、计算任务需耗费的时间、计算任务的惩罚值、以及其他计算任务相关的参数等。

[0089] 作为示例而非限定,本申请中使用以下所述的方式定义计算任务的状态。每个特定的计算任务状态能够表示为: $M = \{mt, mc, md, mp\}$ 。其中, $mt$ 代表传输计算任务时所需的数据流长度,单位为字节; $mc$ 代表处理该计算任务所需要耗费的时钟周期数,单位为赫兹; $md$ 代表计算任务从开始执行到得到计算结果终端设备所能接受的最长时间, $md$ 是一个与任务相关的固定值,在任务产生的时候由任务产生的应用给出该固定值,单位为秒; $mp$ 代表当无法成功处理该计算任务时造成的惩罚值,是终端设备与边缘节点计算收益时的一部分。例如:第一计算任务的第一状态可表示为 $M_1 = \{mt_1, mc_1, md_1, mp_1\}$ 。

[0090] 第一边缘节点根据接收到的一个或多个计算任务的状态确定第二卸载决策,第二卸载决策包括第一边缘节点对所服务的终端设备的计算资源的分配信息。

[0091] 可选的,第一边缘节点可以根据接收到的一个或多个的计算任务的状态使用MADRL算法确定第二卸载决策。关于MADRL算法具体的决策过程这里暂不展开叙述,在图6对应的流程图中会做详细的介绍。

[0092] S520,第一终端设备接收第一边缘节点发送的第二卸载决策。

[0093] 对应的,第一边缘节点向第一终端设备发送第二卸载决策。

[0094] S530,第一终端设备根据第二卸载决策确定第一计算任务的第一卸载决策。

[0095] 其中,第一卸载决策用于指示所述第一终端设备在本地计算第一计算任务或者第一终端设备卸载第一计算任务至第一边缘节点进行计算。

[0096] 可选的,第一卸载决策还可以包括第二工作功率。第二工作功率为第一终端设备根据MADRL算法确定的在本地计算第一计算任务的工作功率。

[0097] 需要说明的是,当第一终端设备卸载第一计算任务至第一边缘节点进行计算时,终端设备的工作功率为第一终端设备的休眠功率。

[0098] 可选的,第一终端设备可以根据接收到的第二卸载决策使用MADRL算法确定第一计算任务的第一卸载决策。关于MADRL算法具体的决策过程这里暂不展开叙述,在图6对应的流程图中会做详细的介绍。

[0099] 上述技术方案中,描述了计算卸载过程中终端设备与边缘节点的交互过程与交互信息的具体内容,终端设备和边缘节点通过使得双方能够在实际决策中对于环境有更广泛的感知能力,这能够有效地提升以上两者的决策收益,同时信息交互过程简单、易于实现。

[0100] 参见图6,图6为本申请提供的计算卸载方法的示意性流程图。在该流程中终端设备与边缘节点的决策过程是一个闭环流程,其静止状态在于“等待”步骤,并在每个时间片开始时触发执行流程。

[0101] 本申请中的决策实体包括两种,为一个或多个终端设备与一个或多个边缘节点,以终端设备*i*和边缘节点*j*为例进行说明,该流程实施过程主要包括终端设备*i*和边缘节点*j*基于MADRL算法离线训练确定计算卸载的决策过程与执行过程,其中,边缘节点*j*为终端设备*i*提供计算资源,即 $conn_i = j$ 。

[0102] 应理解,在本申请中,MADRL算法的离线训练过程可以执行在任何位置,仅需要将训练好的边缘节点参数模型部署在对应的边缘节点,并将训练好的一个或多个终端设备的参数模型部署在每一个终端设备即可,从而实现基于MADRL算法的计算卸载方法。

[0103] 终端设备*i*计算卸载过程如下:

[0104] S601,终端设备*i*(即第一终端设备的一例)的决策过程处于等待状态,当每个时间片开始时执行S602。

[0105] 该方法中基于MADRL算法的时间建模方式与其他相关方法中对于时间的建模方式相同,本实施例中将连续的时间划分为长度为 $\delta$ 秒且不相交的时间片( $\delta > 0$ )。实际应用中可以根据待卸载业务的具体情况来确定时间片的长度 $\delta$ ,本申请对此不做具体限定。

[0106] S602,终端设备*i*判断是否有任务待处理,如果没有,跳转至S601,否则执行S603。

[0107] S603,终端设备*i*判断当前是否正在处理计算任务。

[0108] 终端设备*i*处理计算任务的执行过程,是一个在决策过程中开启新线程,在新的线程中以决策结果执行计算任务的流程,应理解,终端设备*i*在同一时间片处理一个计算任务,并且处理单个计算任务的过程可能持续数个时间片(即处理回合)。假设任务产生与执

行中不考虑队列机制对于决策动作的影响,即如果终端设备在处理计算任务的过程中产生了新的计算任务事件,则会忽略新任务并产生对应的任务惩罚值。与此类似,如果终端设备在决策动作时发现任务执行时长超过限制,也将忽略该任务并产生惩罚值。关于任务的惩罚值参见S510中对mp的定义。

[0109] 具体的,终端设备处理单个计算任务的时间片状态忙的回合数(即处理回合)可通过以下方式计算得到: $round = \left\lceil \frac{T}{\delta} \right\rceil$ 。其中,处理时间T是指从终端设备从决策如何执行所在的时间片开始,截止到任务执行完成的时长, $\lceil \cdot \rceil$ 为天花板函数,即表示向上取整,关于本地处理任务和卸载计算任务的时长计算会在后文中的【收益指标】中做详细的介绍,这里暂不展开叙述。

[0110] 终端设备i在每个时间片先判断是否有待处理的任务(S602),如果没有新任务且当前的 $round_i$ 不等于0,则在每个时间片结束的时候执行 $round_i = \max(round_i - 1, 0)$ 。如果有新任务,例如:新任务为计算任务X,则终端设备i继续判断当前的 $round_i$ 是否等于0,当不等于0时,表示系统当前正在处理任务,则忽略待处理的计算任务X并产生对应的任务惩罚值,跳转至S601,如果等于0时,表示已经系统当前没有任务在执行,可以执行计算任务X,跳转至S604。

[0111] S604,终端设备i向边缘节点j(即第一边缘节点的一例)发送任务摘要。

[0112] 对应的,在S703中边缘节点j接收终端设备i发送的任务摘要。

[0113] 该任务摘要包括待处理计算任务X的状态 $M_i$ (即第一计算任务的第一状态的一例),当终端设备i确定发送成功或等待超过一段时间后执行S605。

[0114] 计算任务X的状态 $M_i$ 可表示为 $M_i = \{mt_i, mc_i, md_i, mp_i\}$ 。例如:在S602中,当终端设备i在时间片开始判断自身没有待处理的计算任务时,可表示为 $M_0 = \{0, 0, 0, 0\}$ 。

[0115] 可选的,该任务摘要还包括当前信道噪声 $w_i$ 与当前的信道增益 $H_i$ 。例如:终端设备i可以以网络报文的形式将上述信息发送给边缘节点j,一种具体的消息内容形式为 $\Pi_i = \{M_i, H_i, w_i\}$ 。

[0116] S605,终端设备i接收边缘节点j广播的状态信息。

[0117] 对应的,在S706中边缘节点j通过连接的无线接入点向所服务的终端设备广播状态信息。

[0118] 该状态信息包括边缘节点j为所服务的终端设备的资源分配策略(即第二卸载决策的一例)。关于边缘节点j广播的状态信息的具体内容参见S705中描述,这里暂不介绍。例如:边缘节点j一种具体的状态内容可以为 $\Pi_{d+j} = \{a_{(j,1)}, a_{(j,2)}, \dots, a_{(j,d)}\}$ 。这里边缘节点的决策序号d+j是为了与终端设备的决策序号i相区分。其中, $a_{(j,i)} \in \{0, 1, \dots, a_{(j,avail)}\}$ 表示边缘节点j分配给终端设备i的计算资源数量,0表示不分配计算资源, $a_{(j,avail)}$ 表示分配当前所有可用资源。

[0119] S606,终端设备i根据边缘节点j广播的状态信息与自身的静态状态(包括终端设备i对于环境的部分观测)使用MADRL算法确定行动决策(即第一卸载决策的一例)。

[0120] 当时间片开始时,终端设备i自身的静态状态 $X_i$ 与对应的静态状态空间如下所示,需要说明的是,该静态状态并非多智能体(即多个终端设备)在深度强化学习算法中做出决

策时的状态。

[0121] 终端设备*i*在时间片开始时自身的静态状态可以为 $X_i = (M_i, \text{conn}_i, \text{link}_i, w_i, \text{round}_i)$ 。终端设备对应的静态状态空间是指终端设备和边缘节点在不交互状态信息时,对于问题环境的部分观测所形成的状态空间,即终端设备*i*对应的静态状态空间可表示为

$$x_i = \{M_0, M_{i_1}, M_{i_2}, \dots, M_{i_{\max}}\} \times \{0\} \cup \mathcal{N} \times \{0\} \cup \mathcal{Z} \times \{0, w_{i_1}, w_{i_2}, \dots, w_{i_{\max}}\} \times \left\{0, 1, \dots, \left\lceil \frac{MD_i}{\delta} \right\rceil\right\},$$

其中, $M_0$ 表示无任务, $M_{i_1}, M_{i_2}, \dots, M_{i_{\max}}$ 为终端设备*i*所有可能产生的计算任务类型。 $0, w_{i_1}, w_{i_2}, \dots, w_{i_{\max}}$ 为离散化的环境干扰程度,具体可根据环境干扰的波动范围确定粒度。 $MD_i = \max(\text{md}_{i_1}, \text{md}_{i_2}, \dots, \text{md}_{i_{\max}})$ ,即取终端设备*i*的所有计算任务中执行时长的最大值。

[0122] 终端设备*i*的状态根据接收到的边缘节点*j*的资源分配状况,将静态状态 $X_i$ 变化为 $X_i' = X_i \cup a_{(j,i)}$ (即第一计算任务的第二状态的一例)。

