Modeling the Performance-Security Trade-off of Gasper's Block Proposal Mechanism Under Latency-Driven Attacks

Shuhan Qi, Qinglin Zhao, *Senior Member, IEEE*, Zijie Liu, MengChu Zhou, *Fellow, IEEE*, Meng Shen, *Member, IEEE*, Peiyun Zhang, *Senior Member,* and Yi Sun

Appendix A: Proofs of all lemmas and theorems.

Proof of Lemma 1: Consider a Δ-synchronous network where $\Delta \leq 2\delta < 2\Delta$. Note that Proposer i broadcasts a block B_i at the beginning of slot i. We now prove Lemma 1(a)-(c):

a) Since $\delta < \Delta \leq 2\delta$, a newly generated block will surely be received within 2 slots, but not necessarily received in one slot. Therefore, in slot i, proposer i surely receives B_{i-2} and all blocks preceding it but may not receive B_{i-1}. This confirms Lemma 1(a).

b) When proposer i-1 proposes B_{i-1} in slot i-1. There are two cases:

i) B_{i-2} is on the canonical chain, indicating that B_{i-2} is the B_H^- in slot i-1.

- If proposer i-1 did not receive B_{i-2} in slot i-1, it missed B_H^-, placing B_{i-1} off the canonical chain.
- If proposer i-1 received B_{i-2} in slot i-1, it received B_H^-, placing B_{i-1} on the canonical chain.

ii) B_{i-2} is off the canonical chain, indicating that B_H^- is another block among blocks from B_0 to B_{i-3}.

- According to Lemma 1(a), in slot i-1, proposer i-1 has received all blocks from B_0 to B_{i-3}, it indicates that proposer i-1 has surely received B_H^- and places B_{i-1} on the canonical chain.

Therefore, B_{i-2} and B_{i-1} cannot both be absent from the canonical chain simultaneously. However, it is possible that either of them or both will be on the global canonical chain. This confirms Lemma 1(b).

c) When proposing a block at slot i, Proposer i follows the fork choice rules and points B_i to the canonical chain in its own view. With Lemma 1(a), it is straightforward for proposer i to determine the canonical chain from B_0 to B_{i-2}. However, how to link B_i depends on whether $B_{i-1}$is the B_H^- and whether proposer i receives B_{i-1}.

When proposer i receives block B_{i-1}, we know that:

- if B_{i-1} is B_H^-, then B_i points to B_{i-1} according to Rule 1;
- otherwise, B_{i-2} is B_H^- according to (b) and hence B_i points to B_{i-2}.

When proposer i does not receive the B_{i-1}, we know that:

- if B_{i-1} is B_H^- and $B_{i-2}$is on the canonical chain, then B_i points to B_{i-2};
- if B_{i-3} is B_H^- according to (b) and hence B_i points to B_{i-3};
- otherwise, B_{i-2} is B_H^- according to (b) and hence B_i points to B_{i-2}.

Thus, B_i points to either block B_{i-1}, B_{i-2} or B_{i-3}, conclusively proving Lemma 1(c).

■

Proof of Theorem 1: Consider a Δ-synchronous network where $\Delta \leqslant 2\delta < 2\Delta$. The state space of this Markov Chain is $S \in \{S_1, S_2\}$ where $\{S_1 = (1,1)$ and $S_2 = (2,0)\}$. Let π_i, i=1,2, denote the steady probability of state S_i, respectively. According to the definition of the state, π_0 (π_1) denotes the probability that the newly generated block in current slot is on (off) the canonical chain. Assume $\Delta = 1$ is a unit time.

i) **Steady-state probabilities ($\boldsymbol{\pi_i}$):** We obtain $\boldsymbol{\pi_i}$ by solving the equation:

$$\begin{cases} \pi_0 = p_1\gamma_h \cdot \pi_0 + \pi_1 \\ \pi_1 = (q_1 + p_1 \cdot \gamma_a)\pi_0 \\ \pi_0 + \pi_1 = 1 \end{cases}$$

