Bolt-Dumbo Transformer 的学习和分析报告

胡佳佳

(南开大学网络空间安全学院 密码科学与技术系, 天津 300350)

摘 要 2022 年 Yuan Lu 在 CCS 会议上发表的 *Bolt-Dumbo Transformer: Asynchronous Consensus As Fast As the Pipelined BFT*[1]介绍了一种新的异步共识框架 Bolt-Dumbo Transformer（BDT），旨在优化区块链服务在互联网上的部署。相比与之前共识协议在同步和异步上的单独实现，BDT 结合了两者的优点，通过快速通道（Fastlane）在网络条件良好时提供高效率，同时在网络不稳定或面临攻击时，利用步调同步（Pace-Synchronization）和悲观路径（Pessimistic Path）确保鲁棒性。该框架通过简化的二元协议减少了通信复杂性，并在实验中展示了与现有技术相比的低延迟和高吞吐量。在区块链上实现了安全和效率方面较好的平衡。本次研究报告基于这篇论文进行学习，分析和总结。

关键词 区块链；同步共识；异步共识

Bolt-Dumbo Transformer Learning and Analysis Essay

Hu Jia-jia

(Department of Cryptography Science and Technology, School of Cyberspace Security, Nankai University, Tianjin 300350)

Abstract In 2022, Yuan Lu presented a novel asynchronous consensus framework called Bolt-Dumbo Transformer (BDT) at the CCS conference in a paper titled "Bolt-Dumbo Transformer: Asynchronous Consensus As Fast As the Pipelined BFT" [1]. This framework is designed to optimize the deployment of blockchain services over the Internet. Unlike previous consensus protocols that focused solely on synchronous or asynchronous implementations, BDT integrates the advantages of both, offering high efficiency through a fast lane (Fastlane) during good network conditions and ensuring robustness through pace-synchronization and a pessimistic path when the network is unstable or under attack. By employing a simplified binary agreement, the framework reduces communication complexity and has demonstrated low latency and high throughput in experiments, striking a good balance between security and efficiency in blockchain applications. This study report is based on learning from, analyzing, and summarizing the insights presented in this paper.

Key words Blockchain; Synchronous; Asynchronous Consensus;

1 引言

在引言部分，论文讨论了去中心化技术的爆炸性流行如何创造了对在全球互联网上部署强大的拜占庭容错（BFT）共识的前所未有的需求。BFT共识协议，特别是原子广播（ABC），旨在在n个参与方之间复制不断增长的交易日志，确保安全性和活跃性，即使在敌对控制通信网络和破坏一些参与方的情况下。然而，互联网的动态特性为实现安全且高效的BFT共识协议带来了新的挑战。传统的BFT协议通常基于网络条件的某些假设，例如同步或部分同步假设，但在广域网中这些假设可能不成立，导致这些协议在面对网络攻击或延迟时可能会停滞不前。

论文指出，完全异步的BFT协议能够在没有任何网络同步形式的情况下同时确保安全性和活跃性，但这种高安全性保障并非没有代价。FLP不可能性原理指出，在异步网络中无法确定性地确保安全性和活跃性，因此异步ABC必须运行随机子程序来规避这一不可能性，这增加了其复杂性。尽管近年来出现了一些实现异步ABC的新颖路径，如HoneyBadgerBFT和DumboBFT，但现有的异步共识协议在性能上仍远远落后于确定性的（部分）同步协议，尤其是在关键的延迟指标上。

基于此，论文提出了Bolt-Dumbo Transformer（BDT）框架来实现了结合同步和异步范式优势的BFT共识的问题，接下来我们来了解其算法实现和性能评估。

2 算法概述

论文为部分的异步共识协议提供了一个一般性的优化转化框架，实现了在平稳网络环境下运行fastlane同步快速方案，而在糟糕的网络环境下切换到特定异步安全方案，并最小化了切换的开销。大体上看，BDT分为以下三个阶段：