[0123] 网络终端根据新的状态 $X_i'$ 基于MADRL算法离线训练中的收益指标计算执行第一计算任务的代价值,直到达到算法终止条件,根据第一计算任务的最小代价值得到最终的行动决策 $A_i = \{c_i\}$ ,其中: $c_i \in \{0, 1\}$ 表示终端设备*i*针对计算任务是否执行计算卸载决策,0表示终端设备*i*本地执行计算任务*X*,1表示终端设备*i*卸载计算任务*X*至边缘节点*j*进行计算。至此完成基于MADRL的计算卸载决策过程。关于收益指标的描述参见后文,这里暂不展开叙述。

[0124] 可选的,终止条件包括收敛标准。例如:算法执行到一定程度,再执行下去难以得到更小的代价值,这时认为算法收敛达到终止条件。

[0125] 可选的,终止条件包括时间标准。例如:预先给定算法的执行时间,迭代次数等。

[0126] 应理解,本申请对MADRL算法的终止条件不做具体限定。可选的,该行动决策还可以包括计算功率因子(即第二工作功率的一例)。例如:终端设备*i*针对计算任务*X*的行动决策可表示为: $A_i = \{c_i, p_i\}$ , $p_i \in \{0, 1, \dots, p_{\max}\}$ 表示当终端设备*i*本地执行任务时的功率因子,0表示以休眠功率处理任务, $p_{\max}$ 表示以最大功率处理任务。

[0127] 应理解,这里的 $p_i$ 为所述MADRL算法达到终止条件时,第一计算任务的最小代价值对应的工作功率。

[0128] 需要说明的是,这里的 $p_i$ 表示功率的档位,并不代表具体的功率值。例如:0表示终端设备以最低档位的功率值处理任务,这里的最低档位对应的功率值为休眠功率 $P_{\text{sleep}}$ , $P_{\text{sleep}}$ 为一个相对较低的功率值,并不表示休眠功率为0, $p_{\max}$ 表示以最大功率档位处理任务,实际处理任务的功率为终端设备具有的最大工作功率值 $P_{\max}$ 。

[0129] 具体地,终端设备*i*的功率因子对应处理计算任务的频率为:

$$F_i = F_i^{\text{sleep}} + \frac{p_i}{p_{\max}} * (F_i^{\max} - F_i^{\text{sleep}})$$

其中 $F_i^{\text{sleep}}$ 与 $F_i^{\max}$ 是与决策选项无关的常数值,分别表示终端设备在休眠功率与最大功率时对应处理计算任务的频率值。

[0130] 具体地,考虑到终端设备往往提供了系统API(Application Programming Interface,应用程序接口)以调整设备的能量模式,进而决定处理计算任务的速度,因此在本申请中设计了功率因子来表示调整计算频率的功能。举例来说,当终端设备决定以本地执行的方式执行任务,并以 $p_i$ 作为处理计算任务的功率因子之后,终端设备将首先判断 $p_i$ 是否等于0:如果 $p_i$ 不等于0(即不是休眠功率)则意味着终端设备需要从休眠功率切换至 $p_i$ 所

对应的功率以执行计算任务,因此会调用系统API切换计算频率,这里将会产生一定的时间延迟与能量开销,关于本地处理任务时间延迟与能量开销计算会在后文中的【收益指标设定】中做详细的介绍,这里暂不展开叙述。

[0131] 此外,对于终端设备的决策包括以下约束条件:

[0132] ①当终端设备决定计算卸载时则调节功率至休眠功率,即:当 $c_i = 1$ 时,则有 $p_i = 0$ ;

[0133] ②当终端设备无法访问边缘节点时则仅能本地执行任务,即:若 $\text{conn}_i = 0$ 或 $\text{link}_i = 0$ ,则 $c_i = 0$ 。

[0134] 终端设备 $i$ 判断行动决策 $A_i$ 中任务处理时长是否超时,当 $T_i$ 小于 $\text{md}_i$ 时,终端设备 $i$ 确定计算任务 $X$ 的处理回合 $\text{round}_i$ ,然后跳转至S607,当 $T_i$ 大于任务时间限制 $\text{md}_i$ 时,表示执行计算任务 $X$ 会超时,则终端设备 $i$ 直接忽略该任务并产生相应的惩罚值 $\text{mp}_i$ ,然后跳转至S601。

[0135] 需要说明的是,当 $T_i$ 等于 $\text{md}_i$ 时,终端设备 $i$ 根据实际计算任务的情况执行601或607,本申请不做具体限定。

[0136] S607,终端设备 $i$ 开启新线程以决策方式处理计算任务 $X$ ,之后跳转至S601。

[0137] 开启的新线程将获取决策 $A_i$ 和计算任务 $X$ 的相关信息,并以异步方式开始执行S608至S612。

[0138] 需要说明的是,终端设备 $i$ 计算任务 $X$ 的执行过程,是一个由决策过程中开启新线程后,在新的线程中执行的流程步骤,因此决策过程与执行过程不会相互阻塞。

[0139] 需要说明的是,终端设备 $i$ 在同一时间片仅能处理一个计算任务,并且处理单个计算任务的过程可能持续数个时间片(即处理回合)。

[0140] 终端设备 $i$ 处理计算任务 $X$ 的新线程步骤如下所示:

[0141] S608,终端设备 $i$ 判断当前决策 $A_i$ 的决策行动方式是否是计算卸载方式,若是则执行S609,否则跳转至S611。

[0142] S609,终端设备 $i$ 向边缘节点 $j$ 发送处理计算任务 $X$ 所需的信息。

[0143] 需要说明的是,由于该过程中传输的数据流相对较大,因此无线传输速率会影响处理时延,而无线传输速率同时还受到环境噪声与带宽争用的影响。

[0144] S610,终端设备 $i$ 接收边缘节点 $j$ 返回的任务 $X$ 的处理结果,之后跳转至S612。

[0145] S611,终端设备 $i$ 决定以决策 $A_i = \{c_i, p_i\}$ 中的计算功率因子 $p_i$ 处理计算任务 $X$ ,处理完成得到结果后跳转至S612。

[0146] S612,终端设备完成了计算任务 $X$ 的处理过程并返回计算结果,之后结束该线程。

[0147] 相对应地,边缘节点 $j$ 的计算卸载过程如下所示:

[0148] S701,边缘节点 $j$ 的决策过程处于等待状态,边缘节点 $j$ 执行S702或S703。

[0149] 可选的,边缘节点 $j$ 在每个时间片开始时执行S702。

[0150] 可选的,边缘节点 $j$ 先执行S703,S704再执行S702。

[0151] S702,确定边缘节点 $j$ 当前可用的计算资源。S703,边缘节点 $j$ 开始接收所服务的终端设备的任务摘要。

[0152] 由于一个边缘节点往往会服务一个或多个的终端设备,所以一个边缘节点可能会接收到所服务的终端设备发送的一个或多个任务摘要,因此该过程将保持一段时间。其中,边缘节点 $j$ 服务的终端设备可以表示为: $i \in D$ 且 $\text{conn}_i = j$ 。

[0153] S704,边缘节点j停止接收任务摘要,此时确定当前时间片中所服务的终端设备的任务摘要信息,之后跳转至S705。

[0154] 边缘节点j接收到的一个或多个任务摘要中包括终端设备i发送的计算任务X的任务摘要,该任务摘要包括计算任务X的状态信息 $M_i$ 。

[0155] 可选的,该任务摘要还包括当前信道噪声 $w_i$ 与当前的信道增益 $H_i$ 。例如:对应S604,边缘节点j接收到终端设备i发送的网络报文 $\Pi_i = \{M_i, H_i, w_i\}$ 。

[0156] S705,边缘节点j根据接收到的一个或多个任务摘要,采用MADRL的方法决策确定当前可用资源的分配策略。

[0157] 边缘节点j在每个时间片的可用资源的分配策略可表示为: $A_{d+j} = \{a_{(j,1)}, a_{(j,2)}, \dots, a_{(j,d)}\}$ 。这里边缘节点的决策序号d+j是为了与终端设备的决策序号i相区分。其中, $a_{(j,i)} \in \{0, 1, \dots, a_{(j,avail)}\}$ 表示边缘节点j分配给终端设备i的计算资源数量,0表示不分配计算资源, $a_{(j,avail)}$ 表示分配当前所有可用资源。具体地,由分配的计算资源数量对应的任务运算频率为: $f_i = f_{(conn_i,i)} = a_{(conn_i,i)} * f_{unit}$ 。其中 $f_{unit}$ 为决策选项无关的常数数值,单位为赫兹,表示每单位计算资源数量代表的运算频率。

[0158] 其中,边缘节点j的静态状态可表示为: $X_{d+j} = (g_{(j,1)}, g_{(j,2)}, \dots, g_{(j,d)})$ 。其中, $g_{(j,i)}$ 为边缘节点j在本轮决策前已经分配给终端设备i使用的计算资源数量,即边缘节点j正在使用的资源数量。

[0159] 给定特定的时间片t与边缘节点j,决策与静态状态之间存在着以下约束:当前可

分配资源与已占用资源的相互关系: $a_{(j,avail)}^t = a_{(j,max)} - \sum_{i \in D \text{ and } conn_i=j} g_{(j,i)}^t$ 。

[0160] 此外, $g_{(j,i)}$ 在任务执行流程中的变化流程有:首先,当终端设备i以计算卸载的方式 $c_i=1$ 设置处理回合 $round_i$ 时,则赋值 $g_{(j,i)}^{t+1} = a_{(j,i)}^t$  (另有在时间片开始时有 $g_{(j,i)}^1 = 0$ ,即算法开始 $t=1$ 时,终端设备处于空闲状态,此时 $g=0$ )。之后,当终端设备i在时间片t结束时有 $round_i=0$ 时,则再次赋值 $g_{(j,i)}^{t+1} = 0$ 。

[0161] 当边缘节点j接收到所服务的终端设备的报文集合后,边缘节点j的状态变化为 $X'_{d+j} = X_{d+j} \cup \Pi_{1j} \cup \Pi_{2j} \cup \dots \cup \Pi_{nj}$  (即第一边缘节点的第四状态),其中, $X_{d+j}$  (即第一边缘节点的第三状态)为边缘节点j接收多个计算任务摘要之前的状态,其中未接收到报文的消息定义为 $\Pi_m = \{0, 0, 0, 0, 0\}$  (表示终端设备n在当前时间未发送报文至边缘节点j,其中终端设备m为边缘节点j服务的终端设备)。