Thus, we can derive that:

$$\begin{cases} \pi_0 = \dfrac{1}{2 - p_1\gamma_h} \\ \pi_1 = \dfrac{1 - p_1\gamma_h}{2 - p_1\gamma_h} \end{cases}$$

When δ=1, each proposer can surely see the previously generated block since $\Delta = 1$.

ii) **Throughput (Γ):** Since $\pi_0$denotes the probability that the newly generated block in current slot is on the canonical chain, the expected number of blocks on the canonical chain per slot is $\pi_0 \times 1 = \pi_0$. In one time unit (i.e., $\Delta = 1$), there are $\frac{1}{\delta}$ slots and hence the throughput is given by

$$\Gamma = \frac{\pi_0}{\delta} = \begin{cases} \dfrac{1}{2 - p_1\gamma_h}, & \delta \in [\frac{1}{2}, 1) \\ 1, & \delta = 1 \end{cases}$$

iii) **Efficiency (η):** According to the definition of efficiency, we have:

$$\eta = \frac{\frac{\pi_0}{\delta}}{\frac{1}{\delta}} = \begin{cases} \dfrac{1}{2 - p_1\gamma_h}, & \delta \in \left[\frac{1}{2}, 1\right) \\ 1, & \delta = 1 \end{cases}$$

iv) **Fork probability (P_f):** Since $\pi_0 = \frac{1}{2-p_1\gamma_h}$, and there are δ blocks per unit time, the probability of $1/\delta$ blocks being all on the canonical chain is ${\pi_0}^{1/\delta}$. Therefore, the fork probability is:

$$P_{\mathrm{F}} = \begin{cases} 1 - \dfrac{1}{(2-p_1\gamma_h)^{\frac{1}{\delta}}}, \delta \in \left[\dfrac{1}{2}, 1\right) \\ 0 \qquad\qquad\qquad , \delta = 1 \end{cases}$$

■

Proof of Lemma 2: Consider a Δ-synchronous network where $\Delta < 3\delta < 3/2\Delta$. Proposer i broadcasts a block B_i at the beginning of slot i. We now prove Lemma 2(a)-(c):

a) Since $\delta<\Delta\leqslant 3\delta$, a newly generated block will surely be received within 3 slots, but not necessarily received in one slot. Therefore, in slot i, proposer i surely receives B_{i-3} and all blocks preceding it but may not receive B_{i-2} and B_{i-1}. This confirms Lemma 1(a).

b) When proposer i-1 proposes B_{i-1} in slot i-1. There are four cases:

i) B_{i-2} and $B_{i-3}$are on the canonical chain, indicating that B_{i-2} is the B_H^- in slot i-1.

- If proposer i-1 did not receive B_{i-2} in slot i-1, it missed B_H^-, placing B_{i-1} off the canonical chain.
- If proposer i-1 received B_{i-2} in slot i-1, it received B_H^-, placing B_{i-1} on the canonical chain.

ii) B_{i-3} is off the canonical chain, B_{i-2} is on the canonical chain, indicating that B_{i-2} is the B_H^- in slot i-1. Same as case i), if proposer i-1 received B_{i-2} in slot i-1, B_{i-1} is on the canonical chain, otherwise it will be off the canonical chain.

iii) B_{i-3} is on the canonical chain, B_{i-2} is off the canonical chain, indicating that B_{i-3} is the B_H^- in slot i-1. If proposer i-1 received B_{i-3} in slot i-1, B_{i-1} is on the canonical chain, otherwise it will be off the canonical chain.

iv) B_{i-2} and B_{i-3} are off the canonical chain, B_H^- is another block among blocks from B_0 to B_{i-4}. According to Lemma 2(a), in slot i-1, proposer i-1 has received all blocks from B_0 to B_{i-3}, it indicates that proposer i-1 has surely received B_H^- and places B_{i-1} on the canonical chain.

