https://doi.org/10.1051/jnwpu/20244250895

一种面向 WLAN 的分布式分层并行仿真方法

贾雯斐, 杨懋, 闫中江, 李波

(西北工业大学 电子信息学院, 陕西 西安 710072)

摘 要:针对采用分布式并行方法仿真 WLAN(wireless local area network)场景时存在的随终端节点个数增加而效率降低的问题,提出了一种面向 WLAN 的分布式分层并行仿真方法。基于 WLAN 的星状网络拓扑结构,令仿真接入节点的进程为主进程,负责 WLAN 全网中其他仿真节点的时间同步;将所有仿真终端节点的进程均匀分为若干组,由组长负责该组内进程的同步。在主进程广播仿真开始事件后,组长进程先收集本组组员终端节点推进结束消息,当收齐后再向主进程汇报。形成"主进程-组长进程-组员进程"的 3 层分层结构。在不同计算负荷下,仿真分析并得到了分层仿真方法的时间增益因子闭合表达式。仿真结果表明,与现有不分层的仿真方法相比,当平均计算负荷为 1.2 倍单位时长、节点个数为 100 时,所提分层仿真方法的增益可达 50%。

关 键 词:WLAN;分层算法;并行仿真

中图分类号:TN92 **文献标志码**:A **文章编号**:1000-2758(2024)05-0895-08

无线局域网(wireless local area network,WLAN)因为其部署成本低和可扩展性好等优点,得到了迅猛发展。有报告[1]指出:2021 年全球企业 WLAN 收入到达 76 亿美元,同比增长 20.4%。电气和电子工程师协会(institute of electrical engineers,IEEE)制定的第一代 WLAN 标准 IEEE802.11[2]成为 WLAN 发展的重要里程碑。为了应对更大数据量、更密集场景部署和更低时延要求带来的挑战,下一代 WLAN 的标准[3]计划于 2024 年发布。

无线通信仿真系统相较于搭建硬件仿真环境具有成本低、可拓展性高和易于调整等特点。利用仿真软件可以较快地搭建仿真场景以验证和改进协议,极大地提高了协议开发的效率。这使得软件仿真成为了标准化推动过程中,开发和验证网络协议的重要工具。NS-3 是一款用于 WLAN 的主流仿真软件,始于 2006 年[4],目前已经迭代发展到 3.39 版本[5]。其对外提供的大量扩展接口让用户可以方便地开发需要的模块或者构建自己的通信场景。由于其基于社区开源的特点以及网络计算的快速发展,NS-3 的维护和开发从未停止[6]。文献[7]通过 NS-3 仿真软件搭建了 DSC(dynamic sensitivity control)仿真平台和上行 OFDMA(orthogonal frequency division multiple access)仿真平台,分别验证了所提 DSC 算法和上行 OFDMA 算法的性能。文献[8]通过 NS-2 仿真软件搭建的仿真平台,验证了 MU MAC(multi-user media access control)协议相较于 802.11ac 协议的性能增益。

随着通信网络的快速发展,网络协议和仿真系统模型变得复杂[9]。在仿真高密集网络或重负载行为时,仿真时长增长迅速,使仿真的效率降低,阻碍协议算法的开发和验证[10]。并行仿真旨在用并行的计算方式提高仿真的效率[11]。文献[12]提出了基于多核 CPU 的仿真计算并行加速方法和基于集群的高效 CPU 并行计算方法,有效提高了通信仿真效率。文献[13]提出 2 级混合并行结构,对原有的仿真程序进行改写,测试结果证明了分布式并行计算的可行性和有效性。文献[14]将划分好的子任务分配给不同节点进行仿真,通过进程间通信保证各节点间的同步,测试结果表明分布式并行仿真的有效性。

NS-3 也面临着需要加速仿真的问题。文献[10]通过减少 NS-3 仿真平台中不必要的仿真事

收稿日期:2023-07-25

基金项目:国家自然科学基金(61771392,61871322,61771390)资助

作者简介:贾雯斐(1999—),硕士研究生

通信作者:闫中江(1983—),副教授 e-mail:zhjyan@nwpu.edu.cn

件,提高仿真速度。为了支持分布式并行仿真,NS-3 使用标准的消息传递接口(message passing interface,MPI)处理多个进程之间发送和接收消息的过程。文献[15]从优化同时刻的异常事件任务、改进网络划分算法和流量分摊方法三方面改进并行方法以提高仿真效率。然而,NS-3 等仿真平台采用分布式并行方法仿真 WLAN 场景依然存在问题。接入点(access point,AP)是指无线局域网用户终端接入网络的设备。终端(station,STA)是指每一个连接到无线网络中的用户设备。空消息(null message)是指子进程完成当前周期的仿真任务后,向目标进程发送的等待同步标记信息。在利用空消息同步策略确保 AP 和 STA 之间的仿真同步时,存在随 STA 个数增长而效率降低的问题。

针对上述问题,本文提出了一种面向 WLAN 的分布式分层并行仿真方法,仿真结果表明所提方法可以有效降低 AP 收集所有 STA 的空消息需要消耗的时间随 STA 个数增大而增加的幅度,进而解决仿真效率降低的问题。