[0162] 此外,对于边缘节点的决策包括以下约束条件:

[0163] ①边缘节点仅为当前服务的终端设备分配资源,即:不存在 $i \in D$ 和 $j \in N$ ,使得 $conn_i \neq j$ 且 $a_{(j,i)} > 0$ 的情况。

[0164] ②边缘节点本轮所分配的资源不会超过当前可用资源,即:总有

[0165] 即:对于任意 $j \in N$ ,有 $\sum_{i \in N} (a_{(j,i)} \times 1_{conn_i=j}) \leq a_{(j,avail)} \leq a_{(j,max)}$

[0166] ③当边缘节点已为特定计算任务分配资源后,便不会追加分配资源,即:对于所有的 $i \in D$ 和 $j \in N$ ,若 $g_{(j,i)} > 0$ ,则有 $a_{(j,i)} = 0$ 。其中, $g_{(j,i)}$ 为当前时间片开始前边缘节点j已经为

终端设备*i*分配的资源数量。

[0167] 基于上述约束条件,边缘节点*j*根据新的状态 $X'_{d+j}$ 基于MADRL算法进行离线训练学习,直到达到算法终止条件,得到最终的行动决策 $A_{d+j} = \{a_{(j,1)}, a_{(j,2)}, \dots, a_{(j,d)}\}$  (即第二卸载决策的一例),并将该行动决策以广播的方式通过无线接入点发送给所服务的终端设备,例如:一种具体的消息内容形式可以为 $\Pi_{d+j} = \{a_{(j,1)}, a_{(j,2)}, \dots, a_{(j,d)}\}$ 。

[0168] S706,边缘节点*j*通过连接的无线接入点广播其资源分配策略,之后跳转至S707。

[0169] S707,边缘节点*j*开启多个新线程以执行任务处理过程,之后跳转至S701。其中,开启的线程数不应当少于所分配资源的终端设备数量。

[0170] 需要说明的是,边缘节点*j*的计算任务执行过程,是一个由决策过程中开启新线程后,在新的线程中执行的流程步骤,因此决策过程与执行过程不会相互阻塞。此外,该步骤中边缘节点*j*也可通过线程池的方法以建立以下异步过程。开启的新线程将接收来自所服务的特定终端设备的计算任务,并根据决策的资源分配策略,以异步方式开始执行S708至S711。

[0171] 边缘节点*j*处理计算任务的每个新线程的步骤如下所示:

[0172] S708,边缘节点*j*接收所服务的特定终端设备的计算任务信息。应理解,这里的特定终端设备为边缘节点*j*服务的终端设备中确定将计算任务卸载至边缘节点*j*计算的终端设备。

[0173] 例如:S608中终端设备*i*的行动决策确定将计算任务*X*卸载至边缘节点*j*计算,则终端设备*i*向边缘节点*j*发送计算任务*X*的信息,对应的,边缘节点*j*接收终端设备*i*发送的计算任务*X*的信息。

[0174] S709,边缘节点*j*按照S605中的资源分配策略处理接收到的计算任务。

[0175] 可选的,一种具体实现方式可通过资源时分复用的方式将处理器的工作时间切分为小份,并按照资源分配所占总资源量的比例的分配给各个计算线程。

[0176] S710,边缘节点*j*中将新线程处理完成后的计算结果发送给对应的终端设备。

[0177] S711,在完成计算任务流程之后,边缘节点*j*结束对应的线程。

[0178] 至此终端设备*i*和边缘节点*j*完成使用MADRL算法进行计算卸载的过程,下面,主要针对S606和S705中MADRL算法的决策收益指标进行详细介绍。

[0179] 终端设备和边缘节点在使用MADRL算法过程中,需要根据各自的收益指标计算执行任务的的代价值,确定自身的行动策略。

[0180] 终端设备的收益指标分为两种情况:本地执行和计算卸载。

[0181] 第一种情况:

[0182] 在S606中终端设备*i*对于计算任务*X*选择本地执行( $c_i=0$ ),即 $A_i = \{0, p_i\}$ ,其中, $M_i = \{mt_i, mc_i, md_i, mp_i\}$ 。

[0183] ①本地执行任务*X*的时间(即S603中的任务处理时间)可由以下公式计算得出:

$T_i = \frac{1}{F_i} (mc_i + mc_i^{set} * 1_{F_i \neq F_i^{sleep}})$  (即第二时延开销的一例)。其中, $F_i$ 表示根据功率因子确定的终端设备*i*处理计算任务的频率, $1_{F_i \neq F_i^{sleep}}$ 是一个指示符函数,当其右下方的不等式成立时函数值为1,否则函数值为0, $mc_i^{set}$ 代表终端设备*i*从休眠功率调整到 $p_i$ 的所需要的时钟周期数,

对于特定终端设备*i*来说是固定值。

[0184] ②本地执行任务*X*消耗能量可由以下公式计算得出：

$E_i = (\varepsilon * F_i^2) * (mc_i + mc_i^{set} * 1_{F_i \neq F_i^{sleep}})$  (即第二能耗开销的一例)。其中,  $(\varepsilon * F_i^2)$  表示每个运算周期所消耗能量的系数, 可选的,  $\varepsilon = 10^{-11}$ 。

[0185] 第二种情况：

[0186] 在S606中终端设备*i*对于计算任务*X*选择计算卸载 ( $c_i = 1$ ), 即  $A_i = \{1, 0\}$ 。

[0187] ①计算卸载执行任务*X*的时间可由以下公式计算得出： $T_i = \frac{mt_i}{R_i(c_{all})} + \frac{mc_i}{f_i}$  (即第一

时延开销的一例), 即为传输该特定计算任务数据流的时间  $\frac{mt_i}{R_i(c_{all})}$  与边缘节点的处理时间

$\frac{mc_i}{f_i}$  之和。其中, 传输该特定计算任务数据流的无线传输速率  $R_i(c_{all})$  可由以下公式计算得

出： $R_i(c_{all}) = W_{link_i} \log_2(1 + P_i H_i / (w_i + \sum_{k \in N \setminus \{i\}} (P_k H_k * 1_{c_k=1} * 1_{link_k=link_i})))$ 。其中,  $c_{all} = \{c_1, c_2, \dots,$

$c_d\}$ , 表示所有终端设备是否执行计算卸载的决策集合。 $W_{link_i}$  表示序号为  $link_i$  的无线信道带宽 (固定常数值),  $P_i$  表示终端设备*i*的传输功率 (本申请中假设其固定不变),  $H_i$  表示终端设备*i*至无线接入点的信道增益。 $w_i$  表示背景干扰功率, 具体包括背景噪声与不参与计算卸载

决策的其它无线设备在该信道上的通信干扰。 $\sum_{k \in N \setminus \{i\}} (P_k H_k * 1_{c_k=1} * 1_{link_k=link_i})$  表示在当前时间片

做出计算卸载决策且使用相同信道的其它设备的干扰和。

[0188] ②计算卸载执行任务*X*的消耗能量可表示为  $E_i = P_i * \frac{mt_i}{R_i(c_{all})} + E^{tail}$  (即第一能耗开

销的一例), 即数据发送能耗与接收能耗之和, 其中, 接收能耗包含一部分接收后等待过程所产生的尾部能耗,  $E^{tail}$  为接收任务耗能, 考虑到接收任务耗时较短, 一般情况下可以将其近似为常数值。

[0189] 综合上述两种效益指标, 终端设备执行计算任务的代价 (即收益的负值) 权衡了任务执行时间与对应的能量消耗, 具体表示为：

$$[0190] \quad C_i(A, X) = \begin{cases} \alpha_i * T_i + \beta_i * E_i & c_i = 1 \text{ and } M_i \neq M_0 \text{ and } T_i \leq md_i \text{ and } round_i = 0 \\ T_i + \beta_i * E_i & c_i = 0 \text{ and } M_i \neq M_0 \text{ and } T_i \leq md_i \text{ and } round_i = 0 \\ mp_i & otherwise \end{cases}$$

[0191] 上述终端设备*i*的代价公式  $C_i(A, X)$  (即第一代价函数的一例) 表示了终端设备*i*根据当前任务的状态与所有决策实体 (即所有的终端设备和边缘节点) 的决策集合*A*, 处理决策流程所产生的资源耗费对应的代价值。其中,  $\alpha_i \in (0, 1]$  (即第一参数的一例) 为技术差异因子, 用来权衡终端设备*i*与对应的边缘节点处理任务的技术差异。当终端设备与对应的边缘节点采用相同的技术方法处理计算任务时有  $\alpha_i = 1$ , 否则若对应的边缘节点采用了性能表现更好的技术处理计算任务, 则需要通过调整该值权衡其处理代价。定义  $\beta_i \geq 0$  (即第二参数的一例) 为能量权衡因子, 用来表示终端设备*i*对于能量消耗的敏感程度取值, 当取基线值时有  $\beta_i = 1$ 。



[0192] 应理解,这里的决策 $A_i$ 并非是终端设备 $i$ 的最终行动决策,该行动决策为终端设备 $i$ 使用MADRL算法迭代学习过程中的某一次决策。终端设备 $i$ 的最终决策为在MADRL算法达到终止条件时,根据该收益指标计算得到计算任务 $X$ 的最小代价值确定的行动决策。

[0193] 边缘节点的收益指标:

[0194] 边缘节点的收益指标评价权衡了其服务的终端设备的平均代价与公平性。综合上述指标,在S705中,边缘节点 $j$ 的收益指标具体表示为: $C_{d+j} = ct_j / cf_j$  (即第二代价函数的一例),边缘节点 $j$ 的收益指标用于计算边缘节点 $j$ 在不同资源分配策略下的代价值。其中, $ct_j$  (即平均代价函数的一例)表示了当前时间片内边缘节点 $j$ 服务的终端设备的平均代价,即有:

$$[0195] \quad ct_j = \begin{cases} \frac{\sum_{i \in D_{(j, serv)}} C_i(A, X)}{|D_{(j, serv)}|} & \text{if } |D_{(j, serv)}| > 0 \\ 0 & \text{otherwise} \end{cases}$$