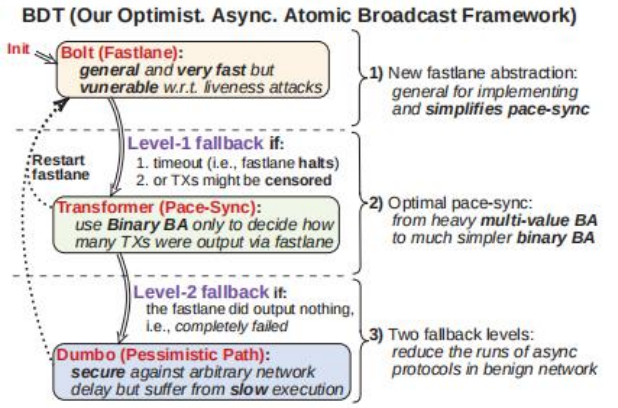


Figure 1: The overview of Bolt-Dumbo Transformer.

1. **快速通道（Fastlane，简称Bolt）：**在这个阶段，BDT运行一个确定性协议作为快速通道，以在同步假设成立的乐观情况下快速进展。这意味着在网络条件良好且没有恶意行为时，系统可以高效地处理交易和区块。
2. **步调同步（Pace-synchronization，简称pace-sync或Transformer）：**如果快速通道由于网络问题或恶意行为者而未能及时进展，就会触发一个快速的步调同步机制。这个机制类似于异步安全中的视图变更，目的是让所有诚实的节点就如何重启或继续达成共识——要么直接重启Bolt快速通道，要么进入悲观路径。
3. **悲观路径（Dumbo）：**如果快速通道完全无法进展，诚实的节点将进入异步BFT的悲观路径，以确保即使在最坏的情况下也能保持系统的活性。这个阶段是BDT框架的最后防线，确保系统的鲁棒性和安全性。

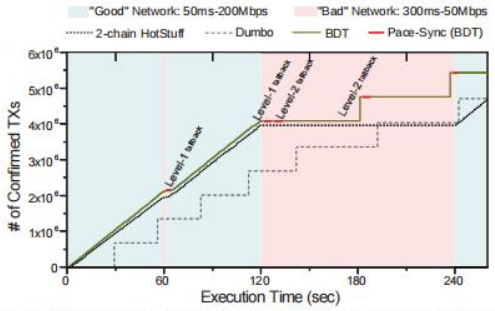


Figure 2: Simulated executions of BDT, 2-chain HotStuff **and** Dumbo-BFT **under fluctuating network (*n*=64). Bad network happens twice: one lasts 2 seconds and one lasts 120 seconds. See Section 7 for the details on simulation setup.**

论文在实验中比较了BDT，基于确定性同步共识的2-chain HotStuff和基于异步共识的Dumbo-BFT在不同网络环境下的TXs确认量。可以发现，在好的网络环境下，BDT和同步共识的效率相当，而在差

的网络环境下，除了需要模式切换的少量开销，和单独实现的 Dumbo-BFT 相比在异步共识的实现上功能相同，而 HotStuff 则因此停机。整体上看，BDT 的效率优于其它两者。这也进一步证明了 BDT 的优势。而这一优势主要是来自 BDT 实现技术的以下四个方面：

1. **新的快速通道抽象：**为了简化复杂的步调同步，BDT 提出了一种新的快速通道抽象，称为可公证的弱原子广播（nw-ABC）。在乐观情况下，它实现了原子广播（ABC），在其他情况下，它确保了“可公证性”：任何输出的区块都有一个足够诚实节点接收到的区块证明。

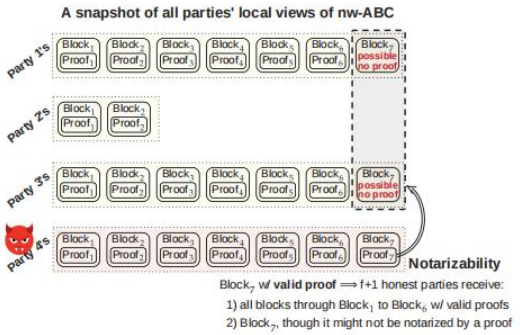


Figure 4: Notarizability of fastlane abstraction (nw-ABC).