1 系统建模

1.1 WLAN 系统模型

WLAN 的基本元素为接入点和终端。本文讨论的背景是基于 AP 组建的基础无线网络。其拓扑形式为常见的星状网络拓扑,如图 1 所示。

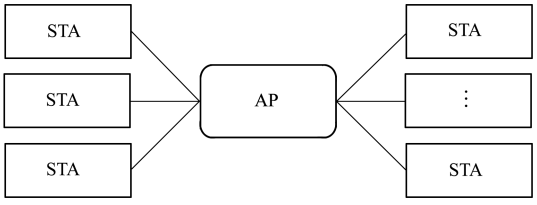

图 1 WLAN 中的星状拓扑

AP 负责接收和转发其他 STA 发出的信息;STA 只需要将试图发送的信息发送给 AP,并接收从 AP 转发而来的信息。也就是说网络中所有的通信都是由 AP 来转发完成。

1.2 仿真系统模型

无论是星状拓扑还是非星状拓扑,利用分布式并行仿真方法加速 WLAN 协议的仿真时,为了充分利用多核的并行性能,每个处理器核心上只部署一个节点(AP 或 STA)的仿真进程。仿真 AP 的进程与仿真 STA 的进程通过 MPI 建立点对点连接以传递空消息,从而同步进程。为方便后续讨论,将仿真 AP 的进程命名为主进程,将仿真 STA 的进程命名为子进程。假设有 n 个子进程和 1 个主进程,形成的网络拓扑如图 2 所示。

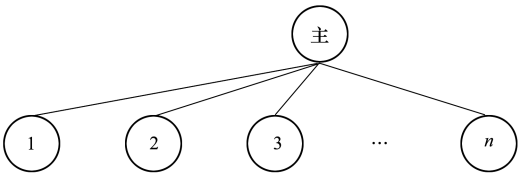

图 2 仿真系统网络拓扑

分布式并行仿真需要确保不同进程间的时间同步。因为空消息同步策略具有良好的拓展性,所以选择此策略进行主、子进程之间的仿真同步。工作流程如图 3 所示:主进程先广播仿真开始事件,然后接收子进程发来的空消息(也称推进结束消息),当收齐后开始广播下一轮仿真开始事件;子进程收到仿真开始事件后处理事件队列中的仿真任务,当处理完后通过 MPI 向主进程发送空消息,接着阻塞接收下一轮仿真开始事件。

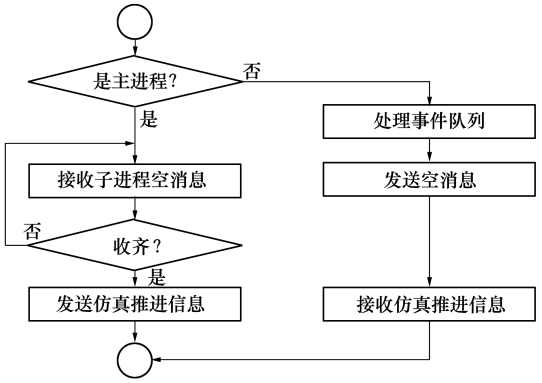

图 3 传统空消息同步策略

为方便后续分析与讨论,做如下定义。在给出定义的基础上,做部分假设。

定义 1 计算负荷:表示子进程在一个同步周期内完成仿真任务需要消耗的墙上时间(wall-clock time)。计算负荷为一随机变量,用参数 μ 表示均值。

定义 2 t_{null}:表示每个子进程发送的空消息被目标进程接收处理的时间。

假设 1 每个子进程的计算负荷是独立同分布的随机变量。

假设 2　t_{null} 的大小是固定的[5]。

假设 3　主进程同时能接收处理一个空消息。

用 T_{nh} 表示采用传统空消息同步策略时,主进程收齐所有子进程的空消息所需的时间。按照图 3 所示的基于传统空消息同步策略的分布式并行仿真方法,T_{nh} 的取值满足不等式(1)。

$$T_{nh} \geqslant n \cdot t_{null} \tag{1}$$

即 T_{nh} 的最小值等于 n 个子进程发送空消息到主进程被接收处理所需时间的线性叠加。这会导致当 STA 个数较大时,AP 收集所有 STA 的空消息消耗的时间占整个仿真时间的比重变大,导致仿真效率降低。

该问题由 t_{null} 的线性叠加导致,可以通过引入分层思想消除 AP 收集所有 STA 的空消息消耗的时间与 n 个 t_{null} 之间的线性关系,提高仿真效率。

2　分布式分层并行仿真方法

2.1　核心思想与工作流程

分布式分层并行仿真方法包含分层流程和基于分层结果的仿真方法。对仿真系统中的 n 个子进程进行分组,每组有 m 个子进程,共分为 n/m 组。每个组包含 1 个组长进程和 $m-1$ 个组员进程。分组之后的子进程和 1 个主进程形成如图 4 所示的拓扑。

