Interplay between (im)perfectness, synchrony and connectivity: Case of Probabilistic Reliable Communication

Abhinav Mehta, Shashank Agrawal, Kannan Srinathan
Center for Security, Theory and Algorithmic Research (C-STAR),
International Institute of Information Technology, Hyderabad, 500032, India.
{abhinav_mehta@research.,sagrawal@research.,srinathan@}iiit.ac.in

ABSTRACT. For unconditionally reliable message transmission (URMT) in synchronous directed networks of n nodes, a subset of which may be Byzantine faulty, it is well-known that the minimum connectivity requirements for zero-error (perfect) protocols to exist is strictly higher than those where a negligible yet non-zero error probability is allowed (*Monte Carlo* protocols) [8]. In this work, we study the minimum connectivity requirements for the existence of (a) synchronous *Las Vegas* protocols, (b) asynchronous Monte Carlo protocols and (c) asynchronous Las Vegas protocols for URMT.

Interestingly, we prove that in any network, synchronous Las Vegas URMT protocols exist if and only if asynchronous Monte Carlo URMT protocols exist too. We further show another interesting result that asynchronous Las Vegas URMT protocols exist if and only if synchronous perfect protocols exist. In a nutshell, our results establish the following hierarchy with respect to the connectivity requirements for URMT protocols (in the following, P stands for perfect, MC stands for Monte Carlo, LV stands for Las Vegas, S denotes synchronous and A denotes asynchronous; it is known that SMC $<$ SP $=$ AP):

$$\text{SMC} < \text{SLV} = \text{AMC} < \text{ALV} = \text{SP} = \text{AP}$$

1 Introduction

In the unconditionally reliable message transmission (URMT) problem, two non-faulty players, the sender **S** and the receiver **R** are part of a communication network modelled as a digraph over n players/nodes influenced by an unbounded active adversary that may corrupt some subset of these n players/nodes. **S** has a message that he wishes to send to **R**; the challenge is to design a protocol such that **R** correctly obtains **S**'s message with arbitrarily small error probability, irrespective of what the adversary (maliciously) does to disrupt the protocol. Note that by "unconditional", we mean that the adversary is of unbounded computational power and therefore modern cryptographic tools for verifying the integrity of the data are irrelevant.

Analogous to randomized sequential algorithms, one may distinguish between two variants of URMT, namely, *Monte Carlo* and *Las Vegas*. In the former variant **R** outputs the sender's message with high probability and may produce an incorrect output with small probability; in the latter, **R** outputs the sender's message with high probability and with small probability may abort the protocol but in no case does the receiver terminates with an incorrect output. While Monte Carlo URMT has been studied in [7], we initiate the study of Las Vegas URMT over directed synchronous networks and characterize the exact gap in

the class of networks over which Las Vegas URMT, as compared to Monte Carlo URMT, is possible.

We also initiate the study of Monte Carlo URMT protocols over *asynchronous* directed networks. Unlike *synchronous* networks, in which the players have full information about the timings of the events in the network, in an asynchronous network, a conservative and more realistic assumption is used, namely that no time-bounds are known to the players regarding the schedule of various events in the network. Clearly, Monte Carlo URMT over asynchronous digraphs is harder to achieve (and indeed requires more network connectivity) than Monte Carlo URMT over synchronous digraphs. Equally evident is the fact that achieving Las Vegas URMT is harder (and again it indeed requires more network connectivity) than achieving Monte Carlo URMT, over synchronous digraphs. Though not seemingly related, interestingly, we prove that the additional requirements in network connectivity in both the aforementioned cases is exactly the same.

In the sequel, we similarly study the minimum connectivity requirements for the existence of asynchronous Las Vegas URMT protocols, which interestingly turns out to be the same as those for the existence of synchronous perfect protocols.

We further improve our insights in the problem by studying on how sparse can a digraph that permits URMT be. Specifically, we say that an edge is *critical* if its removal renders the graph insufficiently connected for URMT protocols (though before its removal the connectivity was sufficient). Ironically, it turns out that for perfect protocols, the number of critical edges is always $O(n)$ where as for the "easier" randomized protocols, we give a family of digraphs with $\Omega(n^2)$ critical edges! We remark that an earlier attempt in [7] to give such a family of digraphs for the case of synchronous Monte Carlo URMT protocols is incorrect and we correct the same; we also give similar families of digraphs (with $\Omega(n^2)$ critical edges) for synchronous Las Vegas (and asynchronous Monte Carlo) protocols.

1.1 Related Work

The problem of URMT was first introduced in [4]. It is proved that in an undirected network $2t + 1$ vertex-disjoint paths between **S** and **R** are necessary for URMT tolerating t-threshold adversary. Efficient protocols for URMT in directed networks abstracted as a collection of vertex-disjoint paths between **S** and **R** appear in [2]. In [8], the first characterization of general directed networks w.r.t Monte Carlo URMT tolerating non-threshold *mixed* adversary is given. A simplified characterization tolerating t-threshold Byzantine adversary follows in [7]. Above results have been derived in the case of synchronous networks.

A part of this work (SLV=AMC) is to appear as a Brief Announcement at DISC 2010.

2 Model and Definitions

We model the underlying network in which a sender **S** and a receiver **R** are two distinguished nodes as a directed graph $\mathcal{N} = (V, \mathcal{E})$, where V is the set of nodes and $\mathcal{E} \subseteq V \times V$ is the set of all directed edges in the network. We assume that all the edges are *secure*, i.e., if $(u, v) \in \mathcal{E}$ then u can send any message to v securely and reliably. We further assume that the topology of the network is known to every node.

Protocols running on such directed networks usually rely heavily on the information of timing of events in the system. We consider two extremes w.r.t timing model, i.e., either all the edges in the network are *synchronous* or *asynchronous*. In the former a protocol is executed in a sequence of *rounds* wherein each round, a player can send messages to his out-neighbors, receive the messages sent in that round by his in-neighbors and perform some computation on the received messages, in that order. During execution of a protocol, a subset of nodes in the network may be faulty, deviating arbitrarily from the designated protocol. We model such faults by a centralized non-threshold Byzantine adversary having unbounded computational power [5]. In the latter case (when the network is asynchronous), there is no fixed upper bound on the timings of events. To model asynchrony in the network we assume that the adversary is additionally equipped with the ability to schedule all the messages exchanged over the network while remaining oblivious to the messages being exchanged [1].

We represent the centralized non-threshold adversary by $\mathbb{A}$ which is the set of all possible "snapshots" of faults in the network. A single snapshot can be described as $B \subseteq V \backslash \{\mathbf{S},\mathbf{R}\}$ which means that nodes in the set B can be corrupted in Byzantine fashion. An adversary structure is a collection of all such B's. We consider the case of *adaptive* adversary who can corrupt atmost one element of the set $\mathbb{A}$ during an execution of a protocol. We call $\mathbb{A}$ a t-threshold adversary if $\mathbb{A} = \{B \mid B \subseteq V \setminus \{\mathbf{S},\mathbf{R}\}$ and $|B| \leq t\}$. The adversary structure is *monotone*: if $B_1 \in \mathbb{A}$ then $\forall\ B_2$ s.t. $B_2 \subseteq B_1$, $B_2 \in \mathbb{A}$. We note that $\mathbb{A}$ can be uniquely represented by listing the elements in its *maximal basis* $\overline{\mathbb{A}} = \{B \mid B \in \mathbb{A}, \nexists X \in \mathbb{A}$ s.t. $B \subset X\}$. Abusing the standard notation, we assume that $\mathbb{A}$ itself is a maximal basis.