[0196] 其中, $D_{(j, serv)} = \{i \mid i \in D \text{ and } conn_i = j\}$

[0197] 其中, $cf_j$  (即公平代价函数的一例)是根据当前正使用边缘节点 $j$ 的计算资源的终端设备的数量设置的公平因子,具体可表示为:

$$[0198] \quad cf_j = K \left( \sum_{i \in D: conn_i = j} 1_{g'_{(j,i)} > 0} \right)$$

[0199] 其中, $g'_{(j,i)}$ 为当前时间片结束时,边缘节点 $j$ 已经为终端设备 $i$ 分配的资源数量。 $K(x)$ 为非负单调递增函数,具体可根据实际情况设计经验函数,本申请不做限定。

[0200] 可选的,一种 $K(x)$ 函数可以设置为 $K(0) = 0.2, K(1) = 1, K(2) = 1.6, K(3) = 1.8, K(4) = 2.2, K(5) = 2.5, K(6) = 2.7, K(7) = 2.8, K(x) = 3 (x \geq 8 \text{ 且 } x \text{ 为整数})$ 。

[0201] 可选的,另一种 $K(x)$ 函数可以设置为 $K(x) = \log_2(x+2)$ ,其中, $x \geq 0$ 且 $x$ 为整数。

[0202] 可选的,又一种 $K(x)$ 函数可以设置为 $K(x) = x+1$ ,其中, $x \geq 0$ 且 $x$ 为整数。

[0203] 应理解,边缘节点 $j$ 的最终决策为在MADRL算法达到终止条件时,根据边缘节点的收益指标计算得到边缘节点 $j$ 的最小代价值确定的行动决策。

[0204] 上述技术方案中,详细描述了计算卸载过程中终端设备与边缘节点的交互过程与交互信息,使得双方能够在实际决策中对于环境有更广泛的感知能力,这能够有效地提升以上两者的决策收益,同时信息交互过程简单、易于实现,并且过程对应的开销也相对较小。

[0205] 同时考虑了计算卸载方(即终端设备)与资源提供方(即边缘节点)的卸载决策与资源分配决策,提出的收益指标权衡了多种因素,同时根据现实情况定义了相应的公平因子,收益指标反映的现实环境更为全面。同时受益于MADRL算法的优势,本申请中的必要近似较少并能同时保障算法在求解问题时的稳定性。这里的必要近似是指,在求解问题时不得不引入的近似假设或者条件,而MADRL在求解时对于问题适应性较强。

[0206] 另外,在终端设备本地处理计算任务的收益指标中,引入功率控制因素,特别分析并解决了功率切换时的代价损失这一问题,这有助于更为精细地描述计算卸载中的决策内容,进而有效地提高用户体验、减少模型误差。另一方面,本申请中针对边缘节点决策项也保证所服务的终端设备的平均体验的同时,也提升了网络终端在资源利用上的公平性,即

在保证资源高效分配的同时也避免了服务用户的数量过少。

[0207] 下面以多智能体深度确定性策略梯度 (multi-agent deep deterministic policy gradient, MADDPG) 算法为例说明计算卸载的过程, 作为一种典型的MADRL算法, MADDPG是DDPG为适应多智能体环境的改进算法, 其中, 每一个智能体都独立运行着一个DDPG。DDPG算法是Actor-Critic算法的升级版, MADDPG的改进之处是每个智能体的Critic部分都能够获取到其余智能体的动作信息, 以此在训练过程中感知到其它智能体的行动策略。

[0208] 基于MADDPG的计算卸载方法包括离线训练和在线推导过程, 其中, 离线训练包括数据采集与训练过程。结合图7-图9对MADDPG的离线训练过程与在线推导过程进行介绍。

[0209] 参见图7, 图7是本申请提供的基于MADDPG的数据采集的示意性流程图。其中, 训练数据采集过程的具体步骤如下:

[0210] (1) 收集边缘节点信息: 收集边缘节点的运算能力以及服务的终端设备列表。

[0211] (2) 收集终端设备静态信息: 从终端设备上收集设备相关的静态信息, 包括: 终端设备的运算能力、传输功率、关联情况、任务种类等。

[0212] (3) 判断是否动态生成数据集, 如果是, 则执行(4), 否则跳转至(6)

[0213] (4) 建立距离-增益拟合模型: 根据传输速率与物理距离等动态信息建立距离-增益的拟合模型, 以获得当前状态中终端设备的信道增益 $H$ , 并随机生成背景噪声 $w$ 。

[0214] (5) 模拟轨迹与任务产生事件: 根据给定参数模拟终端移动轨迹与任务产生机制。假设终端移动以随机路点模式生成移动轨迹, 假设计算任务以随机事件的方式发生并遵从伯努利分布。随后, 跳转至(7)。

[0215] (6) 收集终端动态信息: 从终端上收集信道干扰、传输速率与任务产生事件等决策所需的动态信息。收集完毕后执行下一步。

[0216] (7) 统计信息并制作成数据集。随后结束。

[0217] 参见图8, 图8是本申请提供的基于MADDPG的参数模型训练的示意性流程图。

[0218] 这里仅简单介绍终端设备和边缘节点的参数模型的训练过程, 关于如何使用训练好的参数模型的进行离线训练的过程参见图6中的描述, 这里不再赘述。

[0219] 离线训练参数模型的过程如下所述:

[0220] (1) 导入离线数据集, 将收集过程中制作的数据集导入训练模型。

[0221] (2) 初始化每个终端设备与边缘节点的策略参数、随机函数与初始系统状态 $X$ 。

[0222] 对于终端设备, 系统初始状态包括: 初始的任务状态、网络链路情况、边缘节点连接情况, 网络噪声状态, 此时处理回合为0 (即此时终端设备为空闲状态)。

[0223] 对于边缘节点, 系统初始状态为  $(0, 0, 0, \dots, 0)$ , 表示初始时尚未分配计算资源给所服务的终端设备。

[0224] (3) 执行计算卸载流程, 边缘节点首先根据策略参数、观测状态与随机函数确定决策。

[0225] (4) 终端设备根据策略参数、观测状态与随机函数确定相对应的决策。

[0226] (5) 根据系统状态 $X$ 与决策集合 $A$ , 训练过程得到终端设备与边缘节点对于该轮时间片内任务执行的行动代价 $C$ , 并产生下一个系统状态 $X'$ 。

[0227] (6) 收集上述的系统状态 $X$ 、智能体动作集合 $A$ 、行动代价 $C$ 与下一个系统状态 $X'$ , 形

成经验 $(X, A, C, X')$ ，保持到经验库D中。

[0228] (7) 在训练过程所在的设备(终端设备或边缘节点)中，逐次执行每一个终端设备与边缘节点的训练过程，并更新相应模型的策略梯度。

[0229] (8) 最后，判定训练过程是否达到了终止条件，如果达到了就结束该过程并保存模型参数，否则即继续训练过程。

[0230] 参见图9，图9是本申请提供的基于MADDPG的计算卸载的方法的示意性流程图。图9与图6中描述的计算卸载过程基本一致，这里只做简要说明。

[0231] (1) 初始化：边缘节点根据获得的参数信息进行初始化过程。终端设备上传计算任务种类与运算能力等终端设备静态信息。边缘节点根据以上信息下放类似配置的终端设备模型参数，以供终端进行模型参数的初始化，并执行时间片的同步过程。

[0232] (2) 状态感知：当终端有新任务产生时，就会在时间片开始时向对应的边缘节点发送任务摘要。边缘节点根据当前状态与终端设备的摘要信息统计并执行MADRL的推导过程，决策计算资源分配策略并广播给所服务的终端设备。

[0233] (3) 决策行动：终端设备根据边缘节点的广播信息与自身状态信息执行MADRL的推导过程，在MADDPG算法中仅通过Actor参数来决策相应的行动。

[0234] (4) 处理任务：终端设备与边缘节点完成决策过程，执行相应的行动。

[0235] (5) 结束：当终端设备不再需要计算卸载时，则退出决策过程，当边缘节点所有服务的所有终端设备都不再需要计算卸载后，则边缘节点退出服务过程。

[0236] 上文描述了本申请提供的方法实施例，下文将描述本申请提供的装置实施例。应理解，装置实施例的描述与方法实施例的描述相互对应，因此，未详细描述的内容可以参见上文方法实施例，为了简洁，这里不再赘述。

[0237] 参见图10，图10为本申请提供的通信装置1000的示意性框图。如图10，通信装置1000包括接收单元1100、发送单元1200和处理单元1300。

[0238] 发送单元1200，用于向第一边缘节点发送第一计算任务的第一状态，其中，第一边缘节点为所述装置获取计算资源的边缘节点，第一状态包括传输第一计算任务的数据流的长度、计算第一计算任务所需耗费的时钟数、第一计算任务的惩罚值中的至少一个；

[0239] 接收单元1100，用于接收第一边缘节点发送的第二卸载决策，第二卸载决策是由第一状态确定的，第二卸载决策包括一个或多个第二终端设备的计算资源分配信息，第二终端设备为从第一边缘节点获取计算资源的终端设备且所述装置为一个或多个第二终端设备中的一个终端设备；

[0240] 处理单元1300，用于根据第二卸载决策确定第一计算任务的第一卸载决策，第一卸载决策用于指示是否卸载第一计算任务至第一边缘节点进行计算。

[0241] 可选地，接收单元1100和发送单元1200也可以集成为一个收发单元，同时具备接收和发送的功能，这里不作限定。

[0242] 可选地，在一个实施例中，当第一卸载决策指示该通信装置卸载第一计算任务至第一边缘节点进行计算时，发送单元1200还用于：

[0243] 向第一边缘节点发送第一计算任务；接收单元1100，还用于接收第一边缘节点发送的第一计算任务的计算结果；或者当第一卸载决策指示通信装置不卸载所述第一计算任务时，处理单元1300，还用于在本地确定第一计算任务的计算结果。