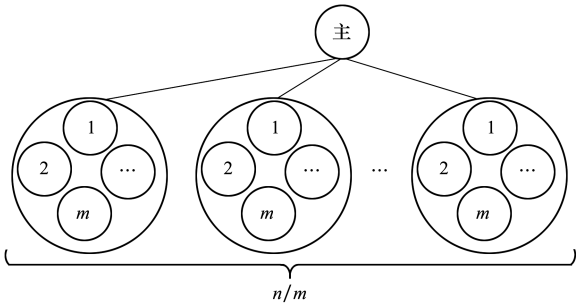

图 4　分层下的仿真系统网络拓扑

基于分层结果的仿真方法工作流程如图 5 所示:主进程先广播仿真开始事件,然后接收组长进程发来的空消息,当收齐后开始广播下一轮仿真开始事件;组员进程收到仿真开始事件后处理事件队列中的仿真任务,当处理完后通过 MPI 向组长进程发送空消息,接着阻塞接收下一轮仿真开始事件;组长进程收到仿真开始事件后处理事件队列中的仿真任务,当处理完后开始接收组员进程发来的空消息,当收齐后通过 MPI 向主进程发送空消息,接着阻塞接收下一轮仿真开始事件。用 T_h 表示采用分布式分层并行仿真方法时,主进程收齐所有子进程的空消息所需的时间。

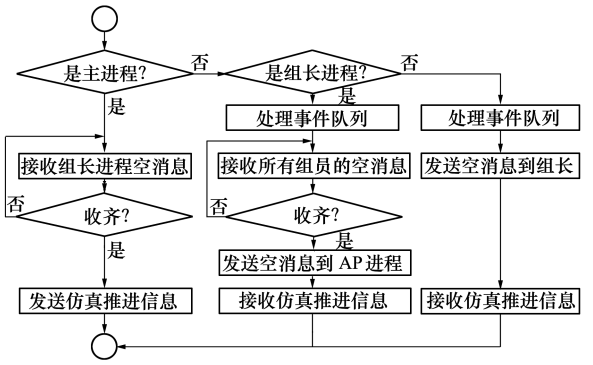

图 5　分布式分层并行仿真方法

2.2　增益因子

将分布式分层并行仿真方法应用于分布式并行仿真是为了消除 AP 收集所有 STA 的空消息消耗的时间与 n 个 t_{null} 之间的线性关系,即减少主进程收齐所有子进程的空消息所需的时间随子进程个数增大而增加的幅度,进而提高仿真效率。定义(2)式所示变量为增益因子以衡量所提分布式分层并行仿真方法的性能增益。

$$G = -\frac{T_h - T_{nh}}{T_{nh}} \tag{2}$$

将在第 3 节理论建模并分析 G 的值。

3　理论分析

平均计算负荷 $\mu = 0$ 时,对 G 的分析结论可由平均计算负荷 $\mu \neq 0$ 的分析结论退化得到,因此本节首先分析当 $\mu \neq 0$ 时的增益。考虑主进程在一次仿真同步过程中用于收集所有子进程的空消息所需的时间,为方便讨论与分析,做出如下假设:

假设 4　n 个子进程收到仿真开始事件的时间是相同的,不妨将其固定为 0。

假设 5　组长进程必须完成当前同步周期内的仿真任务后才能接收组员进程的空消息。

3.1　平均计算负荷非零时($\mu \neq 0$)的增益 G 分析

首先推导计算不采用和采用分层算法下的仿真耗时 T_{nh} 和 T_h 的表达式,然后求解得到增益因子 G 的表达式,最后对增益因子做出分析。

3.1.1 不采用分层算法

设 n 个子进程和 1 个主进程采用基于传统空消息同步策略的分布式并行仿真方法,此时 T_{nh} 的计算过程如下。令 $T_i(i=1,2,\cdots,n,n\in\mathbf{N}^+)$ 表示子进程 i 的计算负荷。$T_{(i)}$ 表示第 i 个空消息到达主进程的时间。主进程处理第 i 个空消息之前,必须等待前 $i-1$ 个空消息都被处理完毕。因此,主进程处理完第 i 个空消息所需的时间 X_i 如(3) 式所示。

$$X_i=\max(T_{(i)},X_{i-1})+t_{\mathrm{null}},\ i\geqslant 2 \tag{3}$$

主进程处理完第 n 个空消息所需的时间 X_n 可以通过(4) 式所示的递推公式组得到。

$$\begin{aligned}X_1&=T_{(1)}+t_{\mathrm{null}}\\X_2&=\max(T_{(2)},X_1)+t_{\mathrm{null}}\\&\vdots\\X_n&=\max(T_{(n)},X_{n-1})+t_{\mathrm{null}},\ n\geqslant 3\end{aligned} \tag{4}$$

利用递推公式组可化简 X_n 的表达式为

$$X_n=\max_{i=1\sim n}[T_{(i)}-(i-1)\cdot t_{\mathrm{null}}]+n\cdot t_{\mathrm{null}} \tag{5}$$