Let the message space be a large finite field $\langle \mathbb{F}, +, \cdot \rangle$. All the computations are done in this field. In this paper we refer to *Las Vegas* URMT as URMT$_{LV}$ and *Monte Carlo* URMT as URMT$_{MC}$. We may also use URMT without any subscript to refer to both the variants together. We now formally define what we mean by a protocol being URMT$_{LV}$ or URMT$_{MC}$ protocol. All the probabilities are taken over the choice of the message $\mathbf{S}$ intends to send, the random inputs of all honest players and the random inputs of the adversary.

Definition 1. *[$(\mathbb{A},\delta)$-URMT$_{MC}$] Let $\delta < \frac{1}{2}$. We say that a protocol for transmitting messages in an asynchronous network $\mathcal{N}$ from $\boldsymbol{S}$ to $\boldsymbol{R}$ is $(\mathbb{A},\delta)$-URMT$_{MC}$ if for all valid Byzantine corruptions of any $B \in \mathbb{A}$, the probability that $\boldsymbol{R}$ outputs $\boldsymbol{m}$ given that $\boldsymbol{S}$ has sent $\boldsymbol{m}$, is at least $(1-\delta)$. Otherwise $\boldsymbol{R}$ outputs $\boldsymbol{m}' \neq \boldsymbol{m}$ or does not terminate. If the network is synchronous, $\mathbf{R}$ must terminate with certainty.*

Definition 2. *[$(\mathbb{A},\delta)$-URMT$_{LV}$] Let $\delta < \frac{1}{2}$. We say that a protocol for transmitting messages in an asynchronous network $\mathcal{N}$ from $\boldsymbol{S}$ to $\boldsymbol{R}$ is $(\mathbb{A},\delta)$-URMT$_{LV}$ if for all valid Byzantine corruptions of any $B \in \mathbb{A}$, the probability that $\boldsymbol{R}$ outputs $\boldsymbol{m}$ given that $\boldsymbol{S}$ has sent $\boldsymbol{m}$, is at least $(1-\delta)$. Otherwise, $\boldsymbol{R}$ outputs a special symbol $\perp$ $(\notin \mathbb{F})$ or does not terminate. If the network is synchronous, $\mathbf{R}$ must terminate with certainty.*

We refer to $(\mathbb{A},\delta)$-URMT protocol as a (t,δ)-URMT protocol when $\mathbb{A}$ is a t-threshold adversary.

Definition 3. *[$\mathbb{A}$-PRMT] We say that a protocol for transmitting messages in a synchronous (or asynchronous) network $\mathcal{N}$ from $\boldsymbol{S}$ to $\boldsymbol{R}$ is $\mathbb{A}$-PRMT if for all valid Byzantine corruptions*

*of any $B \in \mathbb{A}$, **R** outputs **m** when **S** has sent **m**.*

DEFINITION 4. *[Strong path, Weak path] A sequence of vertices $v_1, v_2, v_3, ..., v_k$ is said to be a strong path (resp. weak path) from v_1 to v_k in the network $\mathcal{N} = (V, \mathcal{E})$ if for each $1 \leq i < k$, $(v_i, v_{i+1}) \in \mathcal{E}$ (resp. $(v_i, v_{i+1}) \in \mathcal{E}$ or $(v_{i+1}, v_i) \in \mathcal{E}$). Furthermore, we assume that there vacuously exists a strong path from a node to itself.*

DEFINITION 5. *[Blocked node, Head node] A node u along a weak path p is called a blocked node if its out-degree along p is 0. A node y along a weak path p is called a head node if it is an intermediate node with out-degree 2 or a terminal node with out-degree 1.*

DEFINITION 6. *[Critical edge] In a digraph G for which URMT protocol exists, an edge is said to be critical if the deletion of that edge renders URMT impossible.*

DEFINITION 7. *[Authentication function] Let $K_1, K_2, K_3 \in_R \mathbb{F} \times \mathbb{F} - \{0\} \times \mathbb{F}$ and $m \in \mathbb{F}$. Authentication function χ is defined as $\chi(m; K_1, K_2, K_3) = (m + K_1, (m + K_1) \cdot K_2 + K_3)$.*

We refer to K_1, K_2, K_3 as *keys*. Suppose a random triplet K_1, K_2, K_3 unknown to the adversary is established between two nodes u and v in a network $\mathcal{N}$. The authentication function has the following important properties: (a) Even if u sends $\chi(m; K_1, K_2, K_3)$ along a faulty path to v, adversary does not learn any *information* regarding m. (b) Node v will be able to detect any change in $\chi(m; K_1, K_2, K_3)$'s value except with an error probability of atmost $\frac{1}{|\mathbb{F}|}$. (Proofs for the same appear in [6]).

3 Characterizing synchronous networks for $(\mathbb{A}, \delta)$-URMT$_{LV}$

THEOREM 8. *In a directed synchronous network $\mathcal{N}$, $(\mathbb{A}, \delta)$-URMT$_{LV}$ protocol is possible if and only if for every adversary structure $B \subseteq \mathbb{A}$ such that $|B| = 2$, (B, δ)-URMT$_{LV}$ protocol is possible.*

PROOF. *Necessity*: Obvious. *Sufficiency:* Let f be the field element **S** intends to send to **R** and $\mathbb{A} = \{B_1, B_2, \ldots, B_N\}$. For $3 \leq |\mathcal{A}| \leq N$, we consider $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$, three $\lceil \frac{2|\mathcal{A}|}{3} \rceil$-sized subsets of $\mathcal{A}$ such that each element of $\mathcal{A}$ occurs in at least two distinct $\mathcal{A}_i$'s. We now detail the construction of protocol Γ which is an $(\mathcal{A}, 2\delta - \delta^2)$-URMT$_{LV}$ protocol (as proved in the following lemma): for each $\{i, j\} \subset \{1, 2, 3\}$, $(\{\mathcal{A}_i, \mathcal{A}_j\}, \delta)$-URMT$_{LV}$ protocols are run on f in parallel; **R** takes the majority of the outcomes of $(\{\mathcal{A}_i, \mathcal{A}_j\}, \delta)$-URMT$_{LV}$ protocols and outputs that as the message, in case there is no majority it outputs $\perp$. Repeating Γ sufficiently many times yields an $(\mathcal{A}, \delta)$-URMT$_{LV}$ protocol.

From induction it follows that $(\mathbb{A}, \delta)$-URMT$_{LV}$ exists. ∎

LEMMA 9. *For the directed synchronous network $\mathcal{N}$, the protocol Γ constructed above is an $(\mathcal{A}, 2\delta - \delta^2)$-URMT$_{LV}$ protocol.*

PROOF. Without loss of generality we assume that the set $\mathcal{A}_1$ is corrupt. As per protocol Γ, **R** takes the majority of the outcomes of the three sub-protocols. Hence, Γ *fails* (outputs $\perp$) only if at least one of $(\{\mathcal{A}_1, \mathcal{A}_3\}, \delta)$-URMT$_{LV}$ or $(\{\mathcal{A}_1, \mathcal{A}_2\}, \delta)$-URMT$_{LV}$ fails which happens with atmost $1 - (1 - \delta)^2$ probability. Hence, Γ is an $(\mathcal{A}, 2\delta - \delta^2)$-URMT$_{LV}$ protocol. ∎

Having reduced the problem of URMT$_{LV}$ in a synchronous network tolerating an adversary structure to the problem of URMT$_{LV}$ tolerating all its 2-sized subsets, we now proceed to characterize directed synchronous networks $\mathcal{N} = (V, \mathcal{E})$ in which URMT$_{LV}$ tolerating adversary structure $B = \{B_1, B_2\}$ is possible (where $B_1, B_2 \in \mathbb{A}$).