Therefore, B_{i-3}, B_{i-2}, and B_{i-1} cannot be all absent from the canonical chain. However, it is possible that each or multiple of them are on the canonical chain, thus proving Lemma 2(b).

c) When proposing a block at slot i, Proposer i follows the fork choice rules and points B_i to the canonical chain in its own view. The features influence a proposer's choice are 1) The blocks it received. 2.) the blocks are on the canonical chain or not. Lemma 2(a) and (b) show that a proposer i may not receive B_{i-2} or B_{i-1}, and there are at most 2 continuously blocks off the canonical simultaneously. As analyzed in Proof of Lemma 1(c), a proposer i might fail to receive B_{i-2} and B_{i-1}, resulting in B_i 's pointing to anyone of B_{i-1}, B_{i-2}, or B_{i-3}. Because of the 3-synchronous network, the connection options in case 2 has two more possibilities:

- If proposer i does not receive block B_{i-2} and B_{i-1}, B_{i-4} is on the canonical chain but B_{i-3} is not, B_{i-4} is B_H^-. Proposer i receives B_{i-4} and point B_i to it.
- If B_{i-2}, B_{i-1} are not received, B_{i-3}, B_{i-4} are not on the canonical chain, B_{i-5} is the B_H^-, according to Lemma 2(b). Proposer i receives B_{i-5} and point B_i to it. Also, B_{i-5} the earliest block that B_i may point to.

Thus, B_i can only point to B_{i-5}, B_{i-4}, B_{i-3}, B_{i-2}, or B_{i-1}, proving Lemma 2(c).

■

Proof of Theorem 2: Consider a Δ-synchronous network where $\Delta<3\delta<3/2\Delta$. The state space of this Markov Chain is $S \in \{S_0, S_1, \ldots \ldots S_7\}$ where $\{\ S_0 = (1,0,1),\ S_1 = (1,1,1),\ S_2 = (2,2,0),\ S_3 = (1,2,1), S_4 = (2,1,0), S_5 = (3,3,0),\ S_6 = (1,1,0), S_7 = (2,2,0)\ *\}$. Let π_i, $i = 1, \ldots, 7$, denote the steady probability of state S_i, respectively. Let $\pi = (\pi_0, \ldots, \pi_7)$, according to the definition of the state, π_0, π_1 and π_3 ($\pi_{2,4,5,6\ and\ 7}$) denotes the probability that the newly generated block in current slot is on (off) the canonical chain. Assume $\Delta = 1$ is a unit time. We have the following one-step transition probability matrix:

$$\mathrm{P}=\begin{bmatrix} 0 & \mathrm{P}_{1,0} & \mathrm{P}_{2,0} & 0 & 0 & 0 & 0 & 0 \\ \mathrm{P}_{0,1} & 0 & \mathrm{P}_{2,1} & 0 & \mathrm{P}_{4,1} & 0 & 0 & 0 \\ \mathrm{P}_{0,2} & 0 & 0 & \mathrm{P}_{3,2} & 0 & \mathrm{P}_{5,2} & 0 & 0 \\ 0 & \mathrm{P}_{1,3} & 0 & 0 & 0 & 0 & \mathrm{P}_{6,3} & 0 \\ \mathrm{P}_{0,4} & 0 & 0 & 0 & 0 & 0 & 0 & \mathrm{P}_{7,4} \\ \mathrm{P}_{0,5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathrm{P}_{0,6} & \mathrm{P}_{1,6} & \mathrm{P}_{2,6} & 0 & 0 & 0 & 0 & 0 \\ \mathrm{P}_{0,7} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Steady-state probabilities: π_i is obtained by solving the equation $\begin{cases} \pi=\pi P, \\ \sum_{i=0}^{m} \pi_i = 1 \end{cases}$.

