# Topology Knowledge Versus Fault Tolerance: The Case of Probabilistic Communication
# Or: How Far Must You See to Hear Reliably?

Pranav K. Vasishta    Prasant Gopal    Anuj Gupta    Piyush Bansal    Kannan Srinathan
Center for Security, Theory and Algorithmic Research,
International Institute of Information Technology, Hyderabad, India.
email: {vasishta,prasant,anujgupta,piyush_bansal}@research.iiit.ac.in, srinathan@iiit.ac.in

**Abstract**

We consider the problem of probabilistic reliable communication (PRC) over synchronous networks modeled as directed graphs in the presence of a Byzantine adversary when players' knowledge of the network topology is not complete. We show that possibility of PRC is extremely sensitive to the changes in players' knowledge of the topology. This is in complete contrast with earlier known results on the possibility of perfectly reliable communication over undirected graphs where the case of each player knowing only its neighbours gives the same result as the case where players have complete knowledge of the network. Specifically, in either case, $2t + 1$-vertex connectivity is necessary and sufficient, where $t$ is the number of nodes that can be corrupted by the adversary[7][8]. We use a novel model for quantifying players' knowledge of network topology, denoted by $\mathcal{T}K$. Given a directed graph $G$, influenced by a Byzantine adversary that can corrupt up to any $t$ players, we give a necessary and sufficient condition for possibility of PRC over $G$ for any arbitrary topology knowledge $\mathcal{T}K$.

## 1 Introduction

A number of non-trivial fault-tolerant distributed computing problems may be abstracted as : For a network $\mathcal{N}$ (modeled as a graph), a problem $\pi$, and an adversary $\mathcal{A}$, given any two of them, give a relation of the third with respect to the other two. For example, supposing problem $\pi$ and adversary $\mathcal{A}$ are given, give the condition on $\mathcal{N}$ such that $\pi$ is solvable. In all such cases, it is *generally* assumed that the players in $\mathcal{N}$ have complete knowledge of the network topology, that is, they know the graph $G$, which makes up $\mathcal{N}$. However, this assumption is not true in most practical circumstances such as the Internet. Moreover, theoretically, it is curious to consider the effect of players with varying amounts of knowledge about the network topology in fault-tolerant distributed computation. In this paper, we refer to the notion of knowledge possessed by a node about the graph $G$ by Topology Knowledge $\mathcal{TK}$ at that node. Considering $\mathcal{TK}$ as another parameter in our abstraction, we ask: For a network $\mathcal{N}$, a problem $\pi$, an adversary $\mathcal{A}$, and topology knowledge $\mathcal{TK}$ given for each node in $\mathcal{N}$, give the condition on $\mathcal{N}$ such that $\pi$ is solvable. To the best of our knowledge, there has not been any focused study in the literature on how $\mathcal{TK}$ affects distributed computability in the presence of an adversary. This is probably the first such attempt. Sense of Direction, which is a labeling property has been studied in the extant literature as a part of larger study on structural knowledge. The literature indicates that this property is a fundamental requirement for distributed computability (See [2], [1]). These studies also have results on distributed complexity for various topologies. We wish to indicate that in this paper, we consider labeling of the network as fixed, similar to that in [7] and has no current relation with the extant studies in structural knowledge. We give the first results showing that including $\mathcal{TK}$ as a parameter affects fault-tolerant distributed computation. We take the problem of probabilistic reliable communication ([3],[4],[6],[5]) as an instance to show this result. Note that the problem of reliable communication is a fundamental primitive for any non-trivial distributed protocol. Specifically, if reliable communication were possible between all pairs of players in the network (with some arbitrary topology knowledge $\mathcal{TK}$) then all the non-faulty players can *simulate* and overlay a complete subgraph among them thereby alleviating the issues posed by the adversary as well as the (lack of) topology knowledge! Thus, in some sense, the problem of probabilistic reliable communication is at the core of understanding the effect of topology knowledge on distributed computing in general.

One can easily note that the notion of $\mathcal{TK}$ is relevant only in the presence of a suitable adversary. In the absence of a suitable adversary, each node can let the rest of the players (of course only as far as the connectivity permits!) know its view of the network in order to end up with a global common picture of the topology for each of the connected components respectively. Consequently, while the distributed *complexity* may increase, the distributed *computability* is unaffected. However, in the presence of the adversary, any amount of communication would entail only an (adversary controlled) approximation of the actual topology, thereby perhaps affecting even the distributed computability. Intuitively though, the adversary can at best completely hide the edges between two faulty players and if lucky, succeed in partially hiding even the edges with one faulty end-node. However, useful messages are seldom transmitted via the aforementioned edges. This gives a feeling that distributed *computability* may not be affected — in fact, for the case of perfectly reliable communication it has been proved that the knowledge of one's neighbors alone is as sufficient as the knowledge of the global topology; specifically, $(2t+1)$-connectivity is necessary and sufficient irrespective of $\mathcal{TK}$, where up to $t$ players are Byzantine faulty[7]. Counter-intuitively, for the case of probabilistic communication, we show that the optimal fault-tolerance heavily *depends* on $\mathcal{TK}$.

The contributions of this paper are multi-fold. First, a formal characterization of what $\mathcal{TK}$ entails has been given in Section 2. Second, in Section 3, we reduce the task of deciding the computability of any problem $\pi$ with an arbitrary and complex $\mathcal{TK}$ to the task of deciding the computability of $\pi$ with well-defined and simple $\mathcal{TK}$.We use a reduction from $\mathcal{TK}$ to a uniform $\mathcal{TK}$ so as to solve for possibility of PRC with ease, this is shown in Section 4 Subsequently, we apply our reduction to the problem of probabilistic reliable communication (PRC) and in Section 5.1 prove a necessary and sufficient condition on $\mathcal{TK}$ for the existence of protocols for PRC with a specified fault-tolerance. Implicit in our results is the fact that optimal fault-tolerance increases as knowledge of the network topology improves.

As a motivating example, we give the graphs $G$ and $G'$, see Figures 1,2. Both the graphs satisfy the conditions for a successful PRC between $\mathbf{S}$ and $\mathbf{R}$ in the presence of a $t$-Byzantine adversary, given complete topology knowledge [5]. Supposing that all players, especially $\mathbf{R}$ is aware of all the edges in the network, that is it knows the graph which represents the network, then PRC protocol exists. On the other hand, if topology knowledge of $\mathbf{R}$ includes both the graphs $G$ and $G'$, one of which is the actual graph, then we show that no such PRC protocol exists, as the topology knowledge here,where each player knows only about the edges adjacent to him, is insufficient for successful reliable communication.

## 2 Model and Definitions

The network is modeled as a directed graph $\mathcal{N} = (\mathbb{P}, \mathcal{E})$ where $\mathbb{P}$ is the set of vertices and $\mathcal{E}$ denotes the set of arcs/edges in the directed graph. The system is assumed to be synchronous, that is, the protocol is executed in a sequence of *rounds* wherein in each round, a player can perform some

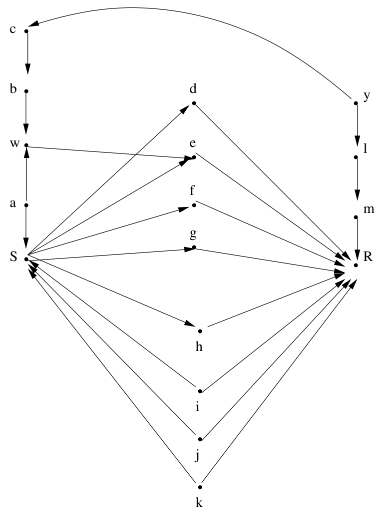


Figure 1: Graph $G$

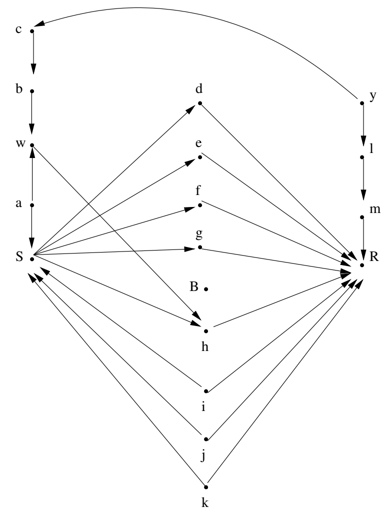


Figure 2: Graph $G'$

local computation, send new messages to his out-neighbors, receive the messages sent in that round by his in-neighbors (and if necessary perform some more local computation), in that order. In the graph, we assume that the channels are *secure*. In other words, if $(u, v) \in \mathcal{E}$ then *the player $u$ can securely send a message to player $v$ in one round*. During the execution, the adversary may corrupt up to any $t$ players. We work with a Byzantine adversary that may completely control all the corrupted players and make them behave in arbitrary fashion. Every honest player that receives a message from its in-neighbor knows the sender as it can identify the channel along which the message is received. We give a couple of definitions before detailing our model further.

**Definition 1 (Topology Knowledge of *player i* ($\mathcal{TK}_i$))** *In a given network $\mathcal{N} = (\mathbb{P}, \mathcal{E})$, we define* Topology Knowledge of player i $\mathcal{TK}_i$ *as the knowledge possessed by* player i *about the network $\mathcal{N}$. It is represented by a set of graphs,i.e. $\mathcal{TK}_i = \{G^i_{k_i}\}$, with a condition that one of the graphs from the set $\mathcal{TK}_i$ is the actual graph $\mathcal{N}$ where $1 \leq k_i < 2^{|\mathbb{P}|^2}$ and $1 \leq |\mathcal{TK}_i| < 2^{|\mathbb{P}|^2}$*

**Definition 2 (Topology Knowledge $\mathcal{TK}$)** *We define the notion Topology Knowledge $\mathcal{TK}$ as the collection of all the individual $\mathcal{TK}_i$'s (for all the players in $\mathbb{P}$). Specifically, $\mathcal{TK} = \{\mathcal{TK}_i | i \in \mathbb{P}\}$.*

We assume each *player i* knows the following at the outset: (a) Set of vertices in $\mathcal{N}, i.e, \mathbb{P}$; (b) Topology Knowledge of *player i*, $\mathcal{TK}_i$. We also assume the worst-case scenario where the adversary knows all the $\mathcal{TK}_i$'s for $i \in \mathbb{P}$ as well as is aware of the actual graph $G$.