THEOREM 10. *In a directed synchronous network $\mathcal{N}$, (B, δ)-URMT$_{LV}$ protocol is possible if and only if for each $\alpha \in \{1, 2\}$, there exists a weak path q_α avoiding nodes in $B_1 \cup B_2$ such that every node u along the path q_α has a strong path to **R** avoiding all nodes in $B_{\overline{\alpha}}$[*] (Paths q_1, q_2 need not be distinct.)*

We prove the theorem in the following two sections.

3.1 Sufficiency

We state the following simple lemma without proof.

LEMMA 11. *Every weak path p from **S** to **R** can be represented as a strong path from **S** to **R** or as an alternating sequence of blocked nodes u_i's and head nodes y_i's, i.e, $\exists\ k > 0$ such that $u_1, y_1, u_2, y_2, \ldots, u_k, y_k$ occur along the path p in that order. (Here u_1 may be **S**, and y_k may be **R**).*

For a directed synchronous network $\mathcal{N}$, which satisfies the conditions given in Theorem 10, we show how to construct a protocol Π tolerating the adversary structure $B = \{B_1, B_2\}$. If either q_1 or q_2 is a strong path from **S** to **R**, **S** can trivially send m along that path. When this is not the case, q_1 and q_2 can be expressed as $u_1, y_1, u_2, y_2, \ldots, u_{n_1}, y_{n_1}$ and $u'_1, y'_1, u'_2, y'_2, \ldots, u'_{n_2}, y'_{n_2}$ respectively. We construct two sub-protocols Π_1 and Π_2. For each $i \in \{1, 2\}$, protocol Π_i uses the honest weak path q_i. We give a construction for Π_1, and the construction of Π_2 follows by symmetry. Π_1 proceeds in the following steps:

1. **S** sends m to u_1 along q_1. For $1 \leq k \leq n_1$, node y_k chooses 3^k random keys namely $K_{k,1}, K_{k,2}, \ldots, K_{k,3^k}$ and sends those to u_k and u_{k+1}. (Here u_{n_1+1} denotes **R**.)
2. Node u_1 receives m from **S** and keys $K_{1,1}, K_{1,2}, K_{1,3}$ from y_1. It calculates $(\psi_{1,1}, \phi_{1,1}) = \chi(m; K_{1,1}, K_{1,2}, K_{1,3})$ and sends it to **R** along a strong path avoiding B_2 in some fixed round r_{u_1}.
   For $1 < k \leq n_1$, u_k receives 3^{k-1} keys from y_{k-1} and 3^k keys from y_k. It authenticates the keys received from y_{k-1} with the keys received from y_k and sends it to **R** along a strong path avoiding B_2 in some fixed round r_{u_k}. Formally, u_k calculates, $\forall j\ 1 \leq j \leq 3^{k-1}$, $(\psi_{k,j}, \phi_{k,j}) = \chi(K_{k-1,j}; K_{k,3j-2}, K_{k,3j-1}, K_{k,3j})$.
3. **R** receives $\{K'_{n_1,1}, K'_{n_1,2}, \ldots, K'_{n_1,3^{n_1}}\}$ from y_{n_1}. $\mathcal{N}$ being a synchronous network, **R** knows exactly the round number, say r'_{u_k}, in which it will receive messages that u_k sent to it in round r_{u_k}. If **R** does not receive valid messages from u_k in round r_{u_k}, it assumes that B_1 is faulty. Else if it receives $\forall k\ \forall j\ 1 \leq k \leq n_1, 1 \leq j \leq 3^{k-1}$, $(\psi'_{k,j}, \phi'_{k,j})$, the protocol proceeds as follows.
   for k in n_1 to 2
   R verifies $\forall j$, $\phi'_{k,j} \stackrel{?}{=} \psi'_{k,j} \cdot K'_{k,3j-1} + K'_{k,3j}$. If the verification fails, **R** concludes that B_1 is faulty and stops. Otherwise, **R** recovers $K'_{k-1,j}$ as $\psi'_{k,j} - K'_{k,3j-2}$.

[*] We denote $\overline{1} = 2$ and vice-versa.

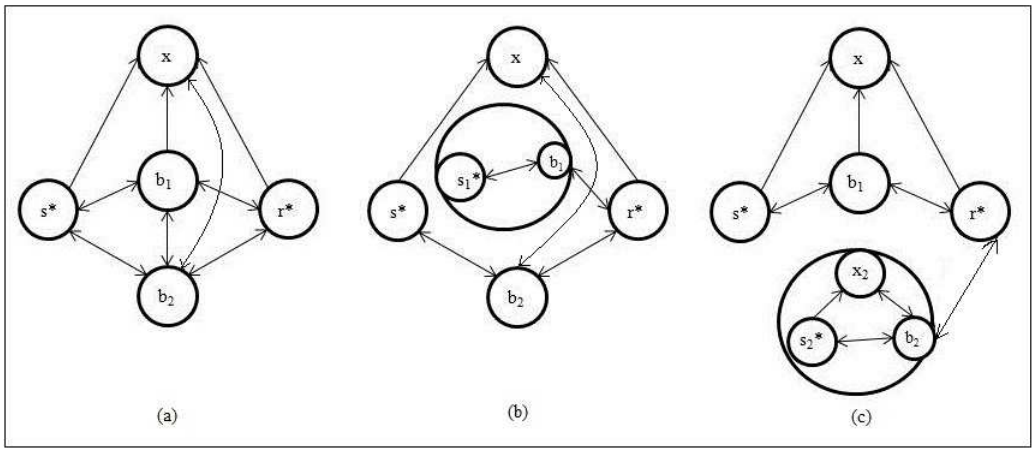

Figure 1: (a) The directed network $\mathcal{N}^*$ (b) Adversary strategy when b_1 is faulty (c) Adversary strategy when b_2 is faulty.

If at the end of the loop, **R** has recovered $K'_{1,1}, K'_{1,2}, K'_{1,3}$ then **R** verifies whether $\phi'_{1,1} \stackrel{?}{=} \psi'_{1,1} \cdot K'_{1,2} + K'_{1,3}$. If the verification passes, **R** recovers $m_1 = \psi'_{1,1} - K'_{1,1}$ as the message.

Now, Π_1 and Π_2 are run in parallel in the synchronous network $\mathcal{N}$. **R** takes one of the following actions based on the outcomes of these protocols: (a) If **R** detects that B_i is corrupt in Π_i, it outputs whatever message it recovered from $\Pi_{\bar{i}}$. (b) If **R** recovers messages from each of the Π_i's and both the messages are same, it outputs that message. (c) If messages recovered through Π_1 and Π_2 are different, it outputs $\perp$. This completes Π. In the following lemma we prove that this is a $(B, \frac{1}{|\mathbb{F}|})$-URMT$_{LV}$ protocol.

LEMMA 12. *Π, as constructed above, is a $(B, \frac{1}{|\mathbb{F}|})$-URMT$_{LV}$ protocol.*

PROOF. We analyse the protocol case wise: (a) For some i, **R** concludes through Π_i that B_i is faulty, and outputs whatever it recovers from $\Pi_{\bar{i}}$. For each i, none of the nodes in $B_{\bar{i}}$ participate in the protocol Π_i. Hence, if some verification fails during Π_i, B_i has to be faulty, and $\Pi_{\bar{i}}$ should recover the correct message m. (b) For each $i \in \{1, 2\}$, all verifications in Π_i pass. Case (i) $m_i = m_{\bar{i}}$, and **R** outputs m_i. Since one of m_i or $m_{\bar{i}}$ has to be same as m, **R**'s output is correct. Case (ii) $m_i \neq m_{\bar{i}}$. This implies that one of B_1 or B_2 was corrupt and managed to change the authenticated message m such that it wasn't detected at **R**. Since this happens with a probability $\leq \frac{1}{|\mathbb{F}|}$, **R** outputs $\perp$ with atmost $\frac{1}{|\mathbb{F}|}$ probability. ∎

3.2 Necessity

We show that if $\mathcal{N}$ does not satisfy the conditions of Theorem 10, $(\{B_1, B_2\}, \delta)$-URMT$_{LV}$ is impossible from **S** to **R**. Without loss of generality, let us assume that the two sets comprising the adversary structure are disjoint [†]. Let the path q_1 be not present between **S** and **R** in $\mathcal{N}$.