[0244] 可选地,在一个实施例中,处理单元1300,具体用于根据第二卸载决策更新第一计算任务的第一状态中的参数得到第一计算任务的第二状态;根据第二状态计算第一计算任务的代价值,第一计算任务的代价值包括第一计算任务的本地开销和卸载开销;根据第一计算任务的代价值确定第一计算任务的第一卸载决策。

[0245] 可选地,在一个实施例中,处理单元1300,具体用于根据第二状态,使用多智能体深度强化学习MADRL算法中的第一代价值函数确定第一计算任务的代价值,第一代价值函数包括卸载开销函数和本地计算开销函数,其中,卸载开销函数用于确定第一计算任务的卸载开销,本地计算开销函数用于确定第一计算任务的本地开销;以及根据MADRL算法迭代更新所述装置的第一计算任务的状态和第一计算任务的代价值;当MADRL算法达到终止条件时,处理单元1300,还用于根据第一计算任务的代价值确定第一计算任务的第一卸载决策。

[0246] 可选地,在一个实施例中,第一计算任务的卸载开销包括第一能耗开销和第一时延开销,其中,第一能耗开销包括所述装置将第一计算任务卸载至第一边缘消耗的能量,第一时延开销包括所述装置将第一计算任务卸载至第一边缘节点的时延以及第一边缘节点确定第一计算任务的计算结果的时延。

[0247] 可选地,在一个实施例中,第一计算任务的本地开销包括第二能耗开销和第二时延开销,其中,第二能耗开销包括所述装置本地计算第一计算任务消耗的能量和装置从休眠功率 $P_{sleep}$ 切换到第一工作功率 $P_1$ 消耗的能量,第一时延开销包括装置本地计算第一计算任务的时延和所述装置从休眠功率切换到第一工作功率 $P_1$ 的时延,第一工作功率 $P_1$ 为所述装置本地计算任务的工作功率。

[0248] 可选地,在一个实施例中,第一卸载决策还包括第二工作功率,第二工作功率为MADRL算法达到终止条件时,第一计算任务的最小代价值对应的工作功率。

[0249] 可选地,在一个实施例中,当第一卸载决策指示所述装置卸载第一计算任务至第一边缘节点进行计算时,处理单元1300,还用于以休眠功率工作。

[0250] 可选地,在一个实施例中,处理单元1300,还用于使用第一参数对第一时延开销进行动态调节,第一参数用于表示处理单元1300与第一边缘节点处理计算任务的技术差异。

[0251] 可选地,在一个实施例中,处理单元1300,还用于使用第二参数对第一能耗开销和第二能耗开销进行动态调节,第二参数用于表示处理单元1300对于能耗开销的敏感程度。

[0252] 在一种实现方式中,通信装置1000可以为方法实施例中的第一终端设备。在这种实现方式中,接收单元1100可以为接收器,发送单元1200可以为发射器。接收器和发射器也可以集成为一个收发器。

[0253] 在另一种实现方式中,通信装置1000可以为第一终端设备中的芯片或集成电路。在这种实现方式中,接收单元1100和发送单元1200可以为通信接口或者接口电路。例如,接收单元1100为输入接口或输入电路,发送单元1200为输出接口或输出电路。

[0254] 处理单元1300可以为处理装置。其中,处理装置的功能可以通过硬件实现,也可以通过硬件执行相应的软件实现。例如,处理装置可以包括至少一个处理器和至少一个存储器,其中,所述至少一个存储器用于存储计算机程序,所述至少一个处理器读取并执行所述至少一个存储器中存储的计算机程序,使得通信装置1000执行各方法实施例中由第一终端设备执行的操作和/或处理。

[0255] 可选地,处理装置可以仅包括处理器,用于存储计算机程序的存储器位于处理装

置之外。处理器通过电路/电线与存储器连接,以读取并执行存储器中存储的计算机程序。可选地,在一些示例中,处理装置还可以为芯片或集成电路。

[0256] 参见图11,图11为本申请提供的通信装置2000的示意性框图。如图11,通信装置2000包括接收单元2100、发送单元2200和处理单元2300。

[0257] 接收单元2100,用于接收一个或多个任务的状态,一个或多个任务的状态包括第一终端设备发送的第一计算任务的第一状态,所述装置是为一个或多个第二终端设备提供计算资源的边缘节点,且第一终端设备为一个或多个第二终端设备中的一个终端设备;

[0258] 处理单元2300,用于根据一个或多个任务状态,确定第二卸载决策,第二卸载决策包括处理单元2300对一个或多个第二终端设备的计算资源分配信息;

[0259] 发送单元2200,用于向一个或多个第二终端设备广播第二卸载决策。

[0260] 可选地,接收单元2100和发送单元2200也可以集成为一个收发单元,同时具备接收和发送的功能,这里不作限定。

[0261] 可选地,在一个实施例中,接收单元2100,还用于接收第一终端设备发送的第一计算任务;处理单元2300,还用于确定第一计算任务的计算结果;发送单元2200,还用于向第一终端设备发送第一计算任务的计算结果。

[0262] 可选地,在一个实施例中,处理单元2300,具体用于根据一个或多个任务的状态更新第一边缘节点的第三状态得到第一边缘节点的第四状态,其中,第三状态是第一边缘节点接收到一个或多个任务状态之前的状态;处理单元2300,还用于根据第四状态确定装置的代价值,代价值为处理单元2300为一个或多个计算任务分配计算资源的开销;处理单元2300,根据代价值确定第二卸载决策。

[0263] 可选地,在一个实施例中,处理单元2300,具体用于根据第四状态,使用多智能体深度强化学习MADRL算法中的第一代价值函数和第二代价值函数确定代价值;第一代价值函数包括卸载开销函数和本地计算开销函数,其中,卸载开销函数用于确定一个或多个任务的卸载开销,本地计算开销函数用于计算一个或多个任务的本地开销;第二代价值函数包括平均代价值函数和公平代价值函数,其中,平均代价值函数用于根据一个或多个任务的卸载开销和本地开销确定一个或多个任务的平均开销,公平代价值函数用于根据使用所述装置的计算资源的第二终端设备的数量确定所述装置的公平代价;处理单元2300,具体用于根据一个或多个任务的平均开销与公平代价确定所述装置的代价值。

[0264] 可选地,在一个实施例中,处理单元2300,具体用于根据MADRL算法迭代更新第一边缘节点的状态和第一边缘节点的代价值;在MADRL算法达到终止条件的情况下,处理单元2300,还用于根据第一边缘节点的代价值确定第二卸载决策。

[0265] 在一种实现方式中,通信装置2000可以为方法实施例中的第一边缘节点。在这种实现方式中,接收单元2100可以为接收器,发送单元2200可以为发射器。接收器和发射器也可以集成为一个收发器。

[0266] 在另一种实现方式中,通信装置2000可以为第一边缘节点中的芯片或集成电路。在这种实现方式中,接收单元2100和发送单元2200可以为通信接口或者接口电路。例如,接收单元2100为输入接口或输入电路,发送单元2200为输出接口或输出电路。

[0267] 处理单元1300可以为处理装置。其中,处理装置的功能可以通过硬件实现,也可以通过硬件执行相应的软件实现。例如,处理装置可以包括至少一个处理器和至少一个存储

器,其中,所述至少一个存储器用于存储计算机程序,所述至少一个处理器读取并执行所述至少一个存储器中存储的计算机程序,使得通信装置2000执行各方法实施例中由第一边缘节点执行的操作和/或处理。

[0268] 可选地,处理装置可以仅包括处理器,用于存储计算机程序的存储器位于处理装置之外。处理器通过电路/电线与存储器连接,以读取并执行存储器中存储的计算机程序。可选地,在一些示例中,处理装置还可以为芯片或集成电路。

[0269] 参见图12,图12为本申请提供的通信装置10的示意性结构图。如图12,通信装置10包括:一个或多个处理器11,一个或多个存储器12以及一个或多个通信接口13。处理器11用于控制通信接口13收发信号,存储器12用于存储计算机程序,处理器11用于从存储器12中调用并运行该计算机程序,以使得本申请各方法实施例中由终端设备执行的流程和/或操作被执行。

[0270] 例如,处理器11可以具有图10中所示的处理单元1300的功能,通信接口13可以具有图10中所示的接收单元1100和/或发送单元1200的功能。具体地,处理器11可以用于执行图5和图6中由第一终端设备内部执行的处理或操作,通信接口13用于执行图5和图6中由第一终端设备执行的发送和/或接收的动作。

[0271] 在一种实现方式中,通信装置10可以为方法实施例中的第一终端设备。在这种实现方式中,通信接口13可以为收发器。收发器可以包括接收器和发射器。可选地,处理器11可以为基带装置,通信接口13可以为射频装置。在另一种实现中,通信装置10可以为安装在第一终端设备中的芯片或者集成电路。在这种实现方式中,通信接口13可以为接口电路或者输入/输出接口。

[0272] 参见图13,图13为本申请提供的通信装置20的示意性结构图。如图13,通信装置20包括:一个或多个处理器21,一个或多个存储器22以及一个或多个通信接口23。处理器21用于控制通信接口23收发信号,存储器22用于存储计算机程序,处理器21用于从存储器22中调用并运行该计算机程序,以使得本申请各方法实施例中由网络设备执行的流程和/或操作被执行。

[0273] 例如,处理器21可以具有图11中所示的处理单元2300的功能,通信接口23可以具有图11中所示的发送单元2200和/或接收单元2100的功能。具体地,处理器21可以用于执行图5和图6中由第一边缘节点内部执行的处理或操作,通信接口33用于执行图5和图6中由第一边缘节点执行的发送和/或接收的动作。

[0274] 在一种实现方式中,通信装置20可以为方法实施例中的第一边缘节点。在这种实现方式中,通信接口23可以为收发器。收发器可以包括接收器和发射器。在另一种实现中,通信装置20可以为安装在第一边缘节点中的芯片或者集成电路。在这种实现方式中,通信接口23可以为接口电路或者输入/输出接口。