式中,$T_{(i)}$ 为 T_i 的顺序统计量。T_i 是独立同分布的随机变量,其分布函数为 $F_{T_i}(x)$,密度函数为 $f_{T_i}(x)$。根据顺序统计量的性质,$T_{(i)}$ 的分布函数和密度函数如(6) 式所示。

$$\begin{aligned}F_{T_{(i)}}(x)&=\sum_{k=i}^{n}C_n^k[F_{T_i}(x)]^k[1-F_{T_i}(x)]^{n-k}\\f_{T_{(i)}}(x)&=nC_{n-1}^{i-1}F_{T_i}^{i-1}(x)[1-F_{T_i}(x)]^{n-i}f_{T_i}(x)\end{aligned} \tag{6}$$

综上,T_{nh} 的取值如(7)式所示。

$$T_{\mathrm{nh}}=\max_{i=1\sim n}[T_{(i)}-(i-1)\cdot t_{\mathrm{null}}]+n\cdot t_{\mathrm{null}} \tag{7}$$

3.1.2 采用分层算法

设 n 个子进程和 1 个主进程采用分布式分层并行仿真方法,此时 T_{h} 的计算过程如下:令 $T_{i,0}(i=1,2,\cdots,n/m)$ 表示组长进程 i 的计算负荷,$T_{i,j}(i=1,2,\cdots,n/m,j=1,2,\cdots,m-1)$ 表示第 i 组的组员进程 j 的计算负荷。$T_{(i,j)}$ 表示第 j 个空消息到达组长进程 i 的时间。组长进程必须在完成当前同步周期内的仿真任务后才能接收组员进程的空消息,所以组长进程 i 处理完第 1 个空消息所需的时间 $X_{1,i}$ 如(8) 式所示。

$$X_{1,i}=\max(T_{(i,1)},T_{i,0})+t_{\mathrm{null}} \tag{8}$$

组长进程处理第 j 个空消息之前,必须等待前 $j-1$ 个空消息都被处理完毕。因此,组长进程 i 处理完第 j 个空消息所需的时间 $X_{j,i}$ 如(9) 式所示。

$$X_{j,i}=\max(T_{(i,j)},X_{j-1,i})+t_{\mathrm{null}},\ j\geqslant 2 \tag{9}$$

组长进程 i 处理完第 $m-1$ 个空消息所需的时间 $X_{m-1,i}$ 可以通过(10) 式所示的递推公式组得到。

$$\begin{aligned}X_{1,i}&=\max(T_{(i,1)},T_{i,0})+t_{\mathrm{null}}\\X_{2,i}&=\max(T_{(i,2)},X_{1,i})+t_{\mathrm{null}}\\&\vdots\\X_{m-1,i}&=\max(T_{(i,m-1)},X_{m-2,i})+t_{\mathrm{null}},\ m\geqslant 3\end{aligned} \tag{10}$$

可利用递推公式组化简 $X_{m-1,i}$ 的表达式为

$$X_{m-1,i}=\max\{\max_{j=1\sim m-1}[T_{(i,j)}-(j-1)\cdot t_{\mathrm{null}}],T_{i,0}\}+(m-1)\cdot t_{\mathrm{null}} \tag{11}$$

式中,$T_{i,0}$ 是均值为 μ 的随机变量。$T_{(i,j)}$ 为 $T_{i,j}(i=1,2,\cdots,n/m,j=1,2,\cdots,m-1)$ 的顺序统计量。$T_{i,j}$ 是独立同分布的随机变量,其分布函数为 $F_{T_{i,j}}(x)$,密度函数为 $f_{T_{i,j}}(x)$。根据顺序统计量的性质,$T_{(i,j)}$ 的分布函数和密度函数如(12) 式所示。

$$\begin{aligned}F_{T_{(i,j)}}(x)&=\sum_{k=j}^{n}C_n^k[F_{T_{i,j}}(x)]^k[1-F_{T_{i,j}}(x)]^{n-k}\\f_{T_{(i,j)}}(x)&=nC_{n-1}^{j-1}F_{T_{i,j}}^{j-1}(x)[1-F_{T_{i,j}}(x)]^{n-j}f_{T_{i,j}}(x)\end{aligned} \tag{12}$$

$X_{(m-1,i)}$ 表示第 i 个来自组长子进程的空消息到达主进程的时间。主进程在处理第 i 个空消息之前,必须等待前 $i-1$ 个空消息都被处理完毕。因此,主进程处理完第 i 个空消息所需的时间 Y_i 如(13) 式所示。

$$Y_i=\max(X_{(m-1,i)},Y_{i-1})+t_{\mathrm{null}},\ i\geqslant 2 \tag{13}$$

主进程处理完第 n/m 个空消息所需的时间 $Y_{n/m}$ 可以通过(14) 式所示的递推公式组得到。

$$\begin{aligned}Y_1&=X_{(m-1,1)}+t_{\mathrm{null}}\\Y_2&=\max(X_{(m-1,2)},Y_1)+t_{\mathrm{null}}\\&\vdots\\Y_{n/m}&=\max(X_{(m-1,n/m)},Y_{n/m-1})+t_{\mathrm{null}},\ n/m\geqslant 3\end{aligned} \tag{14}$$

利用递推公式组化简 $Y_{n/m}$ 的表达式为

$$Y_{n/m}=\max_{i=1\sim n/m}[X_{(m-1,i)}-(i-1)\cdot t_{\mathrm{null}}]+n/m\cdot t_{\mathrm{null}} \tag{15}$$