这样的 nw-ABC 可以很容易地构建得非常快速，例如，通过一系列简单的（可证明的）多播；更重要的是，可公证性保证了在进入步调同步时，任何两个诚实的节点都将处于相邻的区块，因此可以使用更简单的二元协议来代替以前的异步原子广播或多值协议。

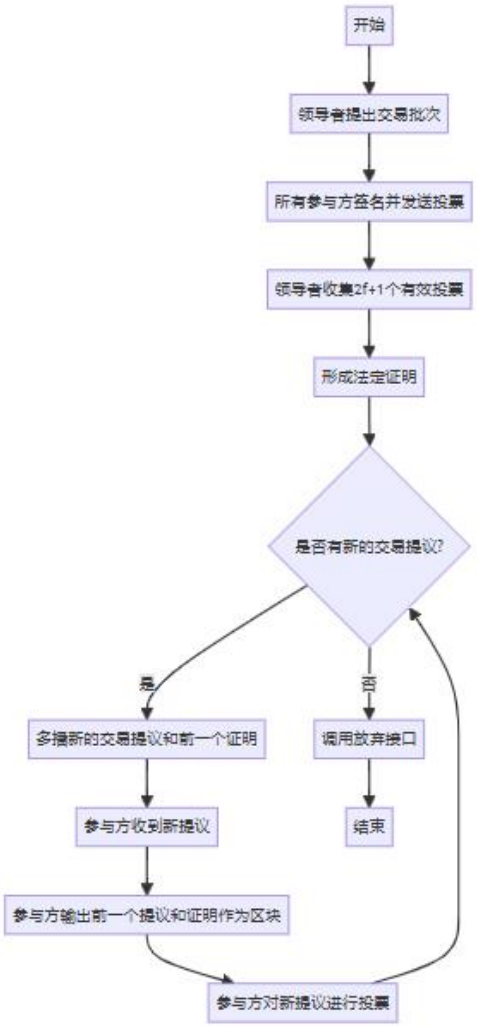
2. **成本最低的步调同步：**利用 nw-ABC 的准备，Transformer 将步调同步简化为一个称为两个连续值的拜占庭协议（tcv-BA）的问题，这本质上是异步二元拜占庭协议（ABBA）。与之前的工作相比，这种方法将步调同步的通信复杂性提高了 O(n)的因子，并且对于步调同步来说本质上是最优的，因为步调同步问题可以被视为异步共识的一个版本，而 ABBA 可以说是最简单的异步共识。在实践中，Transformer 实现了与快速通道延迟类似的最小开销。
3. **尽可能避免悲观路径：**为了进一步利用快速 Transformer 带来的好处，BDT 在步调同步后增加了一个简单检查，创建了两级回退：如果步调同步显示快速通道仍然产生了一些输出，它将立即重新启动另一个快速通道，而无需运行实际的悲观路径。这与以前的工作不同，以前的工作在每次步调同步后总是会运行慢速的悲观路径，这在只有短期网络波动的情况下通常是不必要的成本。
4. **通用框架支持灵活实例化：**BDT 是一个通用框架，允许灵活选择所有三个阶段的底层构建块。例如，论文中展示了两种快速通道的实例化，分别产生了两种 BDT 实现，一种倾向于延迟，另一种倾向于吞吐量，因此可以根据实际应用场景实例化 BDT。此外，Transformer 可以围绕任何异步二元协议构建，从而有可能使用任何更有效的 ABBA 来进一步减少回退开销。