We explain now, as to why $\mathcal{TK}_i$ is so defined. In order that a set of graphs be considered a model of the notion of *knowledge possessed by* player i *about the network $\mathcal{N}$*, we require that the set contains the actual graph $\mathcal{N}$. The intuition behind this way of modeling is based on the assumption (a) above. Given that a *player i* knows the set of players, he also knows that the actual graph $G$ must be one of the $2^{|\mathbb{P}|^2}$ directed graphs that are ever possible among the players. Note that if $\mathcal{TK}_i$ contains all possible graphs it captures the situation where the player $i$ is completely oblivious of the edges in $G$ (including *his own neighbors*). At the other extreme, if $\mathcal{TK}_i$ contains just a single graph $G$ it captures the scenario where the player $i$ is completely knowledgeable about the topology of the network (including the edges between two faulty players!). Note that neither of these extreme cases are likely to occur at runtime since one usually knows at least (the number) of his neighbors and one usually can never be sure about the presence/absense of edges between faulty players.

Note that it is not necessary to represent $\mathcal{TK}_i$ by physically enumerating all the graphs that $\mathcal{TK}_i$ contains. This would certainly be an impractical representation. Rather, it may be enough if the player $i$ has a program/circuit which on input a graph $H$ can decide whether or not $H \in \mathcal{TK}_i$. For instance, if player $i$ is aware of his neighbors, then $\mathcal{TK}_i$ would contain only those graphs which are consistent with respect to player $i$'s neighborhood. This of course can easily be represented by a program that on input a graph $H$, tests the neighborhood of $i$ in $H$ and decides whether or not $H$ belongs to $\mathcal{TK}_i$. However, we remark that a simple counting argument suggests that there exists a $\mathcal{TK}_i$ that cannot be decided via any *efficient* circuit (or for that matter, inefficiency is inexorable irrespective of what representation is used). Such a $\mathcal{TK}_i$ may entail a super-polynomial surge in the complexity of any protocol wishing to use its knowledge fully. While a practical approach may be to approximate $\mathcal{TK}_i$ to the "nearest" efficiently decidable $\mathcal{TK}'_i$ (thereby losing some information), we stress that this work is about *computability* in the presence of the adversary and we therefore, without loss of generality, take the liberty of not focussing on issue regarding the representation of $\mathcal{TK}_i$. In other words, we model the players (as well as the adversary) as interactive Turing machines with unbounded computing power and *not* as probabilistic polynomial-time Turing machines (PPT). We leave the task of studying the effects of topology knowledge in a model where the players are PPT's as an interesting open problem.

**Note 3** *When each of the graphs is written in $G = (V, E)$ form, we note that $G^i_k = (\mathbb{P}, \mathcal{E}^i_{k_i})$, and only the edge-set keeps changing across the graphs for each of the players. Therefore, we can replace every graph $G^i_{k_i}$ in the set $\mathcal{TK}_i$ with the set of its edge-sets $\{\mathcal{E}^i_{k_i}\}$ for each* player i *to make matters more convenient. We will be using $\mathcal{TK}_i$ synonymously with $\{\mathcal{E}^i_{k_i}\}$ from now on.*

A comment on the above modeling is due: The notion of knowledge of the network topology has a number of elements in it – such as the size of the network, the labeling of the network, edge-sets, location awareness etc. Modeling all the elements of the topology knowledge so as to make it as general as possible is not the focus of this paper. We assume that all other parameters are published but for the edge-set of the network. Thus, our focus is on the effect of the knowledge of edge-set (with other topological aspects as public information) on fault-tolerance in distributed computing. Accordingly, note that our model offers flexibility for one to choose between various levels of edge information that may be provided to the players. This concludes our discussion on modeling $\mathcal{TK}$.

**Definition 4 (k-sized $\mathcal{TK}_i$ and k-sized $\mathcal{TK}$)** *A k-sized $\mathcal{TK}_i$ is defined as $\mathcal{TK}_i$ where $|\mathcal{TK}_i| = k$. If there exists an integer $k$ such that every $\mathcal{TK}_i$ in the collection $\mathcal{TK}$ is $\ell$-sized for some $\ell \leq k$, then the topology knowledge $\mathcal{TK}$ is defined as a k-sized $\mathcal{TK}$.*

**Note 5** *The edge knowledge of a player with 2-sized $\mathcal{TK}_i$ is greater than the edge knowledge of a player with k-sized $\mathcal{TK}_i$, $k > 2$. This follows from the fact that a player with 2-sized $\mathcal{TK}_i$ has only 2 edge-sets one of which is the correct one as opposed to a player with k-sized $\mathcal{TK}_i$ who has k edge-sets with the knowledge that one of them is the correct edge-set. That is, the respective probabilities of choosing the correct edge-set is much higher at $1/2$ for the former case as opposed to $1/k$ for the latter.*

## 3 From N-sized $\mathcal{TK}$ to 2-sized $\mathcal{TK}$

In this section, we show that for any problem $\pi$, working with an N-sized $\mathcal{TK}$ is equivalent to working with (several) 2-sized $\mathcal{TK}$ with respect to the solvability of $\pi$.

**Definition 6** *Given a graph $G = (\mathbb{P}, E)$ with N-sized $\mathcal{TK} = \{\mathcal{TK}_1, \mathcal{TK}_2, \ldots, \mathcal{TK}_{|\mathbb{P}|}\}$, $N > 2$, where each $\mathcal{TK}_i = \{\mathcal{E}^i_1, \mathcal{E}^i_2, \ldots, \mathcal{E}^i_k\}$, $k \leq N$, and a 2-sized topology knowledge $\mathcal{Z} = \{\mathcal{Z}_1, \mathcal{Z}_2, \ldots, \mathcal{Z}_{|\mathbb{P}|}\}$ with each $\mathcal{Z}_i = \{A^i_0, A^i_1\}$, $1 \leq i \leq |\mathbb{P}|$, such that one of the $A^i_j$'s is exactly the edge-set $E$ (the edge-set of $G$) while the other is an edge-set (different from $E$) that is present in $\mathcal{TK}_i$. In other words, there exists a $j \in \{0, 1\}$ such that $E = A^i_j$ and $A^i_{\bar{j}} \in \mathcal{TK}_i$, we say that $\mathcal{Z}$ is derived from $\mathcal{TK}$.*

Note that an N-sized $\mathcal{TK}$ can have up to $(N-1)^{|\mathbb{P}|}$ distinct 2-sized topology knowledge sets derived from it.

**Theorem 3.1** *A protocol $\Pi$ that solves a problem $\pi$ over a network $\mathcal{N}$ with $N$-sized $\mathcal{TK}$ ($N > 2$) influenced by an adversary $\mathcal{A}$ exists if and only if, for each of the 2-sized topology knowledge ($\mathcal{Z}$) that can be derived from $\mathcal{TK}$, there exists a protocol solving $\pi$ with topology knowledge $\mathcal{Z}$.*

*Proof:* Refer to AppendixB. ■

# 4 Towards a More Uniform $\mathcal{TK}$

From the prequel, it is enough to characterize the possibility of PRC for a 2-sized $\mathcal{TK}$. For a set of players $\mathbb{P}$, a 2-sized $\mathcal{TK} = \{\mathcal{TK}_1, \mathcal{TK}_2, \ldots, \mathcal{TK}_{|\mathbb{P}|}\}$, where each $\mathcal{TK}_i = \{\mathcal{E}^i_0, \mathcal{E}^i_1\}$, $\forall i \in \mathbb{P}$. Notice that each player can have a different $\mathcal{TK}_i$, and working with varying $\mathcal{TK}_i$s can become cumbersome. In our characterization, we avoid working directly with 2-sized $\mathcal{TK}$ for this reason. In this section, we show that for any problem $\pi$, working with a 2-sized $\mathcal{TK}$ can be brought down to working with a modified $\mathcal{TK}$, say, $\mathcal{TK}'$ which is formed by the intersection of each of the $\mathcal{TK}_i$s in 2-sized $\mathcal{TK}$ and a set denoted by 2-sized $\mathcal{TK}_G$ which in all respects has the properties of a 2-sized $\mathcal{TK}_i$ and is known globally to all the players. There are $2^{|\mathbb{P}|^2}$ possibilities for 2-sized $\mathcal{TK}_G$. We show that solving $\pi$ using a 2-sized $\mathcal{TK}$ is possible if and only if $\pi$ can be solved for all the $2^{|\mathbb{P}|^2}$ possibilities for $\mathcal{TK}'$. Following this, we give a necessary and sufficient condition for the possibility of PRC given $\mathcal{TK}'$.