i) Throughput (Γ): Since π_0, π_1 and π_3 denote the probability that the newly generated block in current slot is on the canonical chain, the expected number of blocks on the canonical chain per slot is $(\pi_0+\pi_1+\pi_3)\times 1=\pi_0+\pi_1+\pi_3$. In one time unit (i.e., $\Delta = 1$), there are $\frac{1}{\delta}$ slots and hence the throughput is:

$$\Gamma = \frac{\pi_0 + \pi_1 + \pi_3}{\delta}, \quad \delta \in [\frac{1}{3}\Delta, \frac{1}{2}\Delta)$$

ii) Efficiency (η): According to the definition of efficiency, we have:

$$\eta = \frac{\frac{\pi_0 + \pi_1 + \pi_3}{\delta}}{\frac{1}{\delta}} = \pi_0 + \pi_1 + \pi_3$$

iii) Fork probability P_F: Since there are $1/\delta$ blocks per unit time, the probability of $1/\delta$ blocks being all on the canonical chain is $(\pi_0 + \pi_1 + \pi_3)^{1/\delta}$. Therefore, the fork probability is:

$$P_F = 1 - (\pi_0 + \pi_1 + \pi_3)^{\frac{1}{\delta}}$$

■

Appendix B: Adversarial Behavior and its Impact

This appendix provides a formal description of the behavioral divergence between an honest and an adversarial proposer, specifically within the context of Case 1.

As shown in Fig. 1, we provide a crucial visual comparison between the actions of an honest proposer and an adversarial one while facing relations R1 to R4.

Honest Proposer Behavior

As depicted in the upper panel of Fig. 1, a protocol-compliant (honest) proposer follows a simple, deterministic rule. Upon entering slot i and having identified B_{i-1} as the unambiguous head of the canonical chain, it builds and broadcasts its own block, B_i, with B_{i-1} as its parent. This action correctly place B_i on the canonical chain and extends the canonical chain.

Adversarial Proposer Behavior

In contrast, an adversarial proposer, controlled by the adversary , intentionally deviates from this protocol to initiate a fork. As illustrated in the lower panel of Fig. 1, despite having also received and validated B_{i-1}, the adversary deliberately ignores this block and connect to an old one, producing a fork branch.

We observed that the adversary's strategy is constrained to attacking only those states where a fork is plausible, specifically by creating a block that is consistent with the differing view of a node experiencing network latency. In this same way, we derived conditions in Case 2 and have a conclusion that attack can be applied on S_0, S_1, S_2, S_3, S_4, and S_6 adaptively.