[†]If the two sets are not disjoint, adversary can always fail-stop nodes in $B_1 \cap B_2$ without revealing the identity of the corrupted set.

(The case where path q_2 is not present can be handled analogously.) Hence, every weak path between **S** and **R** avoiding $B_1 \cup B_2$ has at least one node w such that every strong path from w to **R** passes through B_2.

We first consider the simple network $\mathcal{N}^* = (V^*, \mathcal{E}^*)$ shown in Figure 1(a) consisting of five nodes s^*, r^*, b_1, b_2 and x where s^* is the sender and r^* is the receiver and show that $(\{\{b_1\}, \{b_2\}\}, \delta)$-URMT$_{LV}$ from s^* to r^* is impossible. Note that node x can send messages to node b_2 only. We then prove that the digraph $\mathcal{N}$ can be partitioned into disjoint sets whose connectivity properties are similar to the connectivity between nodes of digraph $\mathcal{N}^*$. Now, if URMT$_{LV}$ is possible in $\mathcal{N}$, it would also be possible in $\mathcal{N}^*$, which is a contradiction. This implies that the conditions mentioned in Theorem 10 are necessary.

LEMMA 13. *In the synchronous network $\mathcal{N}^*$, shown in Figure 1(a), $(\{\{b_1\}, \{b_2\}\}, \delta)$-URMT$_{LV}$ ($\delta < 1/2$) from s^* to r^* is impossible.*

PROOF. W.r.t. to an execution E_i, we define the following: (a) The vector $\vec{C}_i = (c^i_{s^*}, c^i_{r^*}, c^i_{b_1}, c^i_{b_2}, c^i_x)$ which denotes the coin tosses input to nodes, where c^i_n denotes the coin tosses of node n. (b) The view of a node n, $view_n(E_i)$, which comprises of the internal coin tosses c^i_n of node n and the messages it receives during execution E_i. We assume that a protocol Π^* exists in $\mathcal{N}^*$ which is a $(\{\{b_1\}, \{b_2\}\}, \delta)$-URMT$_{LV}$ protocol. Consider the following two executions of Π^*.

1. Execution E_1: s^* chooses to send m_1. r^* outputs m_1. Node b_1 is corrupt. Node b_1 simulates a local copy of s^*, say s^*_1, with message $m_2 \neq m_1$ and random coin tosses c. Node b_1 ignores all messages received from nodes s^*, x and b_2, neither sends any message to them. At the beginning of each round, b_1 receives messages from r^* and from the simulated s^*_1, does local computation and sends out messages to r and s^*_1. During the same round, s^*_1 receives messages from b_1, its state is updated and messages are sent out for the next round. (When s^* chooses to send m_1, r^* must halt with m_1 on majority of coin tosses, hence there exists such an execution E_1).
2. Execution E_2: s^* chooses to send $m_2 (\neq m_1)$. Coin tosses $\vec{C}_2$ of nodes are such that $c^2_{b_1} = c^1_{b_1}$, $c^2_{r^*} = c^1_{r^*}$ and $c^2_{s^*} = c$ (c are the coin tosses adversary gave as input to s^*_1 in E_1). Adversary corrupts node b_2 actively and does the following. It simulates a local copy of s^* and x, say s^*_2 and x_2, giving input m_1 and $c^1_{s^*}$ to s^*_2 and c^1_x to x_2. Node b_2 ignores all messages received from nodes in s^*, x and b_1, neither sends any messages to them. Note that the node x does not have any effect on the output of r^* (as it has no strong path to r^*). Now, b_2 handles simulation of nodes s^*_2 and x_2 locally in the same manner as b_1 handled simulation of s^*_1 in the execution E_1. Messages received by x from r^* in each round of E_1 are delivered to x_2 by the adversary in the corresponding rounds of E_2.

Above mentioned adversary strategy ensures that $view_{r^*}(E_1) = view_{r^*}(E_2)$. Hence r^* halts with output m_1 in execution E_2, violating the condition of Π^* being a URMT$_{LV}$ protocol. This leads us to a contradiction regarding the existence of Π^*. ∎

A pictorial view of adversary strategy is presented in Figure 1(b), (c).

LEMMA 14. *The set of nodes V in the network $\mathcal{N}$ can be partitioned into 5 disjoint sets $S^*, R^*, B'_1 \subseteq B_1, B_2$ and X' such that $\mathbf{S} \in S^*$, $\mathbf{R} \in R^*$ and $\forall\ 1 \leq i < j \leq 5$ an edge exists be-*

tween a node of $F[i]$ and a node of $F[j]$ only if $(f(i), f(j)) \in \mathcal{E}^*$ where $F = [S^*, R^*, B_1', B_2, X']$ and $f = [s^*, r^*, b_1, b_2, x]$ are two ordered lists.

PROOF. In the network $\mathcal{N}$, every weak path between **S** and **R** avoiding $B_1 \cup B_2$ has at least one node w such that every strong path from w to **R** passes through B_2.

We partition the non-faulty nodes $H = V \setminus \{B_1 \cup B_2\}$ into 3 disjoint sets. Let $R^* \subset H$ denote the set of all nodes that have a weak path to **R** (avoiding $B_1 \cup B_2$) such that every node w in the weak path has a strong path to **R** avoiding B_2. Divide the rest of non-faulty nodes in two disjoint sets S^* and X. Define $S^* = \{w \in H \setminus R^* | w$ has a strong path to **R** avoiding $B_2\}$. Define $X = H \setminus \{S^* \cup R^*\}$. Clearly, $\mathbf{R} \in R^*$ and $\mathbf{S} \in S^*$. Moreover, if any node $w \in X$ has a strong path to **R**, it passes through some node in B_2.

Also, divide the set B_1 into two disjoint sets. Define $B_1' = \{u \in B_1 | u$ has a strong path to **R** avoiding $B_2\}$. Let $B_1^X = B_1 \setminus B_1'$. Let us consider the two sets X and B_1^X together as a set X', i.e., $X' = X \cup B_1^X$.

It easily follows from the definitions above that $\nexists\ (u, v) \in \mathcal{E}$ such that $u \in X'$ and $v \in S^* \cup R^* \cup B_1'$, otherwise there would be a path from a node in X' to **R** avoiding B_2. Also, there cannot exist any directed edge between a node in S^* and a node in R^*. Observe that the only edges missing from N^* are $(x, s^*), (x, r^*), (x, b_1)$ and $(s^*, r^*), (r^*, s^*)$. Hence, proved. ∎

LEMMA 15. *In the directed synchronous network $\mathcal{N} = (V, \mathcal{E})$, $(\{B_1, B_2\}, \delta)$-URMT$_{LV}$ is possible from* **S** *to* **R** *only if $(\{\{b_1\}, \{b_2\}\}, \delta)$-URMT$_{LV}$ is possible from s^* to r^* in the network $\mathcal{N}^*$.*

PROOF. Proof is straightforward using standard player simulation technique. ∎

From Lemma 13 we know that $(\{\{b_1\}, \{b_2\}\}, \delta)$-URMT$_{LV}$ is impossible from s^* to r^* in the network $\mathcal{N}^*$. We arrive at a contradiction. Hence, the conditions mentioned in Theorem 10 are necessary.