## 4.1 From 2-sized $\mathcal{TK}$ to $\mathcal{TK}'$

We begin with the following definitions:

**Definition 7 (Global $\mathcal{TK}_G$ and $k$-sized $\mathcal{TK}_G$)** *For a given network $\mathcal{N} = (\mathbb{P}, \mathcal{E})$, a global $\mathcal{TK}_G$ is defined as a globally published topology information accessible to all the players in the network $\mathcal{N}$. Its representation and properties are similar to that of $\mathcal{TK}_i$. It is represented as a set of graphs, i.e. $\mathcal{TK}_G = \{G_i\}$ with the condition that one of the graphs in $\mathcal{TK}_G$ is the actual graph $\mathcal{N}$, where $1 \leq |\mathcal{TK}_G| < 2^{|\mathbb{P}|^2}$. If $|\mathcal{TK}_G| = k$, then the $\mathcal{TK}_G$ shall be called $k$-sized $\mathcal{TK}_G$. Similar to Note 3, $\mathcal{TK}_G$ can be represented using the corresponding edge-sets of the graphs in it.*

**Definition 8 ($\mathcal{TK}'_i$ and $\mathcal{TK}'$)** *Given a network $\mathcal{N} = (\mathbb{P}, \mathcal{E})$, a $\mathcal{TK}_G$ and a $\mathcal{TK}$, then we define $\mathcal{TK}'_i$ as the updated topology knowledge of* player i *from knowing $\mathcal{TK}_G$. Collection of all such $\mathcal{TK}'_i$s forms $\mathcal{TK}'$,i.e., $\mathcal{TK}' = \{\mathcal{TK}_i \cap \mathcal{TK}_G\}$ $\forall i \in \mathbb{P}$. Similar to Note 3, $\mathcal{TK}'$ can be represented using the corresponding edge-sets of the graphs in it.*

**Theorem 4.1** *A protocol $\Pi$ that solves a problem $\pi$ over a network $\mathcal{N}$ with 2-sized $\mathcal{TK}$ influenced by an adversary $\mathcal{A}$ exists if and only if, for each of the $\mathcal{TK}'$ sets formed from the $2^{|\mathbb{P}|^2}$ possibilities for the 2-sized $\mathcal{TK}_G$, there exists a protocol solving $\pi$ with $\mathcal{TK}'$.*

*Proof: Only-If part:* This is the easier part. It is evident that the topology knowledge $\mathcal{TK}'$ is higher than the topology knowledge $\mathcal{TK}$, since every $\mathcal{TK}_i$ is a superset of $\mathcal{TK}'_i = \mathcal{TK}_i \cap \mathcal{TK}_G$. Therefore, if a protocol $\Pi$ works correctly over $\mathcal{TK}$, it would vacuously be a correct protocol over $\mathcal{TK}'$ too.

*If part:* Let there exist protocols for problem $\pi$ for each of the valid 2-sized $\mathcal{TK}_G$s. We now show that this implies that there exists protocols for $\pi$ for any valid 3-sized $\mathcal{TK}_G$. Specifically, let a 3-sized $\mathcal{TK}_G$ be $\mathcal{T} = \{\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3\}$. Consider the three 2-sized subsets, namely, $X_1 = \{\mathcal{E}_1, \mathcal{E}_2\}$, $X_2 = \{\mathcal{E}_2, \mathcal{E}_3\}$ and $X_3 = \{\mathcal{E}_1, \mathcal{E}_3\}$. Note that at least two of the above three $X_i$'s are valid global $\mathcal{TK}_G$'s (that is they contain the actual edge set). Suppose the players execute the protocol for solving $\pi$ for each of these two $\mathcal{TK}_G$s, their outputs must exactly match since the problem is solved over the same network with the same inputs! However, the players are unaware of which of the two $X_i$'s are valid global $\mathcal{TK}_G$s. It turns out that this does not matter since the players could execute protocols for all the three $X_i$'s and perform a majority voting on the outputs to obtain the correct output for problem $\pi$ for the 3-sized $\mathcal{TK}_G$, namely $\mathcal{T}$.

By induction, one can now solve the problem $\pi$ for any given 4-sized $\mathcal{TK}_G$ (since any 4-sized $\mathcal{TK}_G$ can be split into three subsets of size $\leq 3$ such that at least two of them valid and yield exactly the same output). Continuing further, we find that solving $\pi$ with any $m$-sized $\mathcal{TK}_G$ (where $2 < m \leq 2^{|\mathbb{P}|^2}$) is possible if and only if $pi$ is solvable for each of its 2-sized subset $\mathcal{TK}_G$s.

Notice that the case of $m = 2^{|\mathbb{P}|^2}$ is nothing but the case where $\mathcal{TK}$ is exactly equal to $\mathcal{TK}'$. Hence the theorem. ∎

# 5 Complete Characterization of PRC given $\mathcal{TK}'$

We begin with a few definitions and lemmas.

**Definition 9 (PRC Protocol)** *Let $\mathcal{N} = (\mathbb{P}, \mathcal{E})$ be a network, with topology knowledge $\mathcal{TK}$, under the influence of a Byzantine adversary that may corrupt up to any $t$ players. We say that a protocol for transmitting a message from* **S** *to* **R** *is $(t, \delta)$-reliable if for any valid adversary strategy, the probability that* **R** *outputs* **m** *given that* **S** *has sent* **m***, is at least $\delta$ where the probability is over the random inputs of all the players and random inputs of the adversary.*

**Definition 10 (Strong Path)** *A sequence of vertices $v_1, v_2, v_3, \ldots, v_k$ is said to be a strong path from $v_1$ to $v_k$ in the network $\mathcal{N} = (\mathbb{P}, \mathcal{E})$ if for each $1 \leq i < k$, $(v_i, v_{i+1}) \in \mathcal{E}$. Furthermore, we assume that there vacuously exists a strong path from a node to itself.*

**Definition 11 ($t$-(S,R,)-strong-connectivity)** *A digraph is said to be $t$-(***S,R***)-strong-connected if the graph is such that there exists at least $t$ vertex disjoint strong paths from* **S** *to* **R***.*

**Definition 12 (Semi-Strong Path)** *A sequence of vertices $v_1, v_2, v_3, \ldots, v_k$ is said to be a semi-strong path from $v_1$ to $v_k$ in the network $\mathcal{N} = (\mathbb{P}, \mathcal{E})$ if there exists a $1 \leq j \leq k$ such that the sequence $v_j$ to $v_1$ as well as $v_j$ to $v_k$ both are strong paths in the network. We call the vertex $v_j$ as the head of the semi-strong path.*

**Definition 13 (Weak Path)** *A sequence of vertices $v_1, v_2, v_3, \ldots, v_k$ is said to be a weak path from $v_1$ to $v_k$ in the network $\mathcal{N} = (\mathbb{P}, \mathcal{E})$ if for each $1 \leq i < k$, either $(v_i, v_{i+1}) \in \mathcal{E}$ or $(v_{i+1}, v_i) \in \mathcal{E}$ . Furthermore, we assume that there vacuously exists a weak path from a node to itself.*

**Definition 14 ((Player p)-Group)** *(Player $p$)-group is defined with respect to two $t$-sized subsets of $\mathbb{P}$, say $B_1$ and $B_2$. A $p$-group in graph $G$ is the set of all players $q$ (including $p$) such that $q$ has a strong path from it to $p$ not passing through any node in $(B_1 \cup B_2)$.*

**Definition 15 (Critical Combination)** *Given the following:*
*– A network $\mathcal{N} = (\mathbb{P}, \mathcal{E})$, where $\mathbb{P}$ is the set of players, and $\mathcal{E}$, the set of edges between the players in $\mathbb{P}$*
*– Two players identified as* **S** *(denoting sender) and* **R** *(denoting receiver) such that $\{\mathbf{S}, \mathbf{R}\} \in \mathbb{P}$*
*– Two $t$-sized sets - $B_1$ and $B_2$, such that $B_1 \subset \mathbb{P}$, $B_2 \subset \mathbb{P}$ in the network $\mathcal{N}$*
*– A 2-sized $\mathcal{TK}_G = \{E_0, E_1\}$ for the network $\mathcal{N}$*
*– A 2-sized $\mathcal{TK}$ for the network $\mathcal{N}$*
*– A set $X$ of players, $X = \{i | \mathcal{TK}'_i = \mathcal{N}$ and $i \in (\mathbb{P} \setminus B_1 \cup B_2 \cup$ **R**-group$)\}$, i.e., players in $(\mathbb{P} \setminus B_1 \cup B_2 \cup$**R**-group) which know the actual graph $\mathcal{N}$ after knowing $\mathcal{TK}_G$.*
*– $X \cap$**R**-group $= \phi$*
*– A set $Z \in (\mathbb{P} \setminus B_1 \cup B_2 \cup$ **R**-group) of players that are part of all weak paths from* **S** *to* **R** *and those players that have a semi-strong path from itself to* **R**.

*Network $\mathcal{N}$ is said to be in a Critical Combination if any of the following hold:*
*– Either $B_1$ or $B_2$ cut across all strong paths between* **S** *and* **R***s*
*– $B_1 \cup B_2$ cut across all weak paths between* **S** *and* **R**
*– $\exists W$ such that every weak path $p$ that avoids both $B_1$ and $B_2$ between* **S** *and* **R** *has a node, say $w$, that has both its adjacent edges (along $p$) directed inwards and $w \in W$ and the following hold:*
*– Both $B_1$ and $B_2$ are vertex cutsets between $w$ and* **R**. *In other words, every strong path from $w$ to* **R** *passes through both $B_1$ and $B_2$*
*– $\forall$ nodes in $(W \cup (Z \cap X))$, for $i \in \{0, 1\}$, $B_1$ is a vertex cutset to* **R** *in the edge-set $E_i \in \mathcal{TK}_G$, and $B_2$ is a vertex cutset to* **R** *in the edge-set $E_{\bar{i}} \in \mathcal{TK}_G$*

**Note 16** *Following Definition 15, network $\mathcal{N}$ can be divided into four components for a given two $t$-sized sets, $B_1$ and $B_2$ as follows:* **R**-*group, $B_1$, $B_2$ and the rest of the players together as one, say $C = \mathbb{P} \setminus (B_1 \cup B_2 \cup$* **R**-*group). Since the sender* **S** *and the receiver* **R** *are honest, clearly* $\mathbf{S} \in C$. **R**-*group has no knowledge of the actual graph $\mathcal{N}$, and each player in* **R**-*group has the same topology knowledge as that of the globally declared 2-sized $\mathcal{TK}_G$, made of two edge-sets $\{E_0, E_1\}$. Set $X \subset C$. Set $Z \subset C$. Players in $\mathbb{P} \setminus (X \cup$* **R**-*group) all have the same topology knowledge $\{E_0, E_1\}$.*

**Theorem 5.1** *A PRC protocol between* **S** *and* **R** *in the network $\mathcal{N} = (\mathbb{P}, \mathcal{E})$ with a 2-sized $\mathcal{TK}$ tolerating a Byzantine adversary characterized such that it may corrupt up to any $t$ players, exists if and only if Critical Combination does not occur in $\mathcal{N}$.*

*Proof. Necessity:* For Necessity,we must show that a network $\mathcal{N}$ that is in Critical Combination guarantees that no PRC protocol from **S** to **R** exists in $\mathcal{N}$. We now take each of the conditions described for a network to be in Critical Combination (Definition 15) and show that no PRC protocol can exist between S and R when the condition is true.