综上,T_{h} 的取值如(16)式所示。

$$T_{\mathrm{h}}=\max_{i=1\sim n/m}[X_{(m-1,i)}-(i-1)\cdot t_{\mathrm{null}}]+n/m\cdot t_{\mathrm{null}} \tag{16}$$

式中,$X_{(m-1,i)}$ 为 $X_{m-1,i}$ 的顺序统计量。

3.1.3 增益因子

平均计算负荷非零时 ($\mu\neq 0$) 的增益因子表达式如(17)式所示。

$$G=\frac{\{\max\limits_{i=1\sim n}[T_{(i)}-(i-1)\cdot t_{\text{null}}]+n\cdot t_{\text{null}}-\max\limits_{i=1\sim n/m}[X_{(m-1,i)}-(i-1)\cdot t_{\text{null}}]+n/m\cdot t_{\text{null}}\}}{\max\limits_{i=1\sim n}[T_{(i)}-(i-1)\cdot t_{\text{null}}]+n\cdot t_{\text{null}}} \tag{17}$$

从(17)式可以得到如下结论:增益不仅与 STA 总个数及每组内 STA 个数有关,还与 t_{null} 的大小和计算负荷的大小有关。在每个子进程发送的空消息被目标进程接收处理的时间 t_{null} 一定时,计算负荷越大,每个 STA 完成一个同步周期中的仿真任务需要消耗的墙上时间就越长,T_{h} 和 T_{nh} 间的差距越无法由 t_{null} 所体现。由于此时分层算法带来的并行性效果有所下降,增益相应地也会下降。

3.1.4　计算负荷服从指数分布

由于通信系统中各个节点的包到达间隔一般服从指数分布,因此,仿真每个节点的子进程的计算负荷可假设为独立同分布的随机变量且服从参数为 λ 的指数分布,此时 $\mu=1/\lambda$。

此时增益因子 G 的表达式可以进一步退化。(17)式中的 $X_{(m-1,i)}$ 是 $X_{m-1,i}$ 的顺序统计量。$X_{m-1,i}$ 的表达式由(11)式给出。$T_{(i,j)}$ 的分布函数和密度函数由(18)式给出。

$$F_{T_{(i,j)}}(x)=\sum_{k=j}^{n}C_n^k(1-\mathrm{e}^{-\lambda x})^k(\mathrm{e}^{-\lambda x})^{n-k}\quad x\geqslant 0$$

$$f_{T_{(i,j)}}(x)=\lambda nC_{n-1}^{j-1}(1-\mathrm{e}^{-\lambda x})^{(j-1)}(\mathrm{e}^{-\lambda x})^{n-j+1}\quad x\geqslant 0 \tag{18}$$

$T_{i,0}$ 的分布函数和密度函数由(19)式给出。

$$F_{T_{i,0}}(x)=1-\mathrm{e}^{-\lambda x}\quad x\geqslant 0$$

$$f_{T_{i,0}}(x)=\lambda\mathrm{e}^{-\lambda x}\quad x\geqslant 0 \tag{19}$$

$T_{(i)}$ 的分布函数和密度函数由(20)式给出。

$$F_{T_{(i)}}(x)=\sum_{k=i}^{n}C_n^k(1-\mathrm{e}^{-\lambda x})^k(\mathrm{e}^{-\lambda x})^{n-k}\quad x\geqslant 0$$

$$f_{T_{(i)}}(x)=\lambda nC_{n-1}^{i-1}(1-\mathrm{e}^{-\lambda x})^{(i-1)}(\mathrm{e}^{-\lambda x})^{n-i+1}\quad x\geqslant 0 \tag{20}$$

3.2　平均计算负荷为零时($\mu=0$)的增益 G 分析

在推导出 $\mu\neq 0$ 的结论后,对相关公式做部分退化处理即可得到 $\mu=0$ 的相关结论。先分析 $\mu=0$ 情形下 T_{nh} 和 T_{h} 的表达式,然后求解得到增益因子 G 的表达式,最后对增益因子做出分析。

3.2.1　不采用分层算法

因为 $\mu=0$,所以(7)式中的 $T_{(i)}$ 满足 $T_{(i)}\equiv 0$。此时(7)式可退化为(21)式。

$$T_{\text{nh}}=n\cdot t_{\text{null}} \tag{21}$$

3.2.2　采用分层算法

因为 $\mu=0$,所以(11)式可退化为(22)式。(16)式可退化为(23)式。

$$X_{m-1,i}=(m-1)\cdot t_{\text{null}} \tag{22}$$

$$T_{\text{h}}=(n/m+m-1)\cdot t_{\text{null}} \tag{23}$$

3.2.3　增益因子

平均计算负荷为零时($\mu=0$)的增益因子如(24)式所示。此时增益只与 STA 总个数以及每组内 STA 个数有关。

$$G=1-\frac{m^2+n-m}{mn} \tag{24}$$

4　仿真分析

针对 $\mu=0$ 的情形,本节仿真了增益随 STA 总个数以及每组内 STA 个数增加时的变化情况。针对 $\mu\neq 0$ 的情形,本节进行计算负荷服从指数分布情况下的仿真分析,包括增益随 STA 总个数、每组内 STA 个数 μ 值大小和 t_{null} 值大小改变的变化情况。本文选择的对比算法为 NS-3 采用的基于传统空消息同步策略的分布式并行仿真方法[5],后续增益结果均为所提分布式分层仿真方法相较于对比算法能够取得的增益。