4 Characterizing asynchronous networks for $(\mathbb{A}, \delta)$-URMT$_{MC}$

THEOREM 16. *In a directed asynchronous network $\mathcal{N}$, $(\mathbb{A}, \delta)$-URMT$_{MC}$ protocol is possible if and only if for every adversary structure $B \subseteq \mathbb{A}$ such that $|B| = 2$, (B, δ)-URMT$_{MC}$ protocol is possible.*

PROOF. *Necessity*: Obvious. *Sufficiency:* Let f be the field element **S** intends to send to **R** and $\mathbb{A} = \{B_1, B_2, \ldots, B_N\}$. For $3 \leq |\mathcal{A}| \leq N$, we consider $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$, three $\lceil \frac{2\mathcal{A}}{3} \rceil$-sized subsets of $\mathcal{A}$ such that each element of $\mathcal{A}$ occurs in at least two distinct $\mathcal{A}_i$'s. We now detail the construction of protocol Γ which is an $(\mathcal{A}, 2\delta - \delta^2)$-URMT$_{MC}$ protocol (proof is similar to that of Lemma 12): for each $\{i, j\} \subset \{1, 2, 3\}$, $(\{\mathcal{A}_i, \mathcal{A}_j\}, \delta)$-URMT$_{MC}$ protocols are run on f in parallel; **R** waits until two of the three $(\{\mathcal{A}_i, \mathcal{A}_j\}, \delta)$-URMT$_{MC}$ protocols halt with the same output and outputs that as the message. Repeating Γ sufficiently many times yields an $(\mathcal{A}, \delta)$-URMT$_{MC}$ protocol.

From induction it follows that $(\mathbb{A}, \delta)$-URMT$_{MC}$ exists. ∎

Having reduced the problem of URMT$_{MC}$ in an asynchronous network tolerating an adversary structure $\mathbb{A}$ to the problem of URMT$_{MC}$ tolerating all its 2-sized subsets, we now proceed to characterize directed asynchronous networks $\mathcal{N} = (V, \mathcal{E})$ in which URMT$_{MC}$ tolerating adversary structure $B = \{B_1, B_2\}$ is possible (where $B_1, B_2 \in \mathbb{A}$).

THEOREM 17. *In a directed asynchronous network $\mathcal{N}$, $(\{B_1, B_2\}, \delta)$-URMT$_{MC}$ protocol is possible if and only if for each $\alpha \in \{1, 2\}$, there exists a weak path q_α avoiding nodes in $B_1 \cup B_2$ such that every node u along the path q_α has a strong path to **R** avoiding all nodes in $B_{\overline{\alpha}}$. (Paths q_1, q_2 need not be distinct.)*

4.1 Sufficiency

For a directed asynchronous network $\mathcal{N}$, which satisfies the conditions given in Theorem 17, we show how to construct a protocol Π tolerating the adversary structure $B = \{B_1, B_2\}$. If either q_1 or q_2 is a strong path from **S** to **R**, **S** can trivially send m along that path and **R** is bound to receive it. When this is not the case, q_1 and q_2 can be expressed as $u_1, y_1, u_2, y_2, \ldots, u_{n_1}, y_{n_1}$ and $u'_1, y'_1, u'_2, y'_2, \ldots, u'_{n_2}, y'_{n_2}$ respectively. We construct two subprotocols Π_1 and Π_2. For each $i \in \{1, 2\}$, protocol Π_i uses the honest weak path q_i. We give a construction for Π_1, and the construction of Π_2 follows by symmetry. Π_1 proceeds in the following steps:

1. **S** sends m to u_1 along q_1. For $1 \leq k \leq n_1$, node y_k chooses 3^k random keys namely $K_{k,1}, K_{k,2}, \ldots, K_{k,3^k}$ and sends those to u_k and u_{k+1}. (Here u_{n_1+1} denotes **R**.)
2. Node u_1 waits for m to arrive from **S** and keys $K_{1,1}, K_{1,2}, K_{1,3}$ to arrive from y_1. It calculates $(\psi_{1,1}, \phi_{1,1}) = \chi(m; K_{1,1}, K_{1,2}, K_{1,3})$ and sends it to **R** along a strong path avoiding B_2.
   For $1 < k \leq n_1$, u_k waits for 3^{k-1} keys to arrive from y_{k-1} and 3^k keys to arrive from y_k. It authenticates the keys received from y_{k-1} with the keys received from y_k and sends it to **R** along a strong path avoiding B_2. Formally, u_k calculates $\forall j\ 1 \leq j \leq 3^{k-1}$ $(\psi_{k,j}, \phi_{k,j}) = \chi(K_{k-1,j}; K_{k,3j-2}, K_{k,3j-1}, K_{k,3j})$.
   As the communication between u_i's and y_i's occurs along the honest weak path q_1, every u_i receives the keys (or message) eventually.
3. **R** waits for $\{K'_{n_1,1}, K'_{n_1,2}, \ldots, K'_{n_1,3^{n_1}}\}$ to arrive from y_{n_1}. **R** runs the following loop:
   for k in n_1 to 2
   R waits until it receives $\forall j\ 1 \leq j \leq 3^{k-1}$ $(\psi'_{k,j}, \phi'_{k,j})$ from u_k [‡]. If **R** does receive, it verifies $\forall j$, $\phi'_{k,j} \stackrel{?}{=} \psi'_{k,j} \cdot K'_{k,3j-1} + K'_{k,3j}$. If the verification fails, **R** concludes that B_1 is faulty and stops. Otherwise, **R** recovers $K'_{k-1,j}$ as $\psi'_{k,j} - K'_{k,3j-2}$.
   If at the end of the loop, **R** has recovered $K'_{1,1}, K'_{1,2}, K'_{1,3}$ then **R** verifies whether $\phi'_{1,1} \stackrel{?}{=} \psi'_{1,1} \cdot K'_{1,2} + K'_{1,3}$. If the verification passes, **R** recovers $m_1 = \psi'_{1,1} - K'_{1,1}$ as the message.

Π_1 and Π_2 are run in parallel in the asynchronous network $\mathcal{N}$. **R** takes one of the following actions based on the outcomes of these protocols: (a) If for any $i \in \{1, 2\}$, Π_i concludes that B_i is faulty, **R** waits for $\Pi_{\overline{i}}$ to terminate, and outputs $m_{\overline{i}}$ as message. (b) If for

[‡]As these messages are delivered along faulty paths, they may never arrive. However, since Π_1 and Π_2 are run in parallel and **R** waits for only one of them to terminate, the protocol Π always terminates.

any $i \in \{1,2\}$, Π_i halts with m_i as message, **R** outputs that as message without waiting for the protocol $\Pi_{\bar{i}}$ to terminate. Above is a $(B, \frac{1}{|\mathbb{F}|})$-URMT$_{MC}$protocol as proved in the following lemma.