Note that if the $\mathcal{TK}'_i$ of all the players, following the inputs given in the Definition 15, is a singleton set, that is all the players have identified the actual graph $\mathcal{N}$ of the network, then the requirement for the PRC protocol is, as per [5], $(t+1)$-(**S**,**R**)-strong-connectivity and $(2t, t)$-(**S**,**R**)-special-connectivity. In fact, we get a better understanding of the *receiver* **R**-specificity of the definition of $(2t, t)$-(**S**,**R**)-special-connectivity, and we shall be showing that the knowledge of the actual graph or the absence of it for **R** is the key for the existence or non-existence of the PRC protocol.

The following four lemmas shall capture the requirement of our necessity proof.

**Lemma 5.1.1** *Following Definition 15, if either $B_1$ or $B_2$ cut across all strong paths between* $\mathbf{S}$ *and* $\mathbf{R}$ *in* $\mathcal{N}$ *, then a PRC protocol between* $\mathbf{S}$ *and* $\mathbf{R}$ *does not exist.*

*Proof:* It is obvious to see that in this case, an adversary that can corrupt up to any $t$ players can corrupt the set $B_1$ or $B_2$ that cuts across all strong paths between $\mathbf{S}$ and $\mathbf{R}$, and thereby disconnect the two in which case no PRC protocol can ever exist. ■

**Lemma 5.1.2** *Following Definition 15,if $B_1 \cup B_2$ cut across all weak paths between* $\mathbf{S}$ *and* $\mathbf{R}$*, then a PRC protocol between* $\mathbf{S}$ *and* $\mathbf{R}$ *does not exist.*

*Proof:* Refer to AppendixA. ■

**Lemma 5.1.3** *Following Definition 15, if $\exists W$ such that every weak path $p$ that avoids both $B_1$ and $B_2$ between* $\mathbf{S}$ *and* $\mathbf{R}$ *has a node, say $w$, that has both its adjacent edges (along $p$) directed inwards and $w \in W$ and both $B_1$ and $B_2$ are vertex cutsets between $w$ and* $\mathbf{R}$*, then a PRC protocol between* $\mathbf{S}$ *and* $\mathbf{R}$ *does not exist. In other words,if every strong path from $w$ to* $\mathbf{R}$ *passes through both $B_1$ and $B_2$, then a PRC protocol between* $\mathbf{S}$ *and* $\mathbf{R}$ *does not exist.*

*Proof:* Refer to AppendixA. ■

**Lemma 5.1.4** *Following Definition 15, if $\exists W$ such that every weak path $p$ that avoids both $B_1$ and $B_2$ between* $\mathbf{S}$ *and* $\mathbf{R}$ *has a node, say $w$, that has both its adjacent edges (along $p$) directed inwards and $w \in W$ and $\forall$ nodes in $(W \cup (Z \cap X))$, for $i \in \{0, 1\}$, $B_1$ is a vertex cutset to* $\mathbf{R}$ *in the edge-set $E_i \in \mathcal{TK}_G$, and $B_2$ is a vertex cutset to* $\mathbf{R}$ *in the edge-set $E_{\bar{i}} \in \mathcal{TK}_G$, then a PRC protocol between* $\mathbf{S}$ *and* $\mathbf{R}$ *does not exist.*

*Proof:* We now give the topology knowledge effect on the possibility of PRC through this proof. Notice that, in the network $\mathcal{N}$, $X \cap \mathbf{R}$-group $= \phi$, and therefore the entire $\mathbf{R}$-group has the same 2-sized $\mathcal{TK}'_i$. Further, the state of the network $\mathcal{N}$ following Definition 15 is such that players either know the actual edge-set or those players that have a doubleton set as their updated topology knowledge shall all have the same $\mathcal{TK}'_i$ which is set to the edge-sets in $\mathcal{TK}_G$. All nodes in $Z \cap X$ have complete knowledge of the topology of the network, and since they have semi-strong paths from themselves to $\mathbf{R}$, each of these players can share a common key with $\mathbf{R}$, through the head of the semi-strong paths, that is unknown to the adversary. Nodes in $W$ can have strong paths passing through one of $B_1$ or $B_2$, and since they are part of $\mathbf{S}$ to $\mathbf{R}$ weak paths of which some may be semi-strong paths, those that have semi-strong paths can share a key with $\mathbf{R}$ through the corresponding head of the semi-strong path. Each of these nodes can potentially *influence* $\mathbf{R}$ by exchanging information in secret without the knowledge of the adversary making use of their shared key. We shall be showing that in our case here, the presence of a common key between the players and $\mathbf{R}$ is rendered useless. Here is how:

Now, suppose the set $B_1$ were Byzantine corrupt in the edge-set $\mathcal{E}_0$ and $B_2$ were Byzantine corrupt in the edge-set $\mathcal{E}_1$ for two parallel executions of the same protocol. Assuming that the Byzantine faulty players act in a fail-stop fashion. $\mathbf{R}$ cannot distinguish between the case where the set $B_1$ blocks a message from any node, say $x$ in $(W \cup (Z \cap X))$ to $\mathbf{R}$ in the edge-set $\mathcal{E}_\alpha$ and the case where the set $B_2$ blocks a message from $x$ to $\mathbf{R}$ in $\mathcal{E}_{\bar{\alpha}}$ ($\alpha \in \{0, 1\}$) as $\mathbf{R}$ receives a null message in each case. Since all strong paths between $x$ to $\mathbf{R}$ are blocked, communication between $x$ and $\mathbf{R}$ fails.Further, $\mathbf{R}$ cannot identify which of the sets in $B_1$ and $B_2$ is corrupt, because both the scenarios appear equally possible to $\mathbf{R}$ given that it is not sure of the actual edge-set. Thus, there is no change in the state of the receiver $\mathbf{R}$, which in turn renders PRC impossible in this case. ■

This completes the necessity part of the proof for Theorem 5.1.

*Proof. Sufficiency:* For Sufficiency, Network $\mathcal{N}$ that is not in Critical Combination guarantees the existence of a PRC protocol from S to R in $\mathcal{N}$. So, we prove by giving a protocol and its proof of correctness.

Note 16 gives us the glimpse of the state of the network $\mathcal{N}$ following Definition 15.

Since there are n players, and $t$ can be corrupted, there are $\binom{|\mathbb{P}|}{t}$ options in front of the adversary,that is there are exactly $\binom{|\mathbb{P}|}{t}$ distinct ways of corrupting exactly $t$ players. Let each of the $\binom{|\mathbb{P}|}{t}$ distinct subsets of size $t$ be represented as $\{B_1, B_2, B_3, \ldots, B_{\binom{|\mathbb{P}|}{t}}\}$ where $B_i \subset \mathbb{P}$ and $|B_i| = t$. First, we show how to design a "PRC" sub-protocol assuming that the adversary is allowed to choose only from *two* of the $\binom{|\mathbb{P}|}{t}$ options that originally existed. In other words, we are only concerned about an adversary that may corrupt the players in the set $B_\alpha$ or the set $B_\beta$, where $1 \leq \alpha, \beta \leq \binom{|\mathbb{P}|}{t}$ and $\alpha \neq \beta$. Let us denote the resulting sub-protocol as $\Pi_{\alpha\beta}$. In the sequel, we show how to use all the sub-protocols $\Pi_{\alpha\beta}$ (there are clearly $\binom{\binom{|\mathbb{P}|}{t}}{2}$ of them) to design a grand protocol $\Pi$ that can be proved to be the required PRC protocol. In the Definition 15, the two $t$-sized sets $B_1$ and $B_2$ can be understood as the sets of players that an adversary may corrupt in one of the instances of $B_\alpha$ and $B_\beta$.

An honest player is the player not corrupted by the adversary. An honest path is understood as the path that avoids the sets $B_\alpha$ and $B_\beta$. An honest player in the network in possession of the actual edge-set behaves differently from an honest player which has a doubleton set. When the player knows $\mathcal{E}$, all their communication, that is, sending and receiving messages is along the lines of edges in $\mathcal{E}$ only. When the player has a doubleton set as its $\mathcal{TK}_i$, it sends messages along both, the edges in actual edge-set $\mathcal{E}$ and the other edge-set in its $\mathcal{TK}_i$. It is never sure which of its message is valid, because all communication along the false edges is lost, and the identity of the false edges is not with the player. Now, an honest player with a doubleton set as its $\mathcal{TK}_i$ accepts all messages that it receives which it identifies as valid, that is, those belonging to the edges in any of the two edge-sets in its $\mathcal{TK}_i$. All players corrupted by the adversary, w.l.o.g.,can be assumed to know $\mathcal{TK}$ and the actual edge-set $\mathcal{E}$. This is a modest assumption when dealing with an adversary. An honest player drops all messages from a player if it identifies that it is corrupted by the adversary.