4.1　平均计算负荷为零时($\mu=0$)的算法性能

仿真节点个数取值在 40~480 范围内,以 40 为步进值变化,组内节点个数取值为 10,20 或 40 时的增益情况。

图 6 中横坐标代表了节点个数,纵坐标代表了所提方案相较于采用空消息同步策略的传统分布式

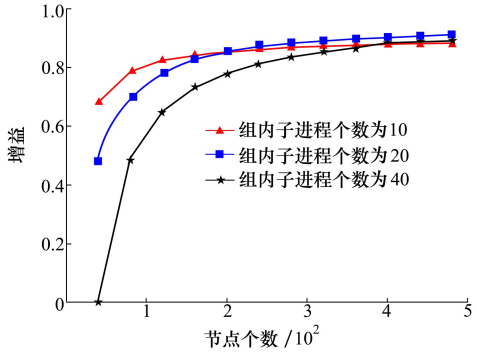

图 6　不同组内成员个数下,增益变化情况

并行仿真的增益,数值越大,所提方案降低 AP 收集所有 STA 的空消息需要消耗的时间随 STA 个数增大而增加的幅度越大。从图 6 可知,组内子进程个数一定,随着节点个数的增加,增益不断提高。节点数达到 480 时,增益可以达到 90%左右。节点个数不变,组内子进程个数增加,增益并不是一直降低。

4.2 平均计算负荷非零时($\mu \neq 0$)的算法性能

仿真节点个数取值在 40~100 范围内,以 20 为步进值变化,组内节点个数取值为 20,计算负荷的均值 μ 取值为 1 倍单位时间,t_{null} 取值在 0.12~0.16 倍单位时间范围内,以 0.02 倍单位时间为步进值变化时主进程收齐所有子进程空消息所需的时间和增益情况。

图 7 中横坐标是节点个数,纵坐标是主进程收齐所有子进程的空消息所需的时间。从图 7 可知,t_{null} 不变,节点个数较大时,采用分层的耗时相较于非分层有明显减少,且减少量随着节点个数增加而增加。相同节点个数时,t_{null} 越大,采用分层的耗时相较于非分层的减少量越大。

图 8 中横坐标代表了节点个数,纵坐标代表了所提方案相较于采用空消息同步策略的传统分布式并行仿真的增益。从图 8 可知,节点个数不变,随着 t_{null} 的增大,增益变大。因为分层带来的并行效果主要是减少 t_{null} 的线性叠加。t_{null} 不变,随着节点个数的增加,增益不断提高。原因是节点个数越多,分层带来的并行性更高。节点个数为 100,$t_{\text{null}}=0.16$,可以取得 60%以上的增益;$t_{\text{null}}=0.12$,可以取得 50%以上的增益。

仿真节点个数取值在 40~100 范围内,以 20 为步进值变化,组内节点个数取值为 20,计算负荷均值 μ 取值在 1~1.2 倍单位时间范围内,以 0.1 倍单位时间为步进值变化,t_{null} 取值为 0.14 倍单位时间时的主进程收齐所有子进程的空消息所需的时间和增益情况。

图 9 中横坐标是节点个数,纵坐标是主进程收齐所有子进程的空消息所需时间。从图 9 可知,计算负荷均值不变,节点个数较大时,采用分层的耗时相较于非分层有明显减少,且减少量随着节点个数增加而增加。计算负荷均值变化对分层的耗时影响较大,对非分层的耗时影响较小。相同节点个数时,计算负荷均值越大,采用分层的耗时相较于非分层的减少量越小。