LEMMA 18. *Π, as constructed above, is a $(B, \frac{1}{|\mathbb{F}|})$-URMT$_{MC}$ protocol.*

PROOF. We analyze the protocol case wise: (a) For some i, **R** concludes through Π_i that B_i is faulty, and outputs whatever it recovers from $\Pi_{\bar{i}}$. For each i, none of the nodes in B_i participate in the protocol $\Pi_{\bar{i}}$. Hence, if some verification fails during Π_i, B_i has to be faulty, and $\Pi_{\bar{i}}$ is bound to terminate with $m_{\bar{i}} = m$. (b) For some i, Π_i terminates successfully with output m_i. Probability that $m_i \neq m$ is at most $\frac{1}{|\mathbb{F}|}$. Hence, **R**'s output is correct with probability at least $\frac{|\mathbb{F}|-1}{|\mathbb{F}|}$. ∎

4.2 Necessity

In a directed asynchronous network $\mathcal{N}$ which does not satisfy the conditions of Theorem 17, we show that $(\{B_1, B_2\}, \delta)$-URMT$_{MC}$ is impossible from **S** to **R**. Let the path q_1 be not present between **S** and **R** in $\mathcal{N}$. (The case where path q_2 is not present can be handled analogously.) We again consider the simple network $\mathcal{N}^* = (V^*, \mathcal{E}^*)$ shown in Figure 1(a). However, this time the links between nodes are asynchronous. We show that $(\{\{b_1\}, \{b_2\}\}, \delta)$-URMT$_{MC}$ from s^* to r^* is impossible in $\mathcal{N}^*$. We have already proved in Lemma 14 that the digraph $\mathcal{N}$ can be partitioned into disjoint sets whose connectivity properties are similar to the connectivity between nodes of digraph $\mathcal{N}^*$. Now, if URMT$_{MC}$ is possible in $\mathcal{N}$, it would also be possible in $\mathcal{N}^*$, which is a contradiction. (The proof is similar to the proof of Lemma 15, only here both $\mathcal{N}$ and $\mathcal{N}^*$ are asynchronous networks.) This implies that the conditions mentioned in Theorem 17 are necessary.

LEMMA 19. *In the asynchronous network $\mathcal{N}^*$, shown in Figure 1, $(\{\{b_1\}, \{b_2\}\}, \delta)$-URMT$_{MC}$ ($\delta < 1/2$) from s^* to r^* is impossible.*

PROOF. Let us assume that a protocol Π^* exists for $(\{\{b_1\}, \{b_2\}\}, \delta)$-URMT$_{MC}$ from **S** to **R** in the network $\mathcal{N}^*$. Consider two executions E_1 and E_2 of Π^*.

Let us consider E_2 first. In execution E_2, s^* wants to send m_2. Adversary fail-stops b_2, i.e., b_2 does not send or receive messages from s^*, x, b_1 and r^*. For Π^* to be a valid asynchronous URMT$_{MC}$ protocol, there must exist a finite time instant T before which r^* halts with at least $1 - \delta$ probability.

In E_1, the node b_1 is corrupt and s^* chooses to send the message m_1. Node b_1 simulates a local copy of nodes in s^*, say s_1^*, giving it an input $m_2 \neq m_1$ and independent random strings. Node b_1 ignores all messages received from nodes s^*, x and b_2, neither sends any message to them. Also, as the adversary has power to schedule messages, it delays all messages along the directed edge (b_2, r^*) till time T.

Till time instant T, the view at r^* in both executions is same. Hence with at least $1 - \delta$ probability, r^* halts before time T. In this case, for $i \in \{1,2\}$, let p_i be the probability with which it outputs m_i ($p_1 + p_2 \leq 1$). With atmost δ probability, r^* may not halt before time T. In this case, let p_i' be the probability with which it outputs m_i ($p_1' + p_2' \leq 1$). Since for at least one m_i it holds that $p_i(1-\delta) + p_i'\delta \leq 1/2$, therefore Π^* is not a valid URMT$_{MC}$ protocol. ∎

COROLLARY 20. *In a directed network $\mathcal{N} = (V, \mathcal{E})$, an $(\mathbb{A}, \delta)$-URMT$_{LV}$ protocol in synchronous timing model exists if and only if an $(\mathbb{A}, \delta)$-URMT$_{MC}$ protocol in asynchronous timing model exists.*

PROOF. Follows from Theorem 8, 10 and 16, 17. ∎

5 Characterizing asynchronous networks for $(\mathbb{A}, \delta)$-URMT$_{LV}$

THEOREM 21. *In a directed asynchronous network $\mathcal{N}$, $(\mathbb{A}, \delta)$-URMT$_{LV}$ protocol is possible if and only if for every adversary structure $B \subseteq \mathbb{A}$ such that $|B| = 2$, (B, δ)-URMT$_{LV}$ protocol is possible.*

PROOF. Similar to the proof of Theorem 16, hence omitted. ∎

Having reduced the problem of URMT$_{LV}$ in an asynchronous network tolerating an adversary structure to the problem of URMT$_{LV}$ tolerating all its 2-sized subsets, we now proceed to characterize directed asynchronous networks $\mathcal{N} = (V, \mathcal{E})$ in which URMT$_{LV}$ tolerating adversary structure $B = \{B_1, B_2\}$ is possible (where $B_1, B_2 \in \mathbb{A}$).

THEOREM 22. *In a directed asynchronous network $\mathcal{N}$, $(\{B_1, B_2\}, \delta)$-URMT$_{LV}$ protocol is possible if and only if there exists a strong path from* **S** *to* **R** *avoiding nodes in $B_1 \cup B_2$.*

PROOF. *Sufficiency*: Let f be the field element **S** intends to send. Send f to **R** along the strong path avoiding nodes in $B_1 \cup B_2$. As the path does not contain any corrupt nodes, f is eventually received by **R**. *Necessity* is proved in the following section. ∎

5.1 Necessity

We first consider the simple asynchronous network $\mathcal{N}^* = (V^*, \mathcal{E}^*)$ with $V^* = \{s^*, r^*, b_1, b_2\}$ and $\mathcal{E}^* = (V^* \times V^*) \setminus \{(s^*, r^*)\}$.

LEMMA 23. *In the asynchronous network $\mathcal{N}^*$, $(\{\{b_1\}, \{b_2\}\}, \delta)$-URMT$_{LV}$ ($\delta < 1/2$) from s^* to r^* is impossible.*

PROOF. We assume that a protocol Π^* exists in $\mathcal{N}^*$ which is a $(\{\{b_1\}, \{b_2\}\}, \delta)$-URMT$_{LV}$ protocol. We use $\vec{C}_i$ and $view_n(E_i)$ as defined in the proof of Lemma 13. Consider the following two executions of Π^*.

1. Execution E_1: s^* chooses to send m_1. b_2 fail-stops. r^* halts with output m_1 at time instant T_{E_1}. (For Π^* to be a valid URMT$_{LV}$ protocol, such an execution E_1 must exist).
2. Execution E_2: s^* chooses to send $m_2 (\neq m_1)$. Coin tosses $\vec{C}_2$ of nodes are such that $c^1_{r^*} = c^2_{r^*}$. Adversary corrupts node b_1 actively and deploys the following strategy to ensure that $view_{r^*}(E_1) = view_{r^*}(E_2)$. It delays all the outgoing messages from b_2 till beyond the time instant T_{E_1} and abandons any communication with s^*. It simulates a local copy of s^*, say s^*_1, with message m_1 and random coin tosses $c^1_{s^*}$, and gives coin tosses $c^1_{b_1}$ to b_1. Adversary schedules messages among the nodes s^*_1, b_1 and r^* the way it scheduled messages among s^*, b_1 and r^* in execution E_1. The messages received by s^* from r^* in execution E_1 are delivered to s^*_1 by the adversary.

Above mentioned adversary strategy ensures that $view_{r^*}(E_1) = view_{r^*}(E_2)$. Therefore r^* halts with output m_1 in E_2, where s^* chose to send m_2. This leads to a contradiction regarding the existence of Π^*. ∎

We now consider a network $\mathcal{N} = (V, \mathcal{E})$ which does not satisfy the conditions of Theorem 22.