*Designing the sub-protocol* $\Pi_{\alpha\beta}$*:* Critical Combination does not occur in network $\mathcal{N}$ and this implies all that all the conditions that cause critical combination are falsified. We see the consequences of the same here, and use this to design our sub-protocol.

Neither of $B_\alpha$ or $B_\beta$ cut across all strong paths between **S** and **R**. Since $B_\alpha$ and $B_\beta$ are the sets chosen by adversary one of which it can corrupt, there must be at least one honest strong path from **S** and **R** that does not pass through either $B_\alpha$ or $B_\beta$ in $\mathcal{N}$.

The deletion of both the sets $B_\alpha$ and $B_\beta$ from the network $\mathcal{N}$ does not cut across all weak paths between **S** and **R**. There must exist at least one honest weak path from **S** to **R** in $\mathcal{N}$ that avoids both the sets $B_\alpha$ and $B_\beta$.

We start with this honest weak path, say $p$. We consider the following two cases in the design of the sub-protocol $\Pi_{\alpha\beta}$:

Case (1) : *The path* $p$ *is such that* $w =$ **S***:* In this case, the path $p$ contains a player $y$ (which may be **S** or **R** too) such that $p$ is the combination of the strong path from $y$ to **S** and the strong path from $y$ to **R**. In other words, $y \in$(**S**-group $\cap$**R**-group). We know that **R**-group has the edge-sets $\{E_0, E_1\}$ as its topology knowledge. If **S**-group$\cap X \neq \phi$, then **S** would know the actual graph, else, it would have the same topology knowledge as **R**-group, $\{E_0, E_1\}$. We give the protocol for the case where both **S** and **R** do not have the actual graph $\mathcal{N}$ and are in possession of $\{E_0, E_1\}$, one of which is known to be $\mathcal{N}$, as per the Definition 7. The other case is similar and follows the same approach.

case (i): Notice that $y \in \mathbf{R}$-group, so even $y$ has $\{E_0, E_1\}$ as its topology knowledge. Each of these edge-sets is such that each has a weak path that does not pass through the two sets $B_\alpha$ and $B_\beta$. Let the path along the actual edge-set (w.l.o.g, say $E_0$) be $p$ and along $E_1$ be $p'$. In $p'$, we have a $y'$ which has a strong path from it to **S** and **R**. The state as defined in Note 16 is the same in both the edge-sets. Note that all players in $p$ and $p'$ are honest. The protocol that is run on one path $p$ is correspondingly replicated on $p'$. We give the protocol for $p$: First, $y$ sends to both **S** and **R**,along its both edge-sets, a message with two parts: one, a set of random keys, two, an array of signatures. Each player appends its signature to the second part of the message as it forwards the message to the next player in the path $p$. The random keys $K_1, K_2$ and $K_3$, along with the list of signatures of the players that the messages have seen, is sent to both **S** and **R** along the path $p$. **S, R** receive two sets of the same three keys along the actual edge-set, and $E_1$ from $y$ in $p$. Along $p'$, suppose the random keys sent by $y'$ be $K_1', K_2'$ and $K_3'$. If **S, R** receive these three keys along both the edge-sets $E_0$ and $E_1$, then they accept them as they cannot distinguish between the two edge-sets as to which is the correct one. **S,R** end up with two distinct sets of keys - ($K_1', K_2'$ and $K_3'$) and ($K_1, K_2$ and $K_3$). Next,**S** computes two values: $\psi$, $\psi_1$; two signatures: $\chi$,$\chi_1$; where $\psi = (M + K_1)$, $\chi = (K_2(M + K_1) + K_3)$ and $\psi_1 = (M + K_1')$, $\chi_1 = (K_2'(M + K_1') + K_3')$, and $M$ is the message that needs to be reliably transmitted. **S** sends two messages in each edge-set $E_0$ and $E_1$ [1] to **R** along *all* the vertex-disjoint strong paths each containing: a value ($\psi/\psi_1$), a signature($\chi/\chi_1$), and an array of signatures. Each player appends its signature to the array of signatures as it forwards the message to the next player in the path $p$ or correspondingly in $p'$. Now, **R** receives two values - two each of $\psi'$ and $\psi_1'$; two signatures - two each of $\chi'$ and $\chi_1'$ along two different paths as are given in each of the edge-sets $E_0$ and $E_1$. Notice that, **R** has knowledge of ($K_1, K_2$ and $K_3$) and ($K_1', K_2'$ and $K_3'$). Hence it can easily verify if $\chi' \stackrel{?}{=} K_2 * \psi' + K_3$ (correspondingly it verifies for $\chi_1'$). **R** reacts as follows: If the received value $\psi'$ has a valid signature ($\chi' = K_2 * \psi' + K_3$), then **R** outputs ($\psi' - K_1$) (correspondingly it outputs ($\psi_1' - K_1'$) in the other edge-set); furthermore, among all the received values, at least one of them is guaranteed to be valid (because at least one honest strong path exists!). The probability that **R** outputs the same message in both the edge-sets is high,namely $1 - \frac{1}{|\mathbb{F}|}$,which can be made $(1 - \delta)$ by suitably choosing $\mathbb{F}$.

Case (2): *The path $p$ is such that there are $k > 0$ players like $w$ ($w \neq$ **S** ), say $w_1, \ldots, w_k$ along $p$:* We will first consider the case when $k = 1$. For each of the subsequent cases ($k > 1$), we repeat the appropriate protocols given below on all $w_i$'s ($1 \leq i \leq k$) and in the sequel succeed in establishing reliable communication between **S** and **R** with a high probability. Owing to space constraints rest of the proof is given in AppendixC. ∎

## 6 Conclusion

We have provided a complete characterization for the problem of Probabilistic Reliable Communication given topology knowledge as a parameter. The significance of our results can be understood from the following implications: (a) *Generalization:* Our results are a strict generalization of the existing results for probabilistic reliable communication [5]; (b) *The "Randomization-effect"*: It is well known that for perfectly reliable communication, fault-tolerance is independent of nodes' knowledge of the network topology [7]; we show that in the case of probabilistic reliable communication, fault-tolerance is extremely sensitive to changes in the knowledge of network topology. (c) *Optimization*: Our results may be used to answer the question: *what is the optimal fault-tolerance that is achievable in reliable communication for a specified $\mathcal{TK}$ and vice-versa.*

[1] Note that **S** can verify if it has at least one honest strong path from it to **R** or not, and distinguish it with $E_1$

## References


[1] Paola Flocchini, Bernard Mans, and Nicola Santoro. Sense of direction in distributed computing. *Theor. Comput. Sci.*, 291(1):29–53, 2003.

[2] Paola Flocchini, Alessandro Roncato, and Nicola Santoro. Backward consistency and sense of direction in advanced distributed systems. In *PODC '99: Proceedings of the eighteenth annual ACM symposium on Principles of distributed computing*, pages 189–198, New York, NY, USA, 1999. ACM.

[3] M. Franklin and R. N. Wright. Secure Communication in Minimal Connectivity Models. In *Proceedings of Advances in Cryptology EUROCRYPT '98*, volume 1403 of *Lecture Notes in Computer Science (LNCS)*, pages 346–360. Springer-Verlag, 1998.

[4] M. Franklin and R.N. Wright. Secure Communication in Minimal Connectivity Models. *Journal of Cryptology*, 13(1):9–30, 2000.

[5] 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, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.

[6] 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.

[7] Lakshminarayanan Subramanian, Randy H. Katz, Volker Roth, Scott Shenker, and Ion Stoica. Reliable broadcast in unknown fixed-identity networks. In *PODC '05: Proceedings of the twenty-fourth annual ACM symposium on Principles of distributed computing*, pages 342–351, New York, NY, USA, 2005. ACM.

[8] DOLEV, D. The byzantine generals strike again. In Journal of Algorithms (1982), pp. 1430.

## A Proof of Lemma 5.1.2, Lemma 5.1.3

**Lemma A.0.5** *Following Definition 15,if $B_1 \cup B_2$ cut across all weak paths between* **S** *and* **R***, then a PRC protocol between* **S** *and* **R** *does not exist.*

Note 16 gives us the glimpse of the state of the network $\mathcal{N}$ following Definition 15.

It is evident from the definition of **R**-group that there do not exist vertices $u \in C$ and $v \in$ **R**-group, such that the edge $(u, v)$ is in $\mathcal{N}$. When $B_1 \cup B_2$ cut across all weak paths between **S** and **R**, there do not exist vertices $u \in C$ and $v \in$ **R**-group, such that the edge $(v, u)$ is in $\mathcal{N}$.

We will prove the impossibility even for the best case where every other edge (other than those between $C$ and R-group) exists and when every player knows the actual graph.