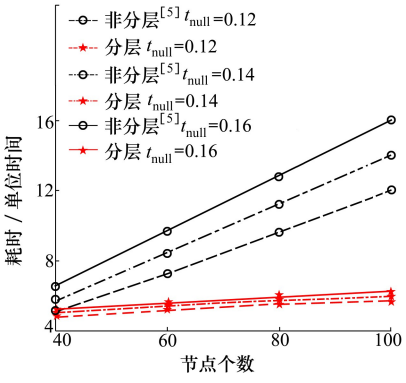

图 7 不同 t_{null} 下,耗时变化情况

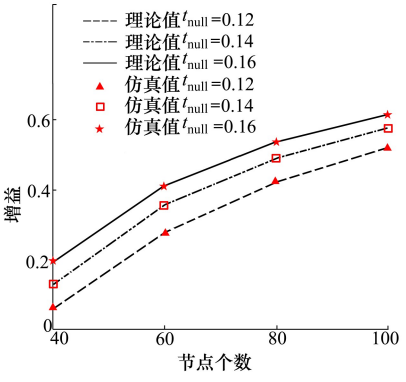

图 8 不同 t_{null} 下,增益变化情况

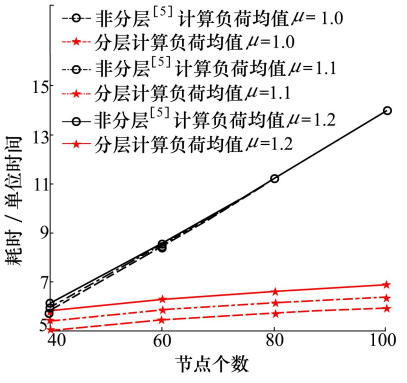

图 9 不同计算负荷均值下,耗时变化情况

图 10 中横坐标代表了节点个数,纵坐标代表了所提方案相较于采用空消息同步策略的传统分布式并行仿真的增益。从图 10 可知,节点个数不变,随着计算负荷均值增大,增益降低。因为计算负荷的增加降低了分层带来的并行效果。节点个数为 100,计算负荷均值为 1,可以取得近 60%的性能增益。

仿真节点个数取值在 40~160 范围内,以 40 为步进值变化,组内节点个数取值为 10,20 或 40,计算负荷的均值 μ 取值为 1 倍单位时间,t_{null} 取值为 0.14倍单位时间时的主进程收齐所有子进程的空消息所需的时间和增益情况。

图 11 中横坐标是节点个数,纵坐标是主进程收齐所有子进程的空消息所需的时间。非分层不涉及分组,故非分层在图中只有一条曲线。从图中可知,节点个数较大时,采用分层的耗时相较于非分层有明显减少,且减少量随着节点个数增加而增加。组内成员个数变化对分层的耗时影响较大。

图 12 中横坐标代表了节点个数,纵坐标代表了所提方案相较于采用空消息同步策略的传统分布式并行仿真的增益。从图 12 可知,增益的理论值和仿真值吻合良好。节点个数在 40~160 之间变化时,随着组内子进程个数增加,分层相较于非分层的增益降低。

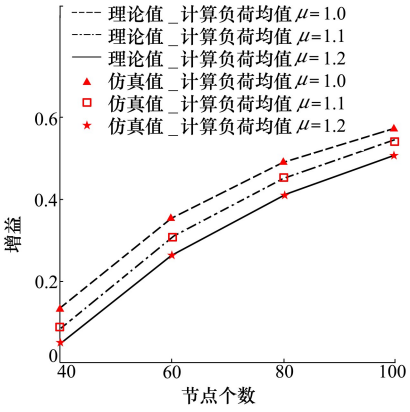

图 10　不同计算负荷均值下,增益变化情况

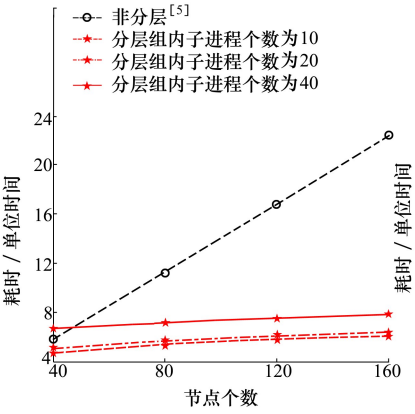

图 11　不同组内成员个数下,耗时变化情况

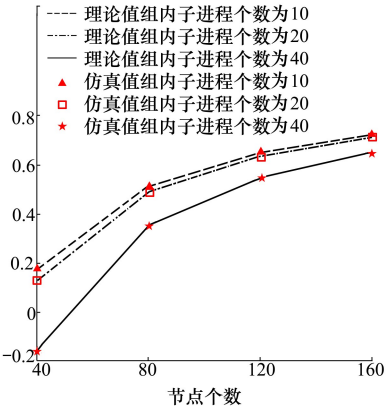

图 12　不同组内成员个数下,增益变化情况

5　结　论

诸如 NS-3 等仿真平台采用分布式并行方法仿真 WLAN 场景时,传统的空消息同步策略会导致仿真 AP 节点的进程收齐仿真 STA 节点的进程的推进结束消息所需要的时间随 STA 个数增长而线性增长最终导致效率降低问题。针对该问题本文提出一种面向 WLAN 的分布式分层并行仿真方法,并分析了分层相较于非分层的增益。仿真表明所提方法可有效降低仿真耗时。

参考文献:

[1] BRANDON Butler. Worldwide enterprise WLAN market shows strong growth in Q4 and the full year 2021, according to IDC [EB/OL]. (2022-03-09)[2023-07-14]. https://www.idc.com/getdoc.jsp? containerId=prUS48940222

[2] CROW B P, WIDJAJA I, KIM J G, et al. IEEE 802.11 wireless local area networks[J]. IEEE Communications Magazine, 1997, 35(9): 116-126

[3] KHOROV E, LEVITSKY I, AKYILDIZ I F. Current status and directions of IEEE 802.11be, the future Wi-Fi 7[J]. IEEE Access, 2020, 8: 88664-88688

[4] 茹新宇, 刘渊, 陈伟. 新网络仿真器 NS-3 的研究综述[J]. 微型机与应用, 2017, 36(20): 14-16
RU Xinyu, LIU Yuan, CHEN Wei. Research review of new network simulator NS-3[J]. Micromachinesand Applications, 2017, 36(20): 14-16 (in Chinese)