LEMMA 24. *The set of nodes V in the network $\mathcal{N}$ can be partitioned into 4 disjoint sets S^*, B_1, B_2 and R^* such that $\mathbf{S} \in S^*$, $\mathbf{R} \in R^*$ and $\forall\ 1 \leq i < j \leq 4$ an edge exists between a node of $F[i]$ and a node of $F[j]$ only if $(f(i), f(j)) \in \mathcal{E}^*$ where $F = [S^*, B_1, B_2, R^*]$ and $f = [s^*, b_1, b_2, r^*]$ are two ordered lists.*

PROOF. We partition the non-faulty nodes in $H = V \setminus \{B_1 \cup B_2\}$ into 2 disjoint sets S^* and R^*. Let R^* denote the set of all nodes in H who have a strong path to $\mathbf{R}$ avoiding nodes in $B_1 \cup B_2$. Let $S^* = V \setminus \{R^* \cup B_1 \cup B_2\}$. Since every strong path from $\mathbf{S}$ to $\mathbf{R}$ passes through some node in $B_1 \cup B_2$, $S \in S^*$. Also, there cannot exist any directed edge from a node u in S^* to a node in R^*, otherwise u would move to R^*. ∎

LEMMA 25. *In the directed asynchronous network $\mathcal{N} = (V, \mathcal{E})$, $(\{B_1, B_2\}, \delta)$-URMT$_{LV}$ is possible from $\mathbf{S}$ to $\mathbf{R}$ only if $(\{\{b_1\}, \{b_2\}\}, \delta)$-URMT$_{LV}$ is possible from s^* to r^* in the network $\mathcal{N}^*$.*

PROOF. Proof is straightforward using standard player simulation technique. ∎

From Lemma 23 we know that $(\{\{b_1\}, \{b_2\}\}, \delta)$-URMT$_{LV}$ is impossible from s^* to r^* in the network $\mathcal{N}^*$. We arrive at a contradiction. Hence, the conditions mentioned in Theorem 22 are necessary.

THEOREM 26. *In a directed synchronous (or asynchronous) network $\mathcal{N} = (V, \mathcal{E})$, $\mathbb{A}$-PRMT from $\mathbf{S}$ to $\mathbf{R}$ is possible if and only if for all $B_1, B_2 \in \mathbb{A}$ there exists a strong path from $\mathbf{S}$ to $\mathbf{R}$ avoiding nodes in $B_1 \cup B_2$.*

PROOF. Follows from [3]. ∎

COROLLARY 27. *In a directed network $\mathcal{N} = (V, \mathcal{E})$, an $(\mathbb{A}, \delta)$-URMT$_{LV}$ protocol in asynchronous timing model exists if and only if an $\mathbb{A}$-PRMT protocol exists.*

PROOF. Follows from Theorem 21, 22 and 26. ∎

6 Critical edges

In [7], Bhavani et. al., for all $t > 0$, construct a family of graphs G_t given by: $G_t = (V, \mathcal{E}_1 \cup \mathcal{E}_2 \cup \mathcal{E}_3)$ with V $= \{\mathbf{S}, v_1, \ldots, v_{t+1}, u_1, \ldots, u_t, \mathbf{R}\}$ and $\mathcal{E}_1 = \bigcup_1^{t+1}\{(\mathbf{S}, v_i), (v_i, \mathbf{R})\}$; $\mathcal{E}_2 = \bigcup_{i=1}^{t}\{(\mathbf{S}, u_i), (\mathbf{R}, u_i)\}$; $\mathcal{E}_3 = \bigcup_{i=1}^{t}\{(u_i, v_1), \ldots, (u_i, v_{t+1})\}$ as shown in Figure 2(a) and claimed that G_t has $\Omega(n^2)$ critical edges w.r.t synchronous (t, δ)-URMT$_{MC}$. In the following theorem, we prove that this is not the case.

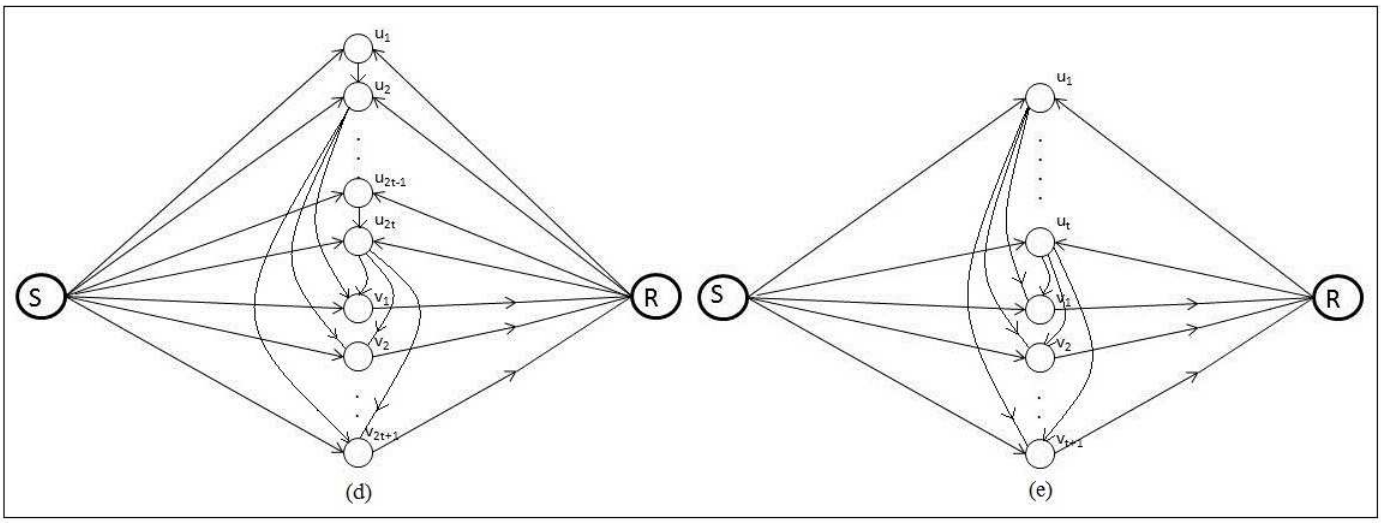

Figure 2: (a) Graph G_t, (b) Graph H_t

THEOREM 28. *G_t has only $\Theta(n)$ critical edges w.r.t synchronous (t,δ)-URMT$_{MC}$.*

PROOF. Consider the subgraph G'_t of G_t given by $G'_t = (V, \mathcal{E}_1 \cup \mathcal{E}_2 \cup \mathcal{E}'_3)$, where $\mathcal{E}'_3 = \bigcup_{i=1}^{t}\{u_i, v_i\}$. We try to construct two sets of nodes B_1, B_2 ($|B_1|, |B_2| \leq t$ and $B = \{B_1, B_2\}$) such that (B,δ)-URMT$_{MC}$ is impossible in G'_t. If we are successful then (t,δ)-URMT$_{MC}$ would also be impossible in G'_t. Otherwise it would be possible.

According to [7], for (B,δ)-URMT$_{MC}$ to be impossible in G'_t, every honest weak path from **S** to **R** must have at least node x such that every strong path from x to **R** passes through both B_1 and B_2. If there exists a strong path from **S** to **R** avoiding nodes in $B_1 \cup B_2$, URMT$_{MC}$ is trivially possible. Hence $\forall i\ 1 \leq i \leq t+1\ v_i \in B_1$ or $v_i \in B_2$. As $|B_1| + |B_2| \leq 2t$, at least one u_i has to be honest. As this leaves an honest weak path from **S** to **R** ($(\mathbf{S}, u_i), (\mathbf{R}, u_i) \in \mathcal{E}_2$) with u_i having a strong path to **R** ($(u_i, v_i) \in \mathcal{E}_3$ and $(v_i, \mathbf{R}) \in \mathcal{E}_1$), node v_i must belong to both B_1 and B_2. This would imply that another $u_{i'}$ ($i' \neq i$) is honest and hence $v_{i'}$ must belong to both B_1 and B_2. Arguing similarly for $t-2$ more steps, we can show that $B_1 = B_2 = \{v_{\alpha_1}, v_{\alpha_2}, \ldots, v_{\alpha_t}\}$ for some $\{\alpha_1, \alpha_2, \ldots, \alpha_t\} \subset \{1, 2, 3, \ldots, t+1\}$. But this leaves a strong honest path from **S** to **R**. Hence, construction of a B such that (B,δ)-URMT$_{MC}$ is impossible in G'_t is not possible. So, (t,δ)-URMT$_{MC}$ must be possible in G'_t.