Define two executions $\mathbf{E}_0$ and $\mathbf{E}_1$ as follows. In both executions the vertices in **R**-group hold the random inputs $\{\rho_u | u \in \mathbf{R}\text{-group}\ \}$. In the execution $\mathbf{E}_\alpha \in \{\mathbf{E}_0, \mathbf{E}_1\}$, the Byzantine set $B_\alpha$ is corrupt and the message $m_\alpha$ is transmitted by **S**, the random inputs of the vertices in $(C \cup B_{\overline{\alpha}})$[2] are $\{\rho_u | u \in (C \cup B_{\overline{\alpha}})\}$. The behavior of the Byzantine set $B_\alpha$ in the execution $\mathbf{E}_\alpha$ is to send no message whatsoever to $C \cup B_{\overline{\alpha}}$ and to send to **R**-group exactly the same messages that are sent to **R**-group by the honest $B_\alpha$ in the execution $\mathbf{E}_{\overline{\alpha}}$. In order for the Byzantine set $B_\alpha$ to behave as specified in the execution $\mathbf{E}_\alpha$, the adversary needs to simulate the behavior of $(C \cup B_\alpha)$ in the execution $\mathbf{E}_{\overline{\alpha}}$. To achieve this task, the adversary simulates round-by-round the behavior of the vertices in $(C \cup B_\alpha)$ for the execution $\mathbf{E}_{\overline{\alpha}}$ using $\{\rho_u | u \in (C \cup B_\alpha)\}$ as the random inputs for the vertices in $(C \cup B_\alpha)$. At the beginning of each round, each simulated player has a history of messages that it got in the simulation of the previous rounds and its simulated local random input. The simulated player sends during the simulation the same messages that the honest player would send in the original protocol in the same state. The simulated messages that (players in) $B_\alpha$ sends to **R** are really sent by the players. All other messages are used only to update the history for the next round. The messages which are added to the history of each simulated vertex are the real messages that are sent by players in **R**-group and the simulated messages that are sent by the vertices in $(C \cup B_\alpha)$. No messages from $B_{\overline{\alpha}}$ are added to history. The history of messages of each simulated vertex in execution $\mathbf{E}_\alpha$ is the same as the history of the vertex in execution $\mathbf{E}_{\overline{\alpha}}$. Therefore, the messages sent by $B_1$ and $B_2$ to members of **R**-group in both executions are exactly the same and the members of **R**-group and in particular the receiver **R** receive and send the same messages in both executions. Thus, the receiver **R** cannot distinguish whether the set $B_1$ is corrupt and the message transmitted by **S** is $m_1$ or the set $B_2$ is corrupt and the message transmitted by **S** is $m_2$. Now, consider all the pairs of executions where the random inputs range over all possible values. In each pair of executions, whenever **R** accepts the correct message in one execution it commits an error in the other. Thus, for any strategy by **R** for choosing whether to receive $m_1$ or $m_2$ there is some $\alpha$ such that when $m_\alpha$ is transmitted, the receiver accepts $m_\alpha$ with probability at most $\frac{1}{2}$. ■

**Lemma A.0.6** *Following Definition 15, if $\exists W$ such that every weak path $p$ that avoids both $B_1$ and $B_2$ between* **S** *and* **R** *has a node, say $w$, that has both its adjacent edges (along $p$) directed inwards and $w \in W$ and both $B_1$ and $B_2$ are vertex cutsets between $w$ and* **R***, then a PRC protocol between* **S** *and* **R** *does not exist. In other words,if every strong path from $w$ to* **R** *passes through both $B_1$ and $B_2$, then a PRC protocol between* **S** *and* **R** *does not exist.*

*Proof:* Note 16 gives us a glimpse of the state of the network $\mathcal{N}$ following Definition 15.

In Lemma 5.1.2, we proved that when there are no weak paths between **S** and **R** that avoid $B_1$ and $B_2$, PRC protocol does not exist. We now show that in spite of the presence of multiple such

[2] We denote $\overline{1} = 2$ and viceversa.

weak paths between **S** and **R** that avoid $B_1$ and $B_2$, if they have a node of the type $w$, with both its edges inwards towards $w$ along the path, PRC protocol does not exist when both $B_1$ and $B_2$ are vertex cutsets between $w$ and **R**. We take the case where every player complete knowledge of the topology in spite of which PRC is shown to be impossible.

At least one edge from these weak paths must be from a node in **R**-group to another node in $C$ (since these are paths outside $(B_1 \cup B_2)$ and from **S** to **R**). We will show that removing that edge has no effect on the possibility of PRC thereby proving the required result.

Firstly, how can these edges be useful? The answer is that they can be used by players in **R**-group to send some secret messages to the players in $C$ such that the adversary, oblivious of these messages, cannot simulate the messages of $C$ without being distinguished by **R**-group. However, if we are able to show that no such secret information can help PRC from **S** to **R**, then we are through. We do the same now.

A node $x$ is said to have no influence on **R** if the output of **R** is independent of values sent by $x$. Otherwise $x$ is said to influence **R**. Consider an edge $(y, x)$ in $\mathcal{N}$ such that $y \in$ **R**-group and $x \in C$. We need to know whether $x$ can influence **R** by using the data received from $y$. Suppose we manage to show that it cannot then we are through since what it means is that data sent along the edge $(y, x)$ has no effect on **R** and hence can be ignored. We now proceed to prove the same.

Suppose that the node **R** can be influenced by $x$. This (at least) means that there must be a path $x, w_1, w_2, \ldots, w_q, \mathbf{R}$ in $\mathcal{N}$ such that $x$ transmits some information to $w_1$, then $w_1$ transmits some information to $w_2$ that depends on the information it got from $x$ and so on until some information gets to **R**.[3]

Given that every path from $x$ to **R** passes through some node(s) in $B_\alpha$ *followed* by some node(s) in $B_{\overline{\alpha}}$ for some $\alpha \in \{1, 2\}$, the adversary if it corrupts the $\alpha^{th}$ set in $\mathcal{A} = \{B_1, B_2\}$, does the following: let $w_j$ be the first vertex in $B_\alpha$ on a path from $x$ to **R**. The corrupt $w_j$ ignores the real messages that it gets from the players in $C \cup B_{\overline{\alpha}}$ and thus the messages that it sends do not depend on the message sent by $x$. Similarly, the messages sent by $x$ when $B_\alpha$ simulates the players in $C$ do not influence the messages it sends to **R** since the path from $x$ to **R** passes through at least one vertex from $B_{\overline{\alpha}}$ and no messages are sent by players in $B_{\overline{\alpha}}$ during the simulation. Thus even if **R** may know that the correct secret (that was exchanged using the edge $(y, x)$) was not used, he will not know which set in $\mathcal{A}$ to blame. Thus the simulated messages of $x$ have no influence on the messages received by **R** and can be ignored. Hence, the impossibility of PRC proved in Lemma 5.1.2 is not altered by using the edges from **R**-group to $C$. ∎

# B Proof of Theorem 3.1

**Theorem B.1** *A protocol* $\Pi$ *that solves a problem* $\pi$ *over a network* $\mathcal{N}$ *with* $N$*-sized* $\mathcal{TK}$ $(N > 2)$ *influenced by an adversary* $\mathcal{A}$ *exists if and only if, for each of the 2-sized topology knowledge* $(\mathcal{Z})$ *that can be derived from* $\mathcal{TK}$*, there exists a protocol solving* $\pi$ *with topology knowledge* $\mathcal{Z}$*.*

*Proof: Necessity:* We need to prove that if $\pi$ cannot be solved with one of the 2-sized topology knowledge $\mathcal{Z}$ derived from $\mathcal{TK}$ then $\pi$ cannot be solved with $\mathcal{TK}$. The proof is trivial following Note 5. If a protocol does not exist to solve a problem using a greater topology knowledge $\mathcal{Z}$, then it certainly cannot solve using lesser $\mathcal{TK}$.

[3]Since the network is synchronous, it may be possible to transmit information without actually sending message bits. However, even such transmissions are possible only between nodes that can actually exchange some message-bits as well. Thus, an information-path is necessarily a physical path too.

*Sufficiency:* We need to prove the statement: If there exists a protocol $\Pi_j$ that can solve the problem $\pi$ for the $j^{th}$ 2-sized topology knowledge $\mathcal{Z}_j$ derived from $\mathcal{TK}$, then there exists a protocol $\Pi$ that can solve $\pi$ with topology knowledge $\mathcal{TK}$. We give a proof by induction on the size of $\mathcal{TK}_i$.
*Base Case:* In this case, $N = 3$. Let $\mathcal{TK}_i = \{\mathcal{E}^i_1, \mathcal{E}^i_2, \mathcal{E}^i_3\}$, $1 \le i \le |\mathbb{P}|$. Let $\Pi_1$ be the protocol that solves $\pi$ using the topology knowledge $\mathcal{Z}_1$ where $\mathcal{Z}_1$ differs from $\mathcal{TK}$ only with respect to player 1, namely, in $\mathcal{Z}_1$ the topology knowledge of player 1 is set as $\mathcal{Z}_{11} = \{\mathcal{E}^1_1, \mathcal{E}^1_2\}$ and let $\Pi_2$ be the protocol that solves $\pi$ using the topology knowledge $\mathcal{Z}_2$ where $\mathcal{Z}_2$ differs from $\mathcal{TK}$ only with respect to player 1, namely, in $\mathcal{Z}_2$ the topology knowledge of player 1 is set as $\mathcal{Z}_{12} = \{\mathcal{E}^1_1, \mathcal{E}^1_3\}$ and let $\Pi_3$ be the protocol that solves $\pi$ using the topology knowledge $\mathcal{Z}_3$ where $\mathcal{Z}_3$ differs from $\mathcal{TK}$ only with respect to player 1, namely, in $\mathcal{Z}_3$ the topology knowledge of player 1 is set as $\mathcal{Z}_{13} = \{\mathcal{E}^1_2, \mathcal{E}^1_3\}$. Notice that using above three protocols $\Pi_1, \Pi_2$ and $\Pi_3$, the problem $\pi$ can be solved even if the player 1 is not aware of the actual topology. Specifically, all the players run all the three protocols $\Pi_1, \Pi_2$ and $\Pi_3$ and obtain three outputs $O_1, O_2$ and $O_3$ respectively. Note that the actual graph is part of all the individual player's topology knowledge in at least two of the three cases. Thus, at least two of the three outputs must be same and equal to the output that a protocol solving $\pi$ would produce. Thus, every player can take the majority of the three outputs and thereby solve $\pi$. Thus, a protocol for $\pi$ with topology knowledge $\mathcal{TK}$ exists if and only if a protocol for solving $\pi$ exists with each of the two valid topology knowledges (i.e. those that contain the actual edge-set) among $\mathcal{Z}_{11}, \mathcal{Z}_{12}$ and $\mathcal{Z}_{13}$.

