RESEARCH STATEMENT
TOWARDS ROBUST DISTRIBUTED DYNAMIC NETWORKS

AMITABH TREHAN
LECTURER, HIGH PERFORMANCE AND DISTRIBUTED COMPUTING
SCHOOL OF ELECTRONICS, ELECTRICAL ENGINEERING AND COMPUTER SCIENCE,
QUEEN'S UNIVERSITY BELFAST, UK
A.TREHAN@QUB.AC.UK

INTRODUCTION

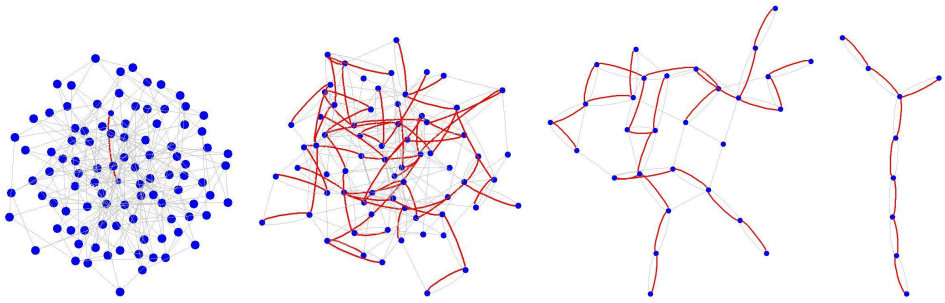
Most things of interest in life exhibit change. Yet, it is the continuity and the patterns in that change that we seek and try to understand. The same is true for most modern networks and systems of loosely interacting components. It is important not just philosophically but from scientific, engineering and economic point of view to understand this dynamicity. The discipline of distributed computing and algorithms is concerned with what can be achieved by interacting agents with limited local knowledge. This vibrant field has a plethora of models and viewpoints on addressing dynamicity. Are there unified models and approaches that bring us a deeper understanding and better engineering designs to accommodate dynamicity in networks? Can we build 'intelligent' systems that automatically recover from failures and errors? Can they self-heal if the failures are fail-stop/hard, or self-stabilize if the failures are transient/soft, or perform meaningful and scalable distributed computation if the failures are byzantine? What levels of fault-tolerance can be achieved? How do newer paradigms help in developing more robust solutions? Are there distributed algorithms to design distributed systems? Does game theory provide insight into such composition processes? Such questions have been driving my recent research. My work has immediate and long term practical implications improving the design, particularly, resilience of overlay and P2P networks, of Exascale high performance computing systems (vis AllScale [37]) and the upcoming *Internet of Things* [2] and *Software Defined Networking* [3, 27, 38].

In what follows, I discuss my research and future vision with a broad classification into (interacting) catregories.

SELF-HEALING AND DYNAMIC NETWORKS

With the ever increasing ubiquitousness of the Internet, mobile, wireless, ad-hoc and P2P networks, there is an urgent need for algorithms for these systems and to address what can and cannot be done on such networks. In my Ph.D. dissertation [35], we formalized and proposed the self-healing model for dynamic networks. In this model, a powerful adversary can insert or remove nodes while the algorithm can add (usually local) edges with the aim of maintaining certain desired invariants within acceptable bounds (this can be informally referred to as 'self-healing'). In a rich and fruitful line of research, we have proposed a number of increasingly sophisticated algorithms that maintain local and global network topological properties (often having competing requirements) with only local changes. Our work mainly deals with constructing and maintaining graph substructures, which is often a difficult problem. Our algorithms are responsive, thus avoiding redundant components, efficient in terms of time and messages, and often optimal i.e. with matching lower bounds. As an illustration, Figure 1 shows the snapshots of a network executing one of our algorithms. The red(dark) edges are the new edges added by the algorithm. Notice that the network stays connected despite repeated attack.

Our algorithms have not only used well known structures and techniques like bounded-degree expanders and spectral analysis but we have also devised new innovative data structures such as half-full trees (described

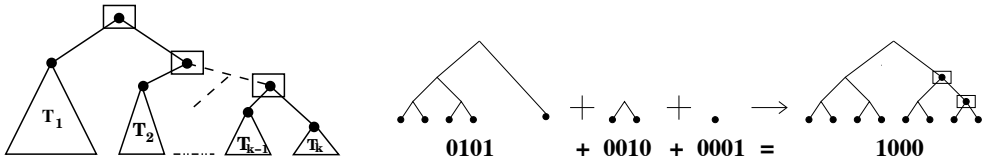


(a) single deletion (b) 30 deletions (c) 70 deletions (d) 90 deletions

FIGURE 1. A timeline of deletions and self healing in a network with 100 nodes. The gray edges are the original edges and the red edges are the new edges added by our self-healing algorithm.