3 算法实现

3.1 Bolt

Bolt 可以以连续多播（Bolt-sCAST）和连续可靠广播(Bolt-sRBC)来实现。

以下给出了 Bolt-sCAST 的实现流程图。



Bolt-sCAST 利用了阈值签名来确保交易的一致性和有效性。

而基于 RBC 实现的 Bolt 的流程为：

与 Bolt-sCAST 相比，基于 RBC 的实现可能在带宽使用上更为均衡，因为使用了可验证信息分散技术。

3.2 Bolt-Dumbo Transformer Framework

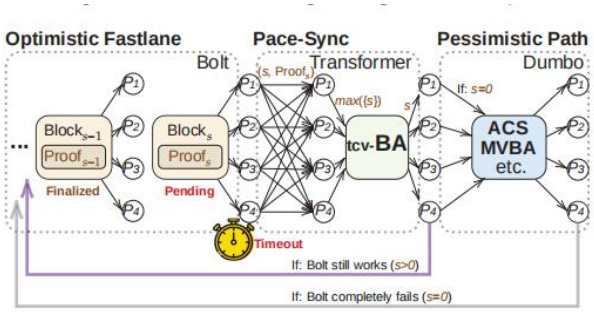


Figure 11: The execution flow of Bolt-Dumbo Transformer

BDT 的 Fastlane 是一个被计时器封装的 Bolt 实例。如果诚实的参与方能够及时收到新的 Bolt 区块，他们会重启计时器以等待下一个 Bolt 区块。否则，当计时器超时，诚实的参与方会多播一个回退请求，该请求包含他们能看到的最新 Bolt 区块的法定证明超时后，每个参与方等待收到 n-f 个带有有效 Bolt 区块证明的回退请求，并进入步调同步（pace-sync）。他们使用接收到的回退请求中的最大区块索引（槽位）作为输入，调用两个连续值的拜占庭协议（tcv-BA）。最终，诚实的参与方进入 tcv-BA，并决定是重试快速通道还是启动悲观路径。

BDT 的安全性来源于以下三个方面：

1. **Transformer 返回一个共同的索引。**所有诚实的参与方必须从 Transformer 获得相同的区块索引，这样他们总是就继续悲观路径（或重试快速通道）的同一个快速通道区块达成一致。这由 tcv-BA 的一致性保证。
2. **Transformer 返回的索引不会“太大”。**对于 Transformer 返回的索引，至少有 f+1 个诚实的参与方已经收到了这个索引之前的所有区块（带有有效证明）。因此，如果任何参与方错过了一些区块，它可以很容易地从这些 f+1 个参与方那里获取正确的区块。这是因为 Bolt 的可公证性防止了对手伪造一个索引高于实际传递区块的快速通道区块的证明。因此，没有诚实的参与方会在 tcv-BA 中输入一个无法检索的区块的索引，然后 tcv-BA 的有效性就简单地保证了这一点。
3. **Transformer 返回的索引不会“太小”。**没有诚实的参与方会撤销任何已经被确定为最终输出的快速通道区块。由于每个诚实的参与方都等待来自不同参与方的 2f+1 个 PaceSync 消息，然后由于 Bolt 的可公证性，至少有一个

PaceSync 消息包含 s-1，其中 s 是所有参与方中最新的快速通道区块。因此，每个诚实的参与方至少输入 tcv-BA 为 s-1。

BDT 的活性则来源于以下三个方面：

1. **快速通道（Fastlane）：**通过设置“超时”参数 τ 来保证。这意味着所有诚实的参与方在超时发生时可以离开快速通道，而不会“陷入僵局”。
2. **步调同步（Pace-Sync）：**在快速通道超时后，所有参与方将调用 tcv-BA（两个连续值的拜占庭协议），并由于 tcv-BA 的终止性，获得一个同步步伐（syncPace）作为输出。此外，如果任何诚实的参与方在获得 syncPace 后意识到错过了一些快速通道区块，它可以在仅两个异步轮次内与 syncPace 同步，因为至少有 f+1 个诚实的参与方可以帮助它获取缺失的区块。
3. **悲观路径（Pessimistic Path）：**如果快速通道完全失败，没有输出任何内容，诚实的参与方将进入悲观路径。在这个阶段，协议必须输出预期大小为 O(B)的交易批次，并且确保所有交易（在所有诚实参与方的待处理队列顶部）以恒定的概率输出，从而即使在最坏的情况下也保证了活性。