Note that in a valid $\mathcal{Z}_{jk}$, the number of elements in the first player's topology knowledge is two (and not three). Now, we repeat the idea for *player 2*; that is, we create three more topology knowledges from each of the three $\mathcal{Z}_{jk}$'s with the second player's topology knowledge split into three parts such that each edge-set occurs in two of them. This would imply that a protocol for $\pi$ with topology knowledge $\mathcal{TK}$ exists if and only if a protocol for solving $\pi$ exists with (several) valid topology knowledges with the size of the topology knowledges of the players 1 and 2 being of size two. Repeating this process of all the $|\mathbb{P}|$ players, we can say that if some protocol solves a problem for each of its 2-sized $\mathcal{TK}$s derived from 3-sized $\mathcal{TK}$, then $\Pi$ solves the problem with 3-sized $\mathcal{TK}$.
*Induction Hypothesis:* Let us suppose that the statement is true for up to any $m$-sized $\mathcal{TK}$. That is to say, If $\pi$ is solvable for each of its 2-sized $\mathcal{TK}$s derived from the $m$-sized $\mathcal{TK}$, then $\pi$ can be solved with topology knowledge $\mathcal{TK}$.
*Induction:* Let $\mathcal{TK}_i$ be $m+1$-sized $\mathcal{TK}_i$. Since $m + 1 > 3$, we can partition $\mathcal{TK}_i$ into 3 sets, say $A, B, C$ each of which is of size less than $m + 1$ such that every element in $\mathcal{TK}_i$ occurs in two among $A, B$ and $C$. From our induction hypothesis, we know that $\pi$ can be solved with topology knowledge $X_A$ (which differs from at player $i$'s topology knowledge, namely $\mathcal{TK}_i$ is replaced with $A$) if it can be solved for each of the 2-sized topology knowledges derived from $X_A$. Similarly for $X_B$ and $X_C$. Also, if $\pi$ is solved over two of the valid topology knowledges among $X_A$, $X_B$ and $X_C$, then it can be solved with $\mathcal{TK}$ too (by majority voting). Note that in each of the two valid topology knowledges among $X_A$, $X_B$ and $X_C$, the size of topology knowledge of player $i$ is less than $m$. Repeating the above process across all players, we can bring down the size to the topology knowledge to less than $m$ for all players.

Thus, the statement is true for an $m+1$-sized $\mathcal{TK}$.Therefore, by induction, it is true $\forall m \in \mathbb{N}$.■

# C Sufficiency Proof of Theorem 5.1

*Proof. Sufficiency:* For Sufficiency, Network $\mathcal{N}$ that is not in Critical Combination guarantees the existence of a PRC protocol from S to R in $\mathcal{N}$. So, we prove by giving a protocol and its proof of correctness.

Note 16 gives us the glimpse of the state of the network $\mathcal{N}$ following Definition 15.

Since there are n players, and $t$ can be corrupted, there are $\binom{|\mathbb{P}|}{t}$ options in front of the adversary,that is there are exactly $\binom{|\mathbb{P}|}{t}$ distinct ways of corrupting exactly $t$ players. Let each of the $\binom{|\mathbb{P}|}{t}$ distinct subsets of size $t$ be represented as $\{B_1, B_2, B_3, \ldots, B_{\binom{|\mathbb{P}|}{t}}\}$ where $B_i \subset \mathbb{P}$ and $|B_i| = t$. First, we show how to design a "PRC" sub-protocol assuming that the adversary is allowed to choose only from *two* of the $\binom{|\mathbb{P}|}{t}$ options that originally existed. In other words, we are only concerned about an adversary that may corrupt the players in the set $B_\alpha$ or the set $B_\beta$, where $1 \leq \alpha, \beta \leq \binom{|\mathbb{P}|}{t}$ and $\alpha \neq \beta$. Let us denote the resulting sub-protocol as $\Pi_{\alpha\beta}$. In the sequel, we show how to use all the sub-protocols $\Pi_{\alpha\beta}$ (there are clearly $\binom{\binom{|\mathbb{P}|}{t}}{2}$ of them) to design a grand protocol $\Pi$ that can be proved to be the required PRC protocol. In the Definition 15, the two $t$-sized sets $B_1$ and $B_2$ can be understood as the sets of players that an adversary may corrupt in one of the instances of $B_\alpha$ and $B_\beta$.

An honest player is the player not corrupted by the adversary. An honest path is understood as the path that avoids the sets $B_\alpha$ and $B_\beta$. An honest player in the network in possession of the actual edge-set behaves differently from an honest player which has a doubleton set. When the player knows $\mathcal{E}$, all their communication, that is, sending and receiving messages is along the lines of edges in $\mathcal{E}$ only. When the player has a doubleton set as its $\mathcal{TK}_i$, it sends messages along both, the edges in actual edge-set $\mathcal{E}$ and the other edge-set in its $\mathcal{TK}_i$. It is never sure which of its message is valid, because all communication along the false edges is lost, and the identity of the false edges is not with the player. Now, an honest player with a doubleton set as its $\mathcal{TK}_i$ accepts all messages that it receives which it identifies as valid, that is, those belonging to the edges in any of the two edge-sets in its $\mathcal{TK}_i$. All players corrupted by the adversary, w.l.o.g.,can be assumed to know $\mathcal{TK}$ and the actual edge-set $\mathcal{E}$. This is a modest assumption when dealing with an adversary. An honest player drops all messages from a player if it identifies that it is corrupted by the adversary.

*Designing the sub-protocol* $\Pi_{\alpha\beta}$*:* Critical Combination does not occur in network $\mathcal{N}$ and this implies all that all the conditions that cause critical combination are falsified. We see the consequences of the same here, and use this to design our sub-protocol.

Neither of $B_\alpha$ or $B_\beta$ cut across all strong paths between **S** and **R**. Since $B_\alpha$ and $B_\beta$ are the sets chosen by adversary one of which it can corrupt, there must be at least one honest strong path from **S** and **R** that does not pass through either $B_\alpha$ or $B_\beta$ in $\mathcal{N}$.

The deletion of both the sets $B_\alpha$ and $B_\beta$ from the network $\mathcal{N}$ does not cut across all weak paths between **S** and **R**. There must exist at least one honest weak path from **S** to **R** in $\mathcal{N}$ that avoids both the sets $B_\alpha$ and $B_\beta$.

We start with this honest weak path, say $p$. We consider the following two cases in the design of the sub-protocol $\Pi_{\alpha\beta}$:

Case (1) :*The path* $p$ *is such that* $w = \mathbf{S}$*:* In this case, the path $p$ contains a player $y$ (which may be **S** or **R** too) such that $p$ is the combination of the strong path from $y$ to **S** and the strong path from $y$ to **R**. In other words, $y \in$(**S**-group $\cap$**R**-group). We know that **R**-group has the edge-sets $\{E_0, E_1\}$ as its topology knowledge. If **S**-group$\cap X \neq \phi$, then **S** would know the actual graph, else, it would have the same topology knowledge as **R**-group, $\{E_0, E_1\}$. We give the protocol for the case where both **S** and **R** do not have the actual graph $\mathcal{N}$ and are in possession of $\{E_0, E_1\}$, one of which is known to be $\mathcal{N}$, as per the Definition 7. The other case is similar and follows the same approach.

case (i): Notice that $y \in$ **R**-group, so even $y$ has $\{E_0, E_1\}$ as its topology knowledge. Each of these edge-sets is such that each has a weak path that does not pass through the two sets $B_\alpha$ and $B_\beta$. Let the path along the actual edge-set (w.l.o.g, say $E_0$) be $p$ and along $E_1$ be $p'$. In $p'$, we have a $y'$ which has a strong path from it to **S** and **R**. The state as defined in Note 16 is the same in both the edge-sets. Note that all players in $p$ and $p'$ are honest. The protocol that is run on one path $p$ is correspondingly replicated on $p'$. We give the protocol for $p$: First, $y$ sends to both **S** and **R**,along its both edge-sets, a message with two parts: one, a set of random keys, two, an array of signatures. Each player appends its signature to the second part of the message as it forwards the message to the next player in the path $p$. The random keys $K_1, K_2$ and $K_3$, along with the list of signatures of the players that the messages have seen, is sent to both **S** and **R** along the path $p$. **S, R** receive two sets of the same three keys along the actual edge-set, and $E_1$ from $y$ in $p$. Along $p'$, suppose the random keys sent by $y'$ be $K'_1, K'_2$ and $K'_3$. If **S, R** receive these three keys along both the edge-sets $E_0$ and $E_1$, then they accept them as they cannot distinguish between the two edge-sets as to which is the correct one. **S,R** end up with two distinct sets of keys - ($K'_1, K'_2$ and $K'_3$) and ($K_1, K_2$ and $K_3$). Next,**S** computes two values: $\psi$, $\psi_1$; two signatures: $\chi$,$\chi_1$; where $\psi = (M + K_1)$, $\chi = (K_2(M + K_1) + K_3)$ and $\psi_1 = (M + K'_1)$, $\chi_1 = (K'_2(M + K'_1) + K'_3)$, and $M$ is the message that needs to be reliably transmitted. **S** sends two messages in each edge-set $E_0$ and $E_1$ [4] to **R** along *all* the vertex-disjoint strong paths each containing: a value ($\psi/\psi_1$), a signature($\chi/\chi_1$), and an array of signatures. Each player appends its signature to the array of signatures as it forwards the message to the next player in the path $p$ or correspondingly in $p'$. Now, **R** receives two values - two each of $\psi'$ and $\psi'_1$; two signatures - two each of $\chi'$ and $\chi'_1$ along two different paths as are given in each of the edge-sets $E_0$ and $E_1$. Notice that, **R** has knowledge of ($K_1, K_2$ and $K_3$) and ($K'_1, K'_2$ and $K'_3$). Hence it can easily verify if $\chi' \stackrel{?}{=} K_2 * \psi' + K_3$ (correspondingly it verifies for $\chi'_1$). **R** reacts as follows: If the received value $\psi'$ has a valid signature ($\chi' = K_2 * \psi' + K_3$), then **R** outputs ($\psi' - K_1$) (correspondingly it outputs ($\psi'_1 - K'_1$) in the other edge-set); furthermore, among all the received values, at least one of them is guaranteed to be valid (because at least one honest strong path exists!). The probability that **R** outputs the same message in both the edge-sets is high,namely $1 - \frac{1}{|\mathbb{F}|}$,which can be made $(1 - \delta)$ by suitably choosing $\mathbb{F}$.