[0275] 可选的,上述各装置实施例中的存储器与处理器可以是物理上相互独立的单元,或者,存储器也可以和处理器集成在一起,本文不做限定。

[0276] 此外,本申请还提供一种计算机可读存储介质,所述计算机可读存储介质中存储有计算机指令,当计算机指令在计算机上运行时,使得本申请各方法实施例中由第一终端设备执行的操作和/或流程被执行。

[0277] 本申请还提供一种计算机可读存储介质,所述计算机可读存储介质中存储有计算

机指令,当计算机指令在计算机上运行时,使得本申请各方法实施例中由第一边缘节点执行的操作和/或流程被执行。

[0278] 此外,本申请还提供一种计算机程序产品,计算机程序产品包括计算机程序代码或指令,当计算机程序代码或指令在计算机上运行时,使得本申请各方法实施例中由第一终端设备执行的操作和/或流程被执行。

[0279] 本申请还提供一种计算机程序产品,计算机程序产品包括计算机程序代码或指令,当计算机程序代码或指令在计算机上运行时,使得本申请各方法实施例中由第一边缘节点执行的操作和/或流程被执行。

[0280] 此外,本申请还提供一种芯片,所述芯片包括处理器。用于存储计算机程序的存储器独立于芯片而设置,处理器用于执行存储器中存储的计算机程序,以使得任意一个方法实施例中由第一终端设备执行的操作和/或处理被执行。

[0281] 进一步地,所述芯片还可以包括通信接口。所述通信接口可以是输入/输出接口,也可以为接口电路等。进一步地,所述芯片还可以包括所述存储器。

[0282] 本申请还提供一种芯片,所述芯片包括处理器。用于存储计算机程序的存储器独立于芯片而设置,处理器用于执行存储器中存储的计算机程序,以使得任意一个方法实施例中由第一边缘节点执行的操作和/或处理被执行。

[0283] 进一步地,所述芯片还可以包括通信接口。所述通信接口可以是输入/输出接口,也可以为接口电路等。进一步地,所述芯片还可以包括所述存储器。

[0284] 此外,本申请还提供一种通信装置(例如,可以为芯片),包括处理器和通信接口,所述通信接口用于接收信号并将所述信号传输至所述处理器,所述处理器处理所述信号,以使得任意一个方法实施例中由第一终端设备执行的操作和/或处理被执行。

[0285] 本申请还提供一种通信装置(例如,可以为芯片),包括处理器和通信接口,所述通信接口用于接收信号并将所述信号传输至所述处理器,所述处理器处理所述信号,以使得任意一个方法实施例中由第一边缘节点执行的操作和/或处理被执行。

[0286] 此外,本申请还提供一种通信装置,包括至少一个处理器,所述至少一个处理器与至少一个存储器耦合,所述至少一个处理器用于执行所述至少一个存储器中存储的计算机程序或指令,使得任意一个方法实施例中由第一终端设备执行的操作和/或处理被执行。

[0287] 本申请还提供一种通信装置,包括至少一个处理器,所述至少一个处理器与至少一个存储器耦合,所述至少一个处理器用于执行所述至少一个存储器中存储的计算机程序或指令,使得任意一个方法实施例中由第一边缘节点执行的操作和/或处理被执行。

[0288] 此外,本申请还提供一种第一终端设备,包括处理器、存储器和收发器。其中,存储器用于存储计算机程序,处理器用于调用并运行存储器中存储的计算机程序,并控制收发器收发信号,以使第一终端设备执行任意一个方法实施例中由第一终端设备执行的操作和/或处理。

[0289] 本申请还提供一种第一边缘节点,包括处理器、存储器和收发器。其中,存储器用于存储计算机程序,处理器用于调用并运行存储器中存储的计算机程序,并控制收发器收发信号,以使第一终端设备执行任意一个方法实施例中由第一边缘节点执行的操作和/或处理。

[0290] 此外,本申请还提供一种无线通信系统,包括本申请实施例中的第一终端设备和

第一边缘节点。

[0291] 本申请实施例中的处理器可以是集成电路芯片,具有处理信号的能力。在实现过程中,上述方法实施例的各步骤可以通过处理器中的硬件的集成逻辑电路或者软件形式的指令完成。处理器可以是通用处理器、数字信号处理器(digital signal processor,DSP)、专用集成电路(application specific integrated circuit,ASIC)、现场可编程门阵列(field programmable gate array,FPGA)或其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。本申请实施例公开的方法的步骤可以直接体现为硬件编码处理器执行完成,或者用编码处理器中的硬件及软件模块组合执行完成。软件模块可以位于随机存储器,闪存、只读存储器,可编程只读存储器或者电可擦写可编程存储器、寄存器等本领域成熟的存储介质中。该存储介质位于存储器,处理器读取存储器中的信息,结合其硬件完成上述方法的步骤。

[0292] 本申请实施例中的存储器可以是易失性存储器或非易失性存储器,或可包括易失性和非易失性存储器两者。其中,非易失性存储器可以是只读存储器(read-only memory,ROM)、可编程只读存储器(programmable ROM,PROM)、可擦除可编程只读存储器(erasable PROM,EPROM)、电可擦除可编程只读存储器(electrically EPROM,EEPROM)或闪存。易失性存储器可以是随机存取存储器(random access memory,RAM),其用作外部高速缓存。通过示例性但不是限制性说明,许多形式的RAM可用,例如静态随机存取存储器(static RAM,SRAM)、动态随机存取存储器(dynamic RAM,DRAM)、同步动态随机存取存储器(synchronous DRAM,SDRAM)、双倍数据速率同步动态随机存取存储器(double data rate SDRAM,DDR SDRAM)、增强型同步动态随机存取存储器(enhanced SDRAM,ESDRAM)、同步连接动态随机存取存储器(synchlink DRAM,SLDRAM)和直接内存总线随机存取存储器(direct rambus RAM,DRRAM)。应注意,本文描述的系统和方法的存储器旨在包括但不限于这些和任意其它适合类型的存储器。

[0293] 本领域普通技术人员可以意识到,结合本文中所公开的实施例描述的各示例的单元及算法步骤,能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行,取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应认为超出本申请的范围。

[0294] 所属领域的技术人员可以清楚地了解到,为描述的方便和简洁,上述描述的系统、装置和单元的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0295] 在本申请所提供的几个实施例中,应该理解到,所揭露的系统、装置和方法,可以通过其它的方式实现。例如,以上所描述的装置实施例仅仅是示意性的,例如,所述单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,装置或单元的间接耦合或通信连接,可以是电性,机械或其它的形式。

[0296] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个



网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0297] 另外,在本申请各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。

[0298] 本申请中术语“和/或”,仅仅是一种描述关联对象的关联关系,表示可以存在三种关系,例如,A和/或B,可以表示:单独存在A,同时存在A和B,单独存在B这三种情况。其中,A、B以及C均可以为单数或者复数,不作限定。

[0299] 所述功能如果以软件功能单元的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读取存储介质中。基于这样的理解,本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备)执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、ROM、RAM、磁碟或者光盘等各种可以存储程序代码的介质。

[0300] 以上所述,仅为本申请的具体实施方式,但本申请的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本申请揭露的技术范围内,可轻易想到变化或替换,都应涵盖在本申请的保护范围之内。因此,本申请的保护范围应以所述权利要求的保护范围为准。