总的来说，BDT 框架通过这些机制确保了即使在网络条件不稳定或存在恶意行为的情况下，系统也能够持续进展并最终达成共识。

3.3 复杂度分析

复杂度分析可以通过计算每个底层模块的复杂度来进行。

在乐观情况下，BDT 的快速通道（Fastlane）具有两种实例化形式：Bolt-sCAST 和 Bolt-sRBC，它们各自具有不同的通信复杂度和延迟特性。

Table 2: Per-block performance of different Bolt instantiations (which is also per-block cost of BDT in the good cases)

	Msg.	Comm.	Per-block latency	Two blocks latency	Bandwidth Cost	
					Leader	Others
Bolt-sCAST	$O(n)$	$O(nB)$	3 rounds	5 rounds	$O(nB)$	$O(B)$
Bolt-sRBC	$O(n^2)$	$O(nB)$	4 rounds	8 rounds	$O(B)$	$O(B)$

1. Bolt-sCAST：
- ◆ 每个区块的消息复杂度为线性 O(n)，其中 n 为节点数量。
- ◆ 领导者的每个区块带宽使用量为 O(nB)，也是线性的，B 为区块大小。
- ◆ 生成两个连续区块的延迟为 5 轮，其中一轮定义为从交易首次多播到所有诚实节点输出带有有效证明的区块的时间间隔。

2. Bolt-sRBC：
- ◆ 每个区块的消息复杂度为二次 O(n^2)。
- ◆ 每个节点的每个区块带宽使用量不超过区块大小 O(B)。
- ◆ 生成一个（待定）区块需要 4 轮，输出两个连续区块需要 8 轮。

在乐观情况下，即快速通道领导者总是诚实的，并且网络条件良好以至于快速通道不会超时，悲观阶段不会执行。

Table 3: Per-block performance of BDT in the worst cases

	Msg.	Comm.	Block latency (rounds)	Bandwidth Cost	
				Leader	Others
BDT-sCAST	$O(n^3)$	$O(nB)$	$3+1+T_{\text{tcv-BA}}+T_{\text{Dumbo}}$	$O(nB)$	$O(B)$
BDT-sRBC	$O(n^3)$	$O(nB)$	$4+1+T_{\text{tcv-BA}}+T_{\text{Dumbo}}$	$O(B)$	$O(B)$

* Note that the worst-case block latency reflects the case of turning off the level-1 fallback.

在最坏情况下（不考虑对手的影响），复杂度由以下几个方面构成：

- ◆ 在乐观的快速通道中，可能需要 O(τ)个异步轮次来退出临时的乐观执行，期间可能不产生任何有效区块。
- ◆ 快速通道停止后，所有参与方进入 Transformer 阶段，进行 tcv-BA，预期消息复杂度为 O(n^2)，通信复杂度为 O(λn^2)，每个参与方的带宽成本为 O(λn)。
- ◆ 如果 tcv-BA 的输出值大于零，可能需要调用 CallHelp 子程序，这将增加 O(n^2)的总体消息复杂度和 O(nB)的每区块通信复杂度，每个参与方平均需要 O(B)的带宽来获取每个区块。
- ◆ 在最坏情况下，如果快速通道完全失败，将执行 Dumbo 协议，平均消息成本为 O(n^3)，通信位数成本为 O(nB)，每区块每个参与方的带宽成本为 O(B)，假设批量大小 B 足够大。
- ◆ 即使在悲观路径中，区块生成的延迟也平均为 O(1)轮次。