[5] NS-3 Project. NS-3 network simulator[EB/OL]. (2023-07-05)[2023-07-14]. https://www.nsnam.org/releases/ns-3-39/documentation/

[6] 刘瑞军. 基于 NS-3 的多波束卫星通信仿真系统设计与实现[D]. 西安: 西安电子科技大学, 2021
LIU Ruijun. Based on NS-3 of multibeam satellite communication simulation system design and implementation[D]. Xi'an : Xidian University, 2021 (in Chinese)

[7] 张振亚. 下一代 WLAN 关键技术的仿真平台设计与实现[D]. 西安: 西安电子科技大学, 2015
ZHANG Zhenya. Design and implementation of simulation platform for key technologies of next-generation WLAN[D]. Xi'an: Xidian University, 2015 (in Chinese)

[8] 周润, 李波, 杨懋, 等. 下一代 WLAN 多用户多址接入技术仿真研究[J]. 计算机仿真, 2019, 36(8): 288-292
ZHOU Run, LI Bo, YANG Mao, et al. Simulation research on next-generation WLAN multi-user multi-access technology[J]. Computer Simulation, 2019, 36(8): 288-292 (in Chinese)

[9] VOICU A M, SIMIC L, PETROVA M. Modelling large-scale CSMA wireless networks[C]//2021 IEEE Wireless Communications and Networking Conference, Nanjing, 2021: 1-6

[10] 马迎华. WLAN 仿真事件聚合与 NS-3 加速技术研究[D]. 南京: 南京邮电大学, 2022

MA Yinghua. Research on WLAN simulation event aggregation and NS-3 acceleration technology[D]. Nanjing: Nanjing University of Posts and Telecommunications, 2022 (in Chinese)
[11] XIN Y, YANG M, YAN Z, et al. Non-uniform time slice parallel simulation method based on offline learning for IEEE 802.11ax [C]// Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, 2023: 161-171
[12] 卢红. 新一代无线通信仿真系统中负荷的高效并行计算方法与实现[D]. 北京: 北京邮电大学, 2018
LU Hong. Efficient parallel computing method and implementation of load in a new generation of wireless communication simulation system[D]. Beijing: Beijing University of Posts and Telecommunications, 2018 (in Chinese)
[13] 宁原霆. 基于 MPI 的三维离散元法计算仿真软件改进研究[D]. 吉林: 吉林大学, 2015
NING Yuanting. Research on improvement of computational simulation software of three-dimensional discrete element method based on MPI[D]. Jilin: Jilin University, 2015 (in Chinese)
[14] 蒋化南,张帅,林宇斐,等. 基于 MPI 的分布式并行 Gazebo 仿真优化与测试[J]. 计算机科学, 2021,48(增刊 2):672-677
JIANG Huanan, ZHANG Shuai, LIN Yufei, et al. Simulation optimization and testing of distributed parallel Gazebo based on MPI [J]. Computer Science, 2021, 48(suppl 2): 672-677 (in Chinese)
[15] 袁进. 基于 NS-3 的网络并行模拟仿真方法研究[D]. 成都: 电子科技大学, 2017
YUAN Jin. Research on network parallel simulation method based on NS-3[D]. Chengdu: University of Electronic Science and Technology of China, 2017 (in Chinese)

A WLAN-oriented distributed hierarchical parallel simulation method

JIA Wenfei, YANG Mao, YAN Zhongjiang, LI Bo
(School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: To solve the problem that the efficiency decreases with the increase of terminal nodes in wireless local area network (WLAN) distributed parallel simulation, a WLAN-oriented distributed hierarchical parallel simulation method is proposed. Firstly, considering the WLAN star network topology, the process of the simulated access point (AP) is denoted as the main process, which is responsible for time synchronization with the other processes of the simulated station (STA) nodes in the WLAN network. Secondly, the processes of all simulated STAs are then evenly divided into groups, and the group leader is responsible for the synchronization of processes within the group. Then, after the main process broadcasts the simulation start event, the group leader process firstly collects the end message of the group member's terminal node, and then reports to the main process. A three-layer hierarchical structure is constructed, consisting of the main process layer, the group leader process layer and the group member process layer. Finally, the time gain factor of the hierarchical simulation method is analyzed and obtained in closed-form expression, under the different computational loads. Simulation results show that comparing to the existing non-hierarchical simulation method, when the average computational load is 1.2x time unit and the nodes number is 100, the gain of the proposed method can reach 50%.

Keywords: WLAN; hierarchical algorithm; parallel simulation

引用格式: *贾雯斐, 杨懋, 闫中江, 等. 一种面向 WLAN 的分布式分层并行仿真方法[J]. 西北工业大学学报,2024,42(5): 895-902*
JIA Wenfei, YANG Mao, YAN Zhongjiang, et al. A WLAN-oriented distributed hierarchical parallel simulation method [J]. Journal of Northwestern Polytechnical University, 2024, 42(5): 895-902 (in Chinese)

© 2024 Journal of Northwestern Polytechnical University.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.