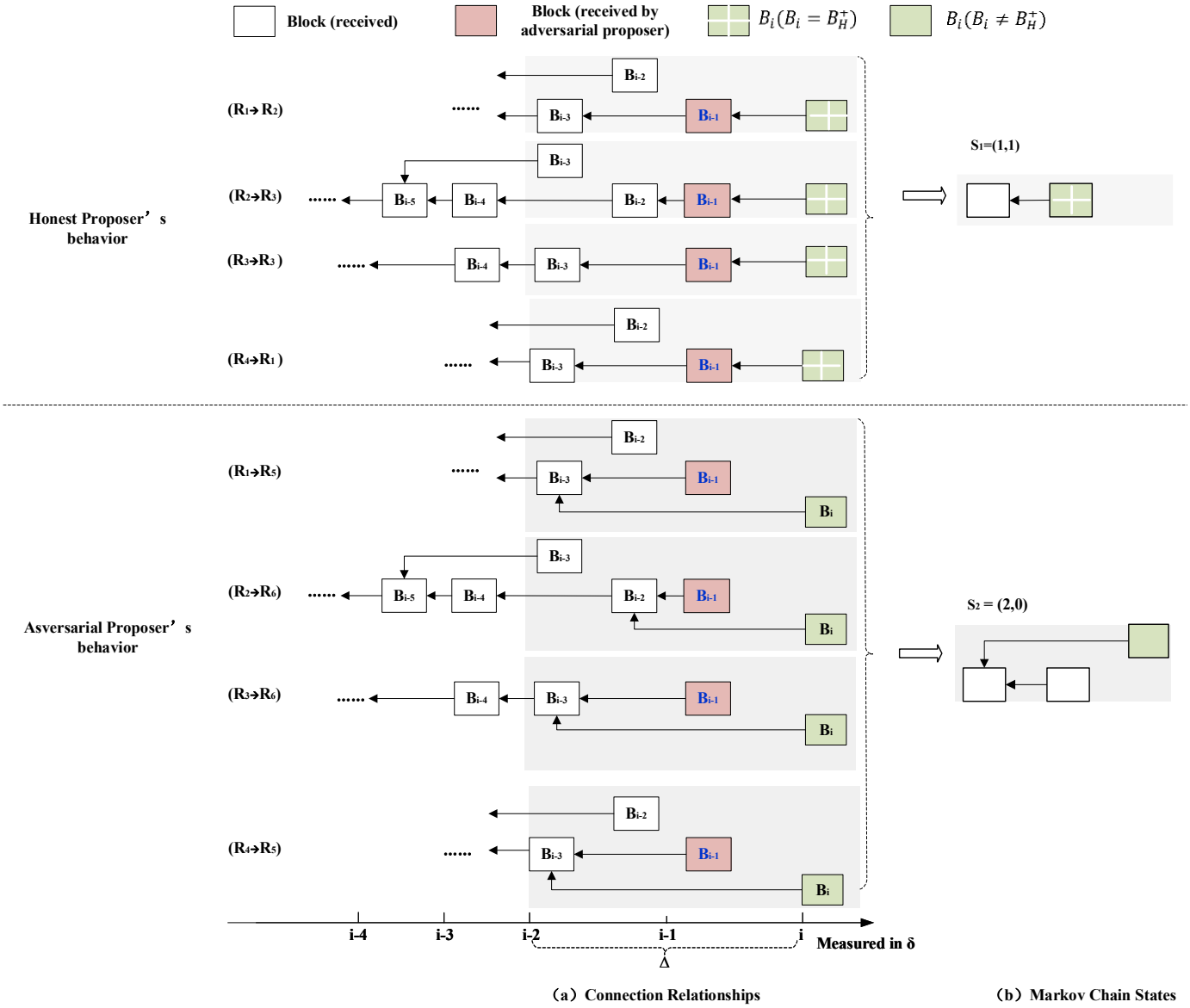

Fig. 1 Connection relationships (Adversarial proposer vs. honest validators)

Appendix C: Clarification of special states (2,2,0)* and (3,3,0)

In case 2, special states like $S_7 \triangleq (2,2,0)^*$ and $S_5 \triangleq (3,3,0)$ occurs. With the state space definition in response to e-ii), let's go through to explain S_7, and S_5 and justify the meaning of the two special states.

The meaning of S_7. The primary justification for defining S_7 as a separate state is that (2,2,0) is not topologically unique. Specifically, as shown in Fig. 2 when a new block B_i is proposed, its parent $\hat{B}_i$ could be either B_{i-2} or B_{i-3}. Both configurations result in the same high-level counts (M=2, L=2, N=0), but they represent distinct chain structures with different future evolutions. As illustrated in Fig. 3, they will transition to different states in the next time step. To ensure our model is valid, we treat these two configurations as distinct states. Define $S_2 \triangleq (2,2,0)$ for the first structure and $S_7 \triangleq (2,2,0)^*$ for the second. The asterisk (*) is the crucial notation used to distinguish S_7 as a unique state, ensuring the model accurately captures all possible transitions.