这表明，在理想网络条件下，BDT 的快速通道能够以较低的通信和带宽成本实现高效的区块生成和确认。

总体而言，对于足够大的批量大小，BDT 的预期摊销成本为每个输出交易 O(n)位，其中 n 代表网络中的节点数量。每个输出区块的延迟预期为固定的几轮。

4 性能分析

论文中作者使用 Python 3 和相关的库以及安全参数实现了 BDT、Dumbo 和 HotStuff 协议，并通过一个单独的 Python 进程实现了一个通用的非阻塞网络层。他们采用了特定的加密技术来实现公共硬币和法定证明，并使用了 HoneyBadger BFT 的混合加密方法和 Reed-Solomon 纠删码库。此外，利用 EC2 实例的时钟来模拟本地时间，代替了安全模型中的对手控制时钟。实验在 Amazon EC2 的全球分布实例上进行，以评估 BDT 在现实广域网环境中的表现，并与 Dumbo 和 HotStuff 进行比较。

4.1 基本延迟

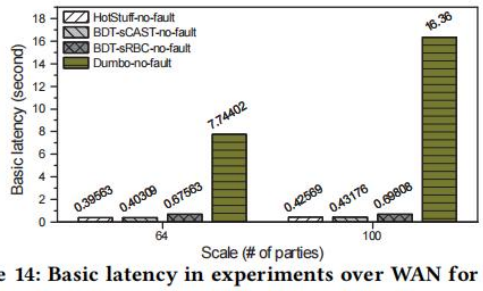


Figure 14: Basic latency in experiments over WAN for two-chain HotStuff, BDT-sCAST, BDT-sRBC and Dumbo.

通过有意设置参数以增加回退成本，研究者们测量了基本延迟，并发现 BDT 在处理低延迟场景时的性能与 HotStuff 相当，但在某些情况下比 Dumbo 显著更快。

4.2 峰值吞吐量

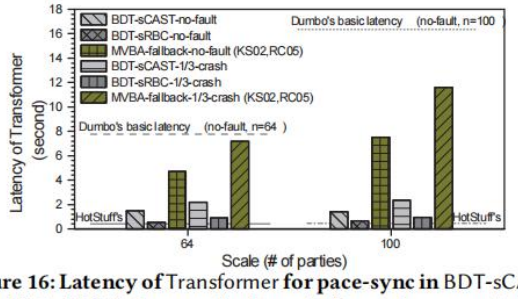


Figure 16: Latency of Transformer for pace-sync in BDT-sCAST and BDT-sRBC (when no fault and 1/3 crash, respectively). MVBA fallback in RC05 is also tested as a reference point.

作者通过不同的设置组合来衡量 Transformer 执行时间，这些设置包括：(i) 使用 BDT-sCAST 或 BDT-sRBC 配置；(ii) 开启或关闭 1/3 节点故障；(iii) 在 64 个或 100 个 EC2 实例上运行。此外，还测量了 MVBA 步调同步的延迟，MVBA 步调同步是使用 Backup/Abstract 原语的实例化，用于结合快速通道和 Dumbo。作为基本的参考点，比较结果表明，与 MVBA 的高成本相比，Transformer 的开销要小得多。这表明 Transformer 在不同网络条件下提供了更高效的步调同步机制。

4.3 延迟和吞吐量变化规律

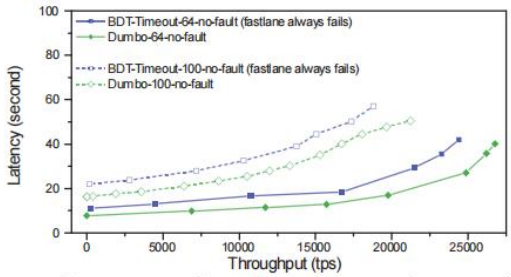


Figure 17: Latency v.s. throughput for experiments of BDT with idling fastlane (i.e., fastlane just timeouts after 2.5 sec).