As G'_t has $O(n)$ edges, this proves an upper bound of $O(n)$ on the number of critical edges in G_t. Hence, the claim in [7] that G_t has $\Omega(n^2)$ critical edges is wrong. Moreover, as deleting any one edge $(\mathbf{S}, v_i)$ leaves only $2t$ disjoint weak paths between **S** and **R**, G_t has $\Omega(n)$ critical edges. It therefore follows that G_t has $\Theta(n)$ critical edges. ∎

For all $t > 0$, consider a family of graphs H_t given by $H_t = (\mathrm{V}^1, \bigcup_{i=1}^{4} \mathcal{E}_i^1)$ with $\mathrm{V}^1 = \{\mathbf{S}, v_1, \ldots, v_{2t+1}, u_1, \ldots, u_{2t}, \mathbf{R}\}$ and $\mathcal{E}_1^1 = \bigcup_1^{2t+1}\{(\mathbf{S}, v_i), (v_i, \mathbf{R})\}$; $\mathcal{E}_2^1 = \bigcup_{i=1}^{2t}\{(\mathbf{S}, u_i), (\mathbf{R}, u_i)\}$; $\mathcal{E}_3^1 = \bigcup_{i=1}^{t}\{(u_{2i-1}, u_{2i}))\}$; $\mathcal{E}_4^1 = \bigcup_{i=1}^{t}\{(u_{2i}, v_1), \ldots, (u_{2i}, v_{2t+1})\}$ as shown in Figure 2(b). Here, number of nodes in graph H_t is $n = 4t + 3$.

THEOREM 29. *H_t has $\Omega(n^2)$ critical edges w.r.t synchronous $(2t,\delta)$-URMT$_{MC}$.*

PROOF. $(2t,\delta)$-URMT$_{MC}$ is possible in H_t (Follows from [7]). Suppose we delete any edge $e = (u_{2i}, v_j) \in \mathcal{E}_3^1$ $(1 \leq i \leq t, 1 \leq j \leq 2t+1)$. Consider, $B_1 = \bigcup_{k=1}^{2t}\{u_k\} \cup \{v_j\} - \{u_{2i-1}\}$, $B_2 = \bigcup_{k=1}^{2t+1}\{v_k\} - \{v_j\}$. Only honest weak path left is $\mathbf{S} \rightarrow u_{2i-1} \leftarrow \mathbf{R}$. Every strong path from u_{2i-1} to $\mathbf{R}$ passes through both B_1 and B_2, rendering $(\{B_1, B_2\}, \delta)$-URMT$_{MC}$, and hence (t,δ)-URMT$_{MC}$, impossible. Hence, H_t has $\Omega(|\mathcal{E}_3^1|)$ or $\Omega(n^2)$ critical edges. ∎

THEOREM 30. *G_t has $\Omega(n^2)$ critical edges w.r.t asynchronous (t,δ)-URMT$_{MC}$ (and synchronous (t,δ)-URMT$_{LV}$ as well).*

PROOF. (t,δ)-URMT$_{MC}$ is possible in asynchronous network G_t (Follows from Theorem 16, 17). Suppose we delete any edge $e = (u_i, v_j) \in \mathcal{E}_3$ $(1 \leq i \leq t, 1 \leq j \leq t+1)$ from G_t. Consider, $B_1 = \bigcup_{k=1}^{t}\{u_k\} \cup \{v_j\} - \{u_i\}$, $B_2 = \bigcup_{k=1}^{t+1}\{v_k\} - \{v_j\}$. Only honest weak path left is $\mathbf{S} \rightarrow u_i \leftarrow \mathbf{R}$. All the strong paths from u_i to $\mathbf{R}$ pass through B_2, rendering $(\{B_1, B_2\}, \delta)$-URMT$_{MC}$, and hence (t,δ)-URMT$_{MC}$, impossible. Therefore all the edges in $\mathcal{E}_3$ are critical. Hence $\Omega(n^2)$ critical edges. ∎

7 Discussion and Open Problems

In this paper, we show that in a directed network under the influence of a Byzantine adversary, a protocol for synchronous *Las Vegas* URMT exists if and only if a protocol for asynchronous *Monte Carlo* URMT exists. This equivalence has immediate impact in practice. Suppose a hardware vendor has to design a network such that Monte Carlo URMT between given two nodes is possible even when the network is asynchronous and deploy hardware for it. Testing of asynchronous networks is a tedious and time consuming job. So, he may rather test if the hardware achieves Las Vegas URMT when the network is synchronous, and be rest assured. Hence it is important to explore such equivalences w.r.t. other Distributed Computing problems.

We briefly discuss a few related open problems: (a) There are networks for which directed hypergraphs are a better model. Hence, it is useful to study the problem of URMT in directed hypergraphs under various timing models, fault models, etc. It would be interesting to see if the equivalence shown above for directed graphs in Theorem 20 extends to directed hypergraphs as well. (a) Given a network $\mathcal{N}$, a sender $\mathbf{S}$, a receiver $\mathbf{R}$ and an adversary $\mathbb{A}$, the decision problem of finding out whether URMT$_{LV}$ in synchronous network is not possible is trivially in the complexity class NP. It is non-trivial to prove this decision problem to be certain complexity class hard. (b) As we focussed on characterizing directed networks in which URMT is possible in this paper, our protocols may be inefficient. It remains open to give efficient protocols or establish lower bounds.

References

[1] Michael Ben-Or, Ran Canetti, and Oded Goldreich. Asynchronous secure computation. In *STOC '93: Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 52–61, New York, NY, USA, 1993. ACM.

[2] Y. Desmedt and Y. Wang. Perfectly Secure Message Transmission Revisited. In *Proceedings of Advances in Cryptology EUROCRYPT '02*, volume 2332 of *LNCS*, pages 502–517, 2002.

[3] Danny Dolev, Cynthia Dwork, Orli Waarts, and Moti Yung. Perfectly secure message transmission. *J. ACM*, 40(1):17–47, 1993.

[4] M. Franklin and R. N. Wright. Secure Communication in Minimal Connectivity Models. In *Proceedings of Advances in Cryptology EUROCRYPT '98*, volume 1403 of *LNCS*, pages 346–360, 1998.

[5] M. Hirt and U. Maurer. Player Simulation and General Adversary Structures in Perfect Multiparty Computation. *Journal of Cryptology*, 13(1):31–60, April 2000.

[6] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *STOC '89: Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 73–85, New York, NY, USA, 1989. ACM.

[7] Bhavani Shankar, Prasant Gopal, Kannan Srinathan, and C. Pandu Rangan. Unconditionally reliable message transmission in directed networks. In *SODA '08: Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1048–1055, 2008.

[8] Kannan Srinathan and C. Pandu Rangan. Possibility and complexity of probabilistic reliable communications in directed networks. In *Proceedings of 25th ACM Symposium on Principles of Distributed Computing (PODC'06)*, 2006.