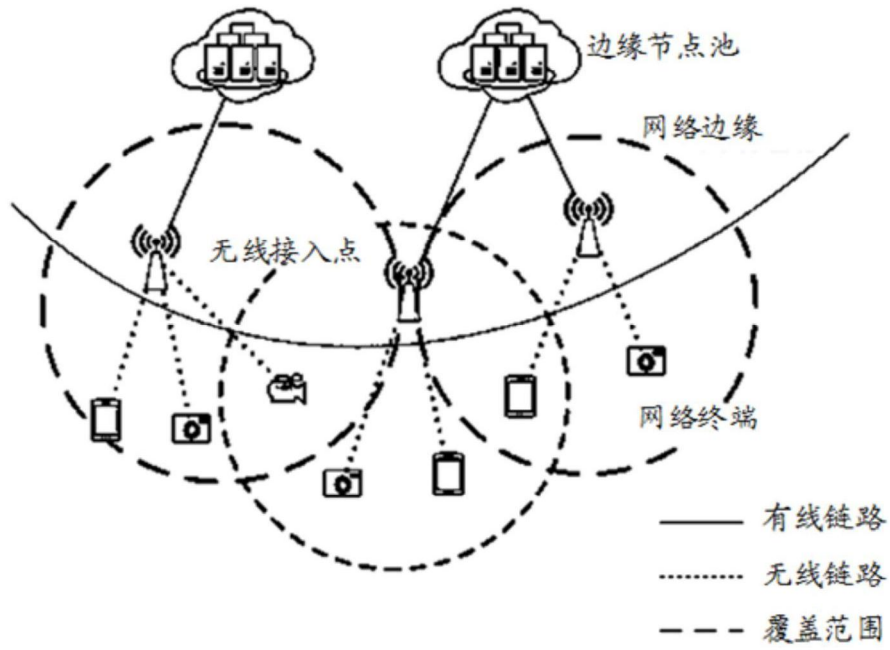


图1

状态  $s(t+1)$

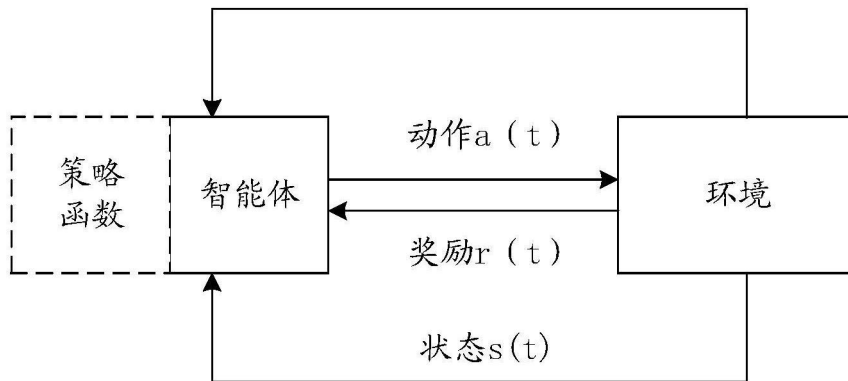


图2

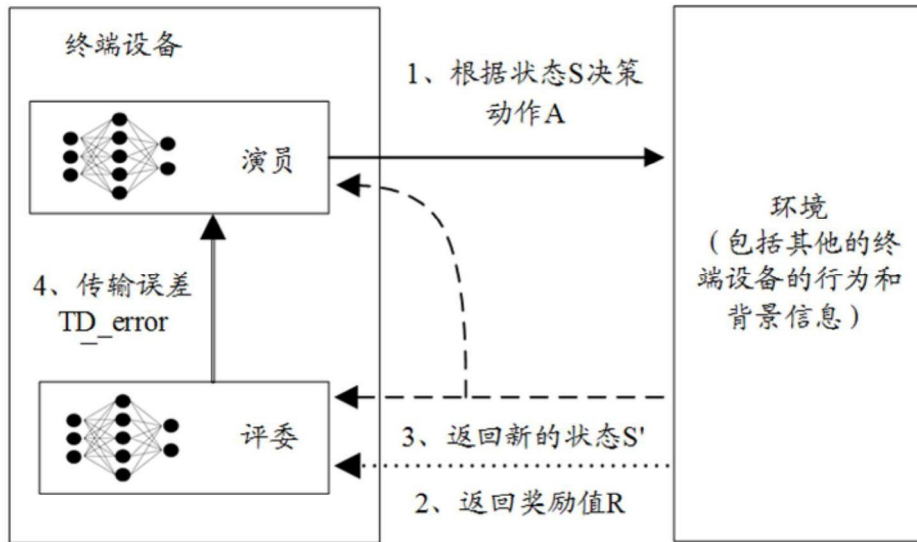


图3

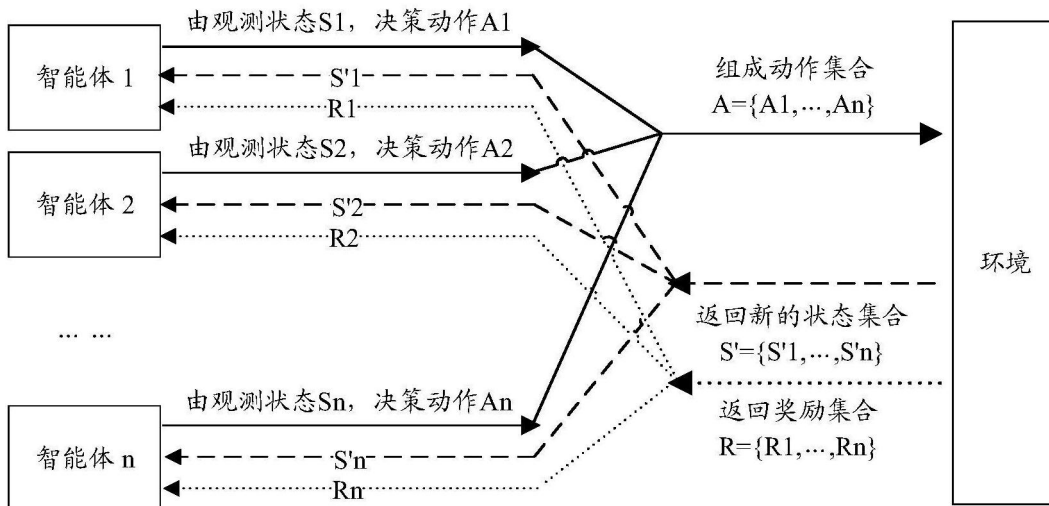


图4

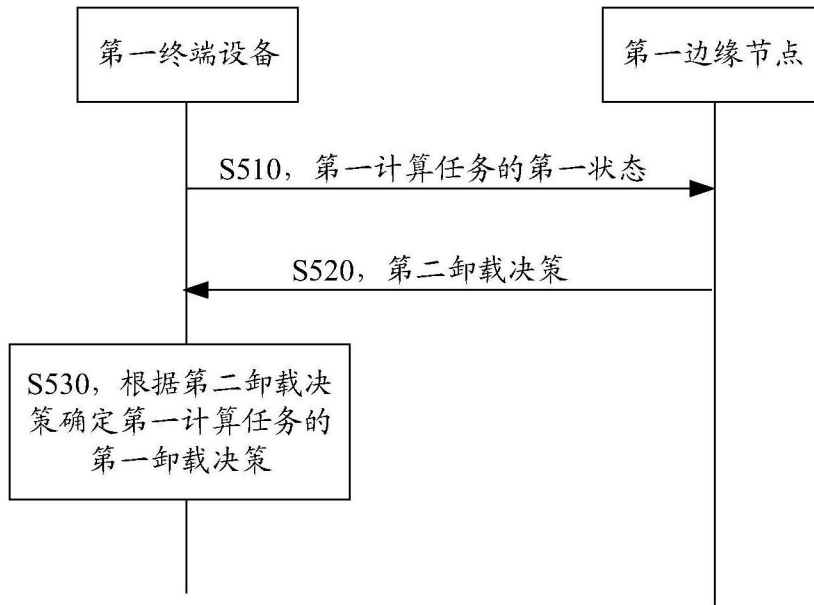


图5

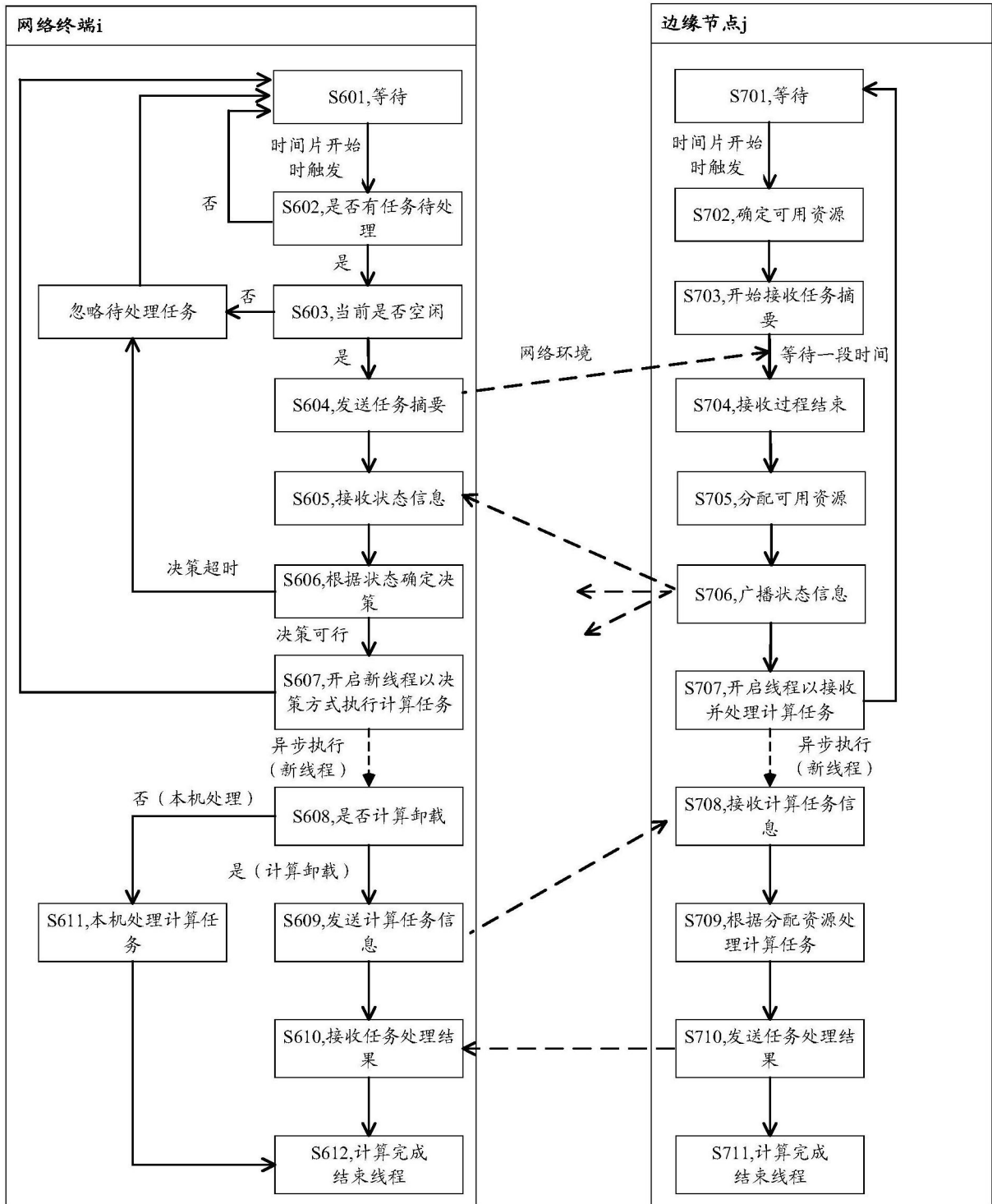


图6