作者还测量了 BDT-Timeout 的延迟与吞吐量权衡，目的是观察当 BDT 的快速通道遭受拒绝服务攻击时，BDT 的性能相比 Dumbo 如何下降。这可以视为对 BDT 最不利的测试场景，因为在与 Dumbo 相比的情况下，BDT 由于超时而总是额外花费 2.5 秒，然后才执行 Transformer 子协议。尽管如此，BDT 的性能仍然与 Dumbo 相近。具体来说，为了达到相同的吞吐量，BDT 只比 Dumbo 多花费了几秒钟（这主要是由于我们保守设置的 2.5 秒超时参数所导致的）。这表明即使在最坏的情况下，BDT 也能保持相对高效的性能。

4.4 延迟和吞吐量的平衡

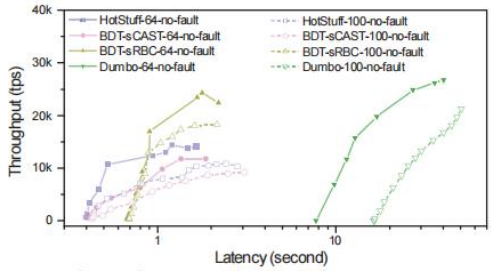


Figure 18: Throughput v.s. latency for experiments over WAN when n = 64 and 100, respectively (in case of periodically running pace-sync in BDT per only 50 fastlane blocks).

上图展示了在广域网环境中，当参与方数量为 64 和 100 时，BDT-sCAST、BDT-sRBC、HotStuff 和 Dumbo 之间的延迟与吞吐量的权衡。结果表明，BDT 在不同系统负载下都具有接近 HotStuff 的低延迟。无论是 BDT-sCAST 还是 BDT-sRBC，在所有情况下都比 Dumbo 快了几个数量级。两种 BDT 实例针对不同场景有不同的优势：BDT-sCAST 的延迟-吞吐量权衡与 2 链 HotStuff 相似，由于故意在每 50 个快速通道块后触发超时，它们的延迟变化很小。而 BDT-sRBC 的延迟-吞吐量趋势与 HotStuff 和 BDT-sCAST 明显不同，在固定较大的吞吐量时，BDT-sRBC 的延迟小于 BDT-sCAST；在固定较小的吞吐量时，BDT-sRBC 可能会慢一些。这明确区分了它们的应用场景：

BDT-sRBC 更适合于偏好大吞吐量的情况，而 BDT-sCAST 更适合于对延迟敏感的场景。

基于上述四项性能的测评,我们可以看出BDT在WAN环境中：

1. 在最佳情况下（即没有故障的同步网络），BDT 的速度与 2-chain HotStuff 一样快；
2. 在最坏情况下（即快速通道总是完全失败），BDT 与底层的异步悲观路径一样稳定。

这表明 BDT 能够在不同的网络条件下保持高效的共识机制，无论是在网络状况良好时的高效率，还是在面对网络攻击或故障时的强鲁棒性。

5 总结

通过对以上在区块链方面的前沿论文进行研读，我发现需要针对特定的功能需求来实现在网络中的安全协议。比如，在网络安全通信的 SSL/TLS 协议中，通过密钥协商来保证消息的机密性；在区块链这样的分布式系统中，通过共识协议来防止双重支付等。

本篇论文是考虑如何在由于网络状况问题导致的异步环境下，在区块链上达到安全且高效的共识。目前的研究主要有两类协议：同步和异步。它提出的 BDT 框架可以将同步协议，在不影响性能和功能的前提下，配置到部分异步协议上。这样就可以在不同的网络环境下选择适合的协议来实现 ABC。通过在不同网络环境下的实验，BDT 证明了其在实际部署中的安全性和有效性，增强了社区对区块链系统安全性的信心。

参 考 文 献

[1] Lu Y., Lu Z., Tang Q. Bolt-Dumbo Transformer: Asynchronous Consensus As Fast As the Pipelined BFT. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS ’22), November 7 – 11, 2022, Los Angeles, CA, USA. ACM, New York, NY, USA, 15 pages.

[2] Yuan Lu, Zhenliang Lu, and Qiang Tang. 2021. Bolt-dumbo transformer: Asynchronous consensus as fast as the pipelined bft. arXiv preprint arXiv:2103.09425