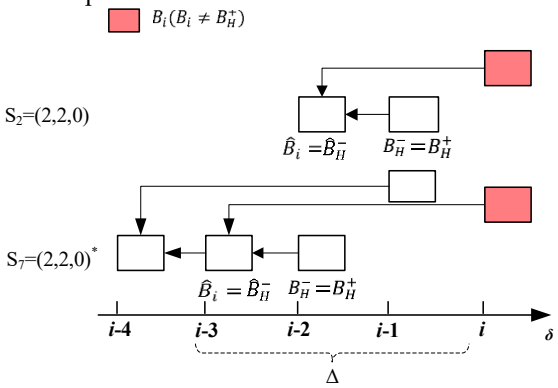

Fig. 2: States S_2 and S_7 from Fig. 8 of the manuscript

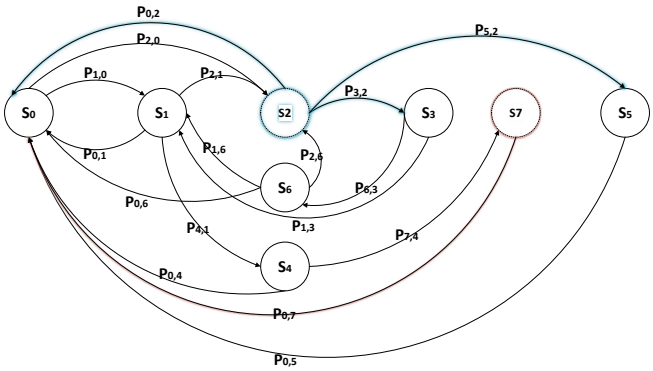

Fig. 3: Markov chain state transition for Case 2: $\delta \in [\frac{1}{3}\Delta, \frac{1}{2}\Delta)$.

The meaning of S_5 (3,3,0). The state $S_5 \triangleq (3,3,0)$ emerges as the critical state that captures the maximum achievable continuous forks under Case 2's specific slot setting where $\delta \in [\frac{1}{3}\Delta, \frac{1}{2}\Delta]$. While Lemma 2(a) guarantees that only blocks prior to B_{i-3} are visible to all proposers, B_{i-2} may be invisible to proper $i-1$, while both B_{i-2} and B_{i-1} may be invisible to proposer i due to network delays. This causes B_{i-2}, B_{i-1}, and B_i to build simultaneously on B_{i-3}, creating the precise topology illustrated in Fig. 4. In this state, both the parent of the new block ($\hat{B}_i$) and the parent of the canonical head ($\hat{B}_H^-$) are identical: $\hat{B}_i = \hat{B}_H^- =$ B_{i-3}. This block has three children, yielding $M = 3$ and $L = 3$ according to the state definition in case 2. Further, the new block B_i becomes one of these children but fails to become the new canonical head, resulting in $N = 0$.

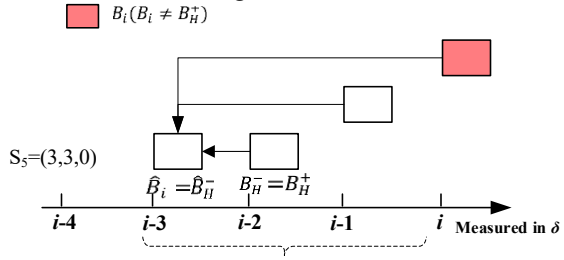

Fig. 4: States S_5 from Fig. 8 of the manuscript

Appendix D: Impact of Bursty Traffic on Throughput

This appendix evaluates the robustness of our analytical model against a more complex and realistic network delay profile. The main paper assumes a stationary truncated exponential distribution for block propagation delays. Here, we test the model's predictive power in a non-stationary environment characterized by intermittent, high-latency bursts, simulating periods of network congestion. The objective is to demonstrate that our analytical framework captures the fundamental system dynamics even when its core assumptions are stressed.

The Bursty Network Model

The goal of this experiment is to test the robustness of our theoretical model under a realistic network condition, which is characterized by intermittent congestion. To achieve this, we model the network's condition using a 2-state Markov model, as shown in Fig. 5, that transitions between two distinct traffic patterns.

- In the 'Normal' traffic state, network traffic exhibits message delays that are independently and identically distributed according to a truncated exponential distribution.
- In the 'Bursty' traffic state, each message delay is set to a fixed, high value to simulate sudden latency spikes caused by network congestion or traffic bursts.

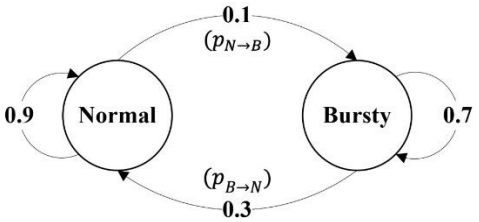

Fig. 5 Markov chain of bursty traffic model.