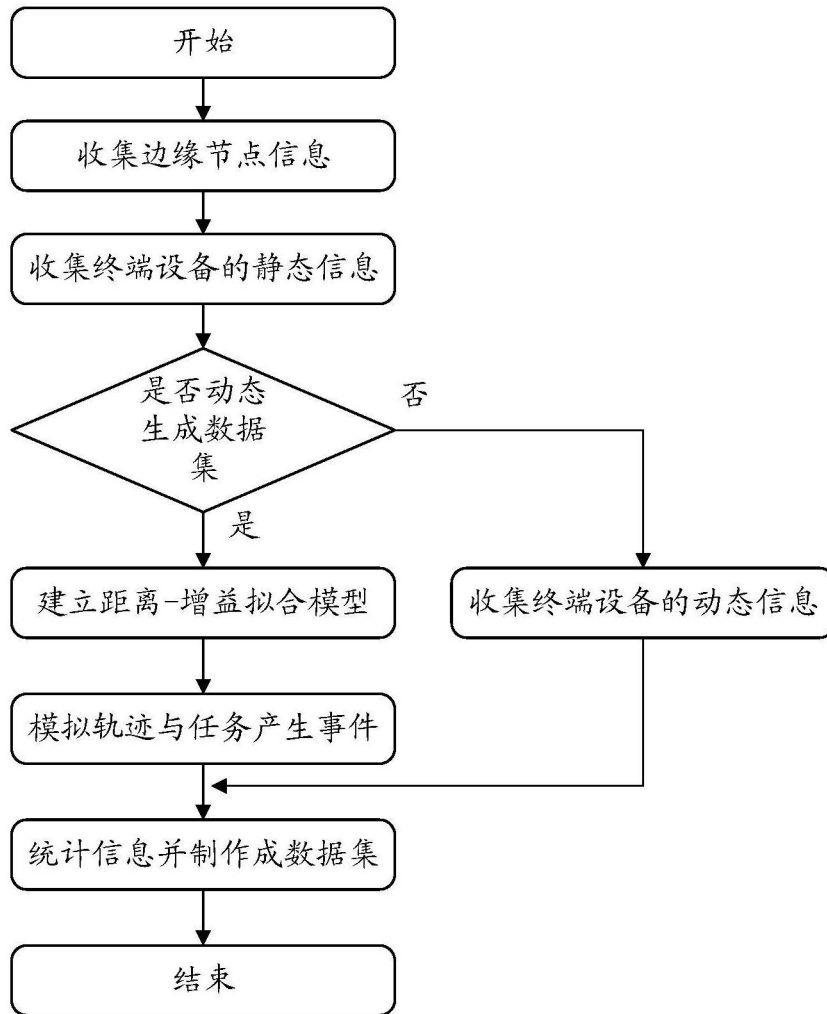


图7

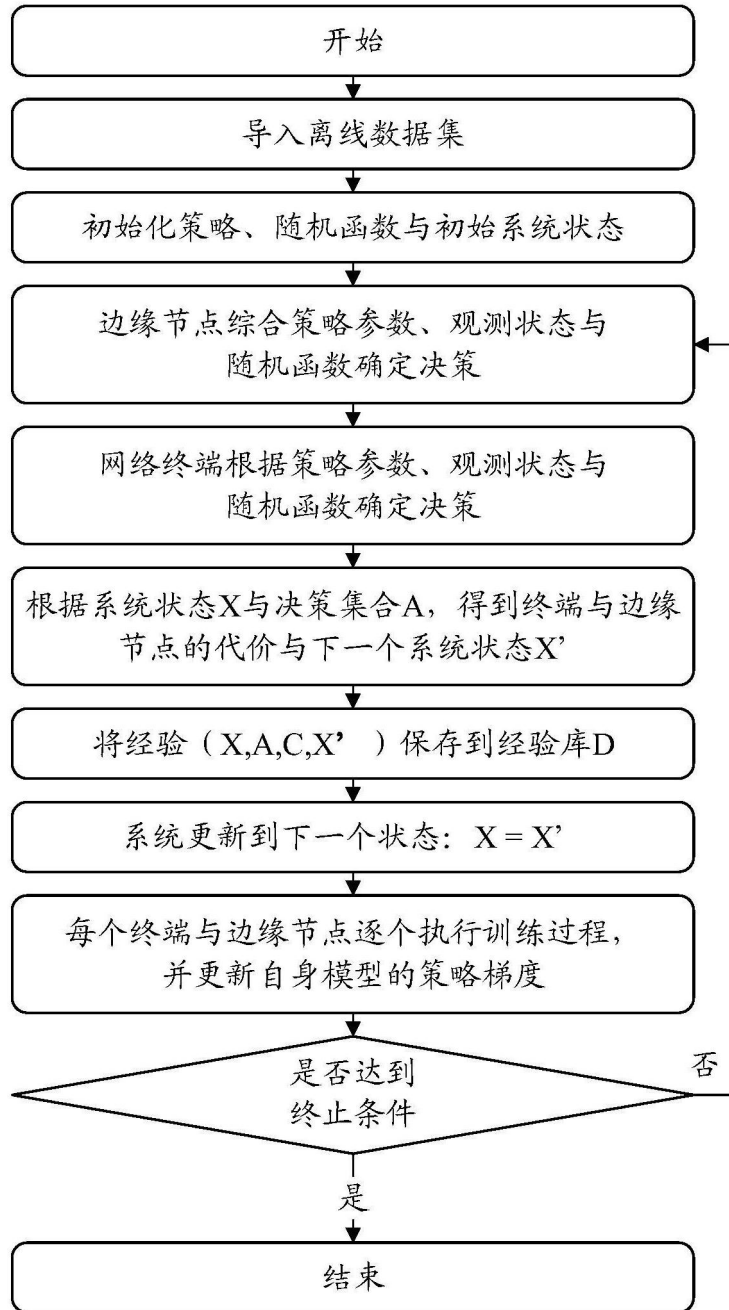


图8

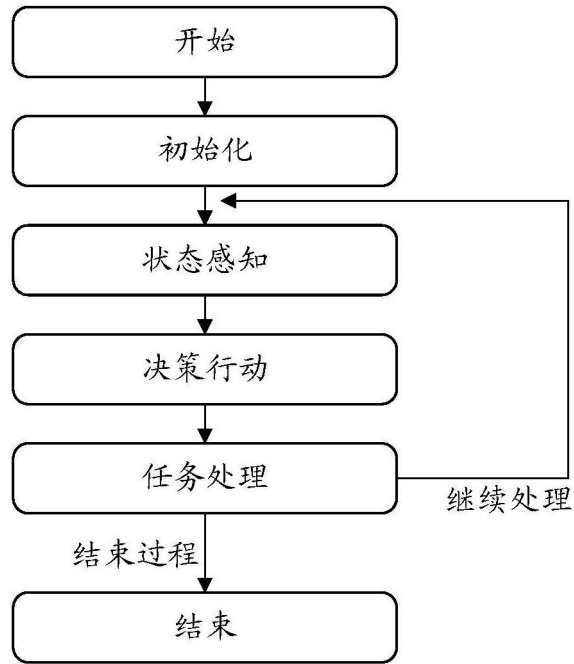


图9

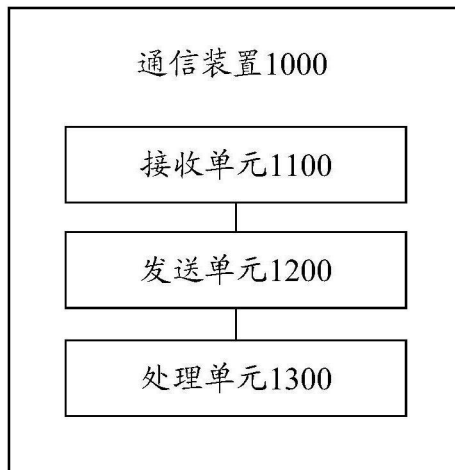


图10



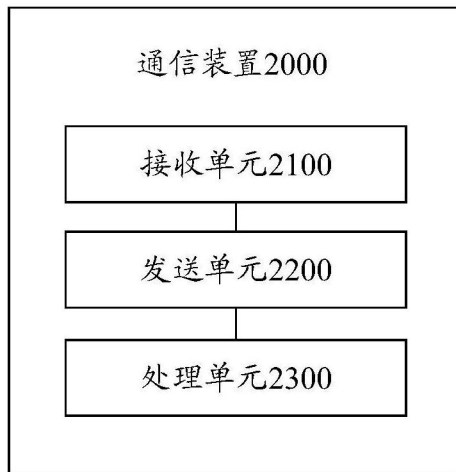


图11

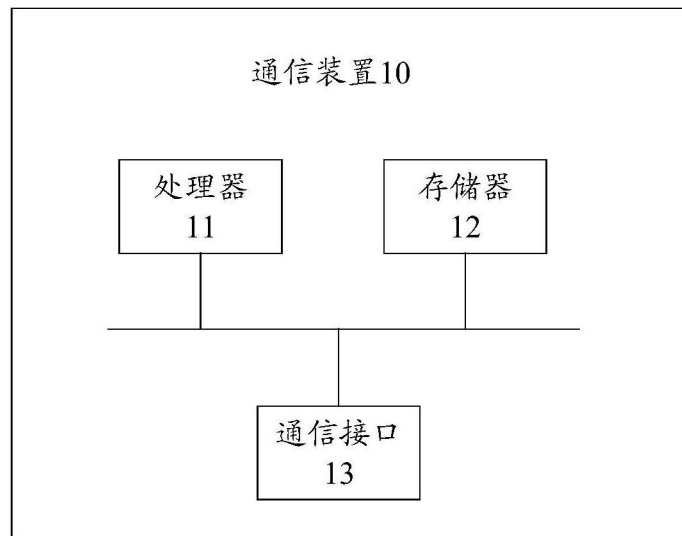


图12

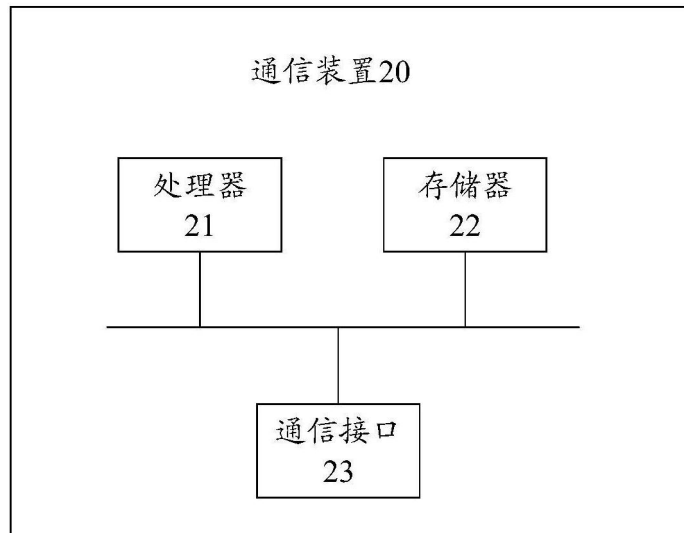


图13