Case (2): *The path $p$ is such that there are $k > 0$ players like $w$ ($w \neq$ **S** ), say $w_1, \ldots, w_k$ along $p$:* We will first consider the case when $k = 1$. For each of the subsequent cases ($k > 1$), we repeat the appropriate protocols given below on all $w_i$'s ($1 \leq i \leq k$) and in the sequel succeed in establishing reliable communication between **S** and **R** with a high probability. Owing to space constraints rest of the proof is given in AppendixC.

Since we start with the assumption that conditions on Definition 15 are falsified, note that every strong path from $w$ to **R** does not pass through both $B_\alpha$ and $B_\beta$ for a $w \in W$ that is on the honest weak path $p$ from $w_1$ to **R**. That is, there must exist a strong path $Q$ from $w_1$ to **R** that does not pass through nodes in either the set $B_\alpha$ or the set $B_\beta$.

Recall that $p$ must contain a node $y$ (which may be **R**) such that there is strong path from $y$ to $w_1$ (along $p$) and there is a strong path from $y$ to **R** (also along $p$). In other words, $y \in (w_1 - group \cap \mathbf{R} - group)$. We know that **R**-group has the edge-sets $\{E_0, E_1\}$ as its topology knowledge. If $w_1$-group$\cap X \neq \phi$, then $w_1$ would know the actual graph, else, it would have the same topology knowledge as **R**-group, $\{E_0, E_1\}$. The protocol we give below is in two parts. Part I deals with the protocols to for communication between $w_1$ and **R** for each of the four cases given above. Part II deal with how, from Part I, we move on to ensure reliable communication between **S** and **R** with a very high probability.

Part I: *Protocols for communication between $w_1$ and* **R***:*

case (i): We give the protocol for the case where both $w_1$ and **R** do not have the actual graph $\mathcal{N}$ and are in possession of $\{E_0, E_1\}$, one of which is known to be $\mathcal{N}$, as per the Definition 7. The other case is similar and follows the same approach. The protocol for $w_1, y$,**R** is similar to the case (i) in Case 1 above till each of $w_1$, **R** end up with two distinct sets of keys - ($K'_1, K'_2$ and $K'_3$) and

[4] Note that **S** can verify if it has at least one honest strong path from it to **R** or not, and distinguish it with $E_1$

($K_1, K_2$ and $K_3$) just as **S,R** do.

Next, $w_1$ computes two values: $\psi$, $\psi_1$; two signatures: $\chi$,$\chi_1$; where $\psi = (M_{w_1} + K_1)$, $\chi = (K_2(M_{w_1} + K_1) + K_3)$ and $\psi_1 = (M_{w_1} + K_1')$, $\chi_1 = (K_2'(M_{w_1} + K_1') + K_3')$, and $M_{w_1}$ is the message from $w_1$ that is to be reliably transmitted to **R**. Let the path along the $E_0$ be $Q$ and along $E_1$ be $Q'$. $w_1$ sends two messages in each edge-set ($E_0$ and $E_1$) to **R** along $Q$ and $Q'$ each containing: a value ($\psi/\psi_1$), a signature($\chi/\chi_1$), and an array of signatures. Each player appends its signature to the array of signatures as it forwards the message to the next player in the path $Q$ or correspondingly in $Q'$. Now, **R** receives two values - two each of $\psi'$ and $\psi_1'$; two signatures - two each of $\chi'$ and $\chi_1'$ along two different paths as are given in each of the edge-sets $E_0$ and $E_1$. **R** verifies using both the set of keys in its possession. Using these, it can easily verify if $\chi' \stackrel{?}{=} K_2 * \psi' + K_3$. It verifies on all combinations of $\psi'$,$\psi_1'$,$\chi'$ and $\chi_1'$. **R** reacts as follows: If the received value $\psi'$ has a valid signature on at least one of the combination (say $\chi' = K_2 * \psi' + K_3$), then **R** outputs $(\psi' - K_1)$; else (that is if either the signature is invalid($\chi' \neq K_2 * \psi' + K_3$) or the original message is not received), **R** *knows* the identity (among the two possibilities of $\alpha$ or $\beta$) of the set that is the corrupt set.

Since we start with the assumption that conditions on Definition 15 are falsified, we note that $\forall$ nodes in $(W \cup (Z \cap X))$, for $i \in \{0, 1\}$, $B_1$ is not a vertex cutset to **R** in the edge-set $E_i \in \mathcal{TK}_G$, and $B_2$ is not a vertex cutset to **R** in the edge-set $E_{\overline{i}} \in \mathcal{TK}_G$, the path $Q$ completely avoids the players from one of these sets say $B_j$, $j \in \{1, 2\}$; This clearly means that a faulty path $Q$ (since a wrong message was delivered) entails that set $B_{\overline{j}}$ is corrupt (where $\overline{j} = \{1, 2\} - \{j\}$).

In Case (2) and its sub-cases above, if the set $B_j$, $j \in \{1, 2\}$, is not corrupt (which means that the other set may be corrupt), then **R** receives the correct message with certainty while the adversary has no information about the message.On the other hand, if the set $B_j$ is corrupt, then though the adversary still has no information about the transmitted message, he has complete control over **R**'s output. **R**'s output could therefore either be a valid message or a null message with the knowledge that (any subset of) $B_j$ is corrupt. But, if **R** receives a valid message, it is the correct message with a very high probability.

Protocols for the four cases in Part I aim at one of the following: (a) Simulating a direct edge (in other sense, having a strong path that passes only through honest players) between $w_1$ and **R** so that message from $w_1$ can be successfully communicated to **R** (or) (b) Simulation of the direct edge fails, and **R** identifies which of the two sets in $B_\alpha$, $B_\beta$ is corrupt.

Part II: If a protocol in Part I succeeds in (a) above, then depending on whether $w_1$ (and thereby all such $w_i$'s) belongs to **S**-group or vice-versa, that is, $w_1 \in$ **S**-group or **S** $\in w_1$-group we have two cases. Note that, there will exist $w_i$'s where neither $w_i \in$ **S**-group nor **S** $\in w_i$-group. It is precisely for this reason that we repeat the appropriate protocols given below on all $w_i$'s ($1 \leq i \leq k$) so as to arrive at a case where one of these two cases occurs.

If $w_1 \in$ **S**-group, then, there exists a direct path from $w_1$ to **S**. From the protocol in Part I, there is a direct path from $w_1$ to **R**. That is, $w_1 \in$ **R**-group, which implies that $w_1 \in$ (**S**-group $\cap$ **R**-group). Notice that $w_1$ is similar to the player $y$ in path $p$ in Case (1) of the sub-protocol $\Pi_{\alpha\beta}$. So, in this case, we follow the protocol in Case (1) to establish reliable communication between **S** and **R**. If **S** $\in w_1$-group, then **S** sends the message that it wants to reliably communicate to **R** through the path between $w_1$ and **R**. Since the path is secure, and passes only through honest players, reliable communication takes place.

If a protocol in Part I succeeds in (b) above, then we do the following: Since there exists at least one honest strong path from **S** to **R**, it must avoid $B_j$. **S** sends the message along its both edge-sets which has two parts: the message $M$ and an array of player indices. Each player appends its index to the array of player indices as it forwards the message to the next player along all these paths to **R**. The knowledge that $B_j$ is corrupt is sufficient for **R** to recover the correct message passing through the honest path.

Note that this gives us the protocol for our uniform topology knowledge built with the help of $\mathcal{TK}_G$s. To give the protocol for a given 2-sized $\mathcal{TK}$, we use the means shown in the *If part* proof of the Theorem 4.1 given in Section 4.

This completes our exercise of constructing the sub-protocol $\Pi_{\alpha\beta}$ that is guaranteed to work correctly only if one of $B_\alpha$ or $B_\beta$ is chosen by the adversary.

Note that **R** can simulate the sub-protocol $\Pi_{\alpha\beta\gamma}$ which assumes that one among the *three* sets $B_\alpha$ or $B_\beta$ or $B_\gamma$ is chosen by the adversary. The simulation is done as follows: **R** takes the majority among the outputs of the three protocols $\Pi_{\alpha\beta}$, $\Pi_{\beta\gamma}$ and $\Pi_{\alpha\gamma}$. A majority is bound to exist since any set chosen by the adversary is tolerated in *two* of the three protocols. Next, $R$ can simulate the sub-protocol which behaves like a PRC protocol as long as any one among a collection of *four* sets is chosen by the adversary. Continuing further, **R** will be able to simulate the protocol that behaves correctly if one among the collection of $\binom{n}{t}$ sets is chosen by the adversary. This protocol by definition is the PRC protocol from **S** to **R**! We conclude the sufficiency part of the proof.