The transitions between these states are governed by the following probabilities:

- **Burst Probability ($p_{N\to B}$):** The probability of transitioning from 'Normal' to 'Bursty' in the next slot.
- **Recovery Probability ($p_{B\to N}$):** The probability of transitioning from 'Bursty' back to 'Normal'.

This two-state model allows us to simulate a network that experiences unpredictable periods of high latency, followed by recovery.

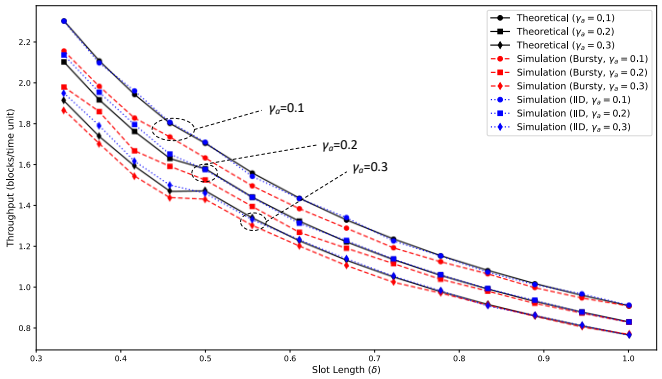

Fig. 6: Sensitivity analysis of throughput under bursty conditions (μ=0.2) and the presence of malicious validators.

Experimental Setup and Sensitivity Analysis

We conducted a sensitivity analysis to understand how system performance degrades under bursty traffic conditions and the presence of malicious validators. Fig. 6 presents the results for a specific scenario with the following parameters:

- Mean delay during 'Normal' traffic: μ = 0.2
- Meay delay when 'Normal' and 'Bursty' traffic coexist: μ = 0.4
- Burst Probability from 'Normal' to 'Bursty' traffic: $p_{N\to B}$ = 0.1
- Recovery Probability from 'Bursty' to 'Normal' traffic: $p_{B\to N}$ = 0.3
- Proportion of malicious validators: γ_a varying across [0.1, 0.2, 0.3], where we assume the PBFT security threshold that guarantees safety with fewer than 1/3 malicious nodes [39].

Results and Interpretation

Our analysis of Fig. 6 yields three key insights:

Our theoretical model factors in adversarial node with proportion γ_a and assumes i.i.d. network delays but does not model bursty traffic. To test the model's robustness under bursty traffic, Fig. 6 contrasts the analytical/simulation results when bursty traffic does not occur and the simulation results when bursty traffic occurs. From this figure, we have the following observations:

Observation 1: Model Robustness (Slight Deviation under Bursty Traffic). As our theoretical model does not account for bursty traffic, its predictions show a slight deviation when compared to simulation results under bursty conditions. However, the small magnitude of this deviation demonstrates our model's robustness under normal network conditions. This performance degradation is caused by the high latency of block propagation during traffic bursts, which increases the probability of forks and thereby slows the growth of the canonical chain. Consequently, for any given adversarial probability (γ_a), the measured throughput under bursty scenarios is slightly lower than what our theoretical model predicts.

Observation 2: Maximal Impact in Stable Environments. The impact of bursty traffic is most severe at the lowest adversarial ratio (i.e., when $\gamma_a = 0.1$), causing the most significant performance drop. The reason is that at this low adversarial ratio, the system is in a relatively healthy state where forks are naturally infrequent. In this stable environment, the introduction of bursty traffic is highly disruptive, causing a large decrease in performance as it is sufficient to push the system from a high-throughput regime into a more contentious, fork-prone state.

Observation 3: Diminished Impact in Hostile Environments. As the adversarial ratio increases, the performance gap between the bursty traffic simulation and the theoretical model narrows. This is because the impact of bursty traffic is less pronounced in a system already degraded by a high frequency of adversary-induced forks (i.e., when $\gamma_a = 0.2$ or $\gamma_a = 0.3$). As malicious actions become the dominant factor causing instability, the relative effect of network stochasticity diminishes, causing the two performance curves to converge.