later). Some of our algorithms also use the idea of virtual graphs (graphs with nodes simulated by real nodes) - this approach is more formally discussed in [36]. Outlines of some of our algorithms follows:

- FORGIVINGGRAPH [10, 11]: ForgivingGraph efficiently maintains a general graph of the network, handling both deletions and insertions, while guaranteeing at worst a constant multiplicative degree increase and the simultaneously challenging property of a low ($\log n$) stretch (maximum multiplicative distance increase between any two nodes). Also, we introduced a novel mergable data structure called half-full trees(*hafts*) having a one-to-one correspondence to binary numbers, with the merge corresponding to binary addition. This is illustrated in Figures 2(a) and 2(b).



(a) Structure of a haft. Each T_i is a complete binary tree, with $|T_1| > |T_2| > \cdots > |T_k|$. The spine nodes (nodes outside the complete trees) are shown boxed.

(b) Merging three hafts. The vertices in the square boxes are the new isolated vertices used to join the complete trees. Merging is analagous to binary number addition, where the number of leaves are represented as binary numbers.

FIGURE 2. Half-full trees (hafts)

- *DEX and DConstructor: Resilient Distributed Expander Constructions* [30, 32]*:* A series of my work deals with distributed construction and maintenance of sparse expanders which is an extremely useful topology e.g for P2P networks. DEX is our work on construction and maintainance of deterministic distributed expander graphs[1] in the self-healing model [29]. Having such an algorithm not only has immediate impact on many algorithms that use expander construction as a building block such as Xheal [31], but also extends additional functionality to existing popular overlay networks such as Chord [33] and Re-Chord [17]. Only a few, and that too, randomized distributed construction of expander graphs are known [9, 6, 26]. Therefore, in our knowledge, this is the first deterministic distributed expander graph construction and maintenance algorithm. In work under progress, we have developed algorithms (*DConstructor*) that can take almost any initial P2P topology and convert it resiliently into an expander.

[1]here, by deterministic expanders, we imply that the expansion properties hold deterministically, not just *with high probability*

- *Compact Routing Messages in Self-Healing Trees of low memory nodes* [2]*:* Our latest work is a first term in the direction of self-healing 'computations' or running reliable computations on faulty networks with the aid of self-healing. Not only that, our methods are compact (i.e. use only $O(polylog\, n)$ memory. This makes them applicable on nodes with low memory ($O(polylog\ n)$) and thus, on networks such as *mobile networks* and to future networks such at the *Internet of Things*. This work is a combination of a variant of a tree based routing of Thorup-Zwick [34] and a compact version of ForgivingTree [12] that we have developed and allows messages to be delivered from a sender to a live receiver despite node failures.

We believe our model and work so far has a lot to contribute in the near future:

(1) *A unified understanding of dynamicity:* As one possible extension addressing fundamental issues, introducing even a bit of dynamicity (e.g. changing nodes and egdes in a graph) makes many problems far more difficult, be it in the static or distributed setting. In distributed algorithms, there are many models addressing dynamicity with various assumptions such as Kuhn, Lynch and Oshman's *Dynamic Graph* model [20], Kutten and Korman's (from Afek, Awerbuch, Plotkin and Saks [1]) dynamic Controller model [18] and our self-healing model. I seek to explore these models, their limits and possible interactions with a view towards developing a deeper understanding of dynamicity in distributed networks.

(2) *AllScale: Self-healing and Self-Stabilizing Exascale Systems* [37]*:* As a Co-Investigator on the EU H2020 Future and Emerging Technology grant *AllScale*, I am driving the efforts to incorporate resilience in future exascale high performance systems. Exascale systems will do billion billion calculations per second approaching the processing power of the human brain. This resilience is envisaged byprotecting against both soft faults (transient memory faults) by self-stabilisation at the application level and by protecting against hard faults (node crashes) by self-healing at both the application and run-time/architectural level.

(3) *Self-healing Software Defined Networks and Streaming Data Analytics:* SDN is a new paradigm that provides a very important use case for self-healing. SDN is commercially important (with investments from all major networking companies) and with its virtualisation and logical separation ideas, a ripe environment for augmentation with robust self-healing. We have already begun preliminary work on this [3]. In fact, our ideas led to me winning the prestigious **Newton Incoming Fellowship** of the Royal Society for a proposal based on self-healing SDN. Sadly, I had to decline the fellowship since it conflicted with the terms of my faculty position. Our research plan centres around developing robust self-healing SDN working in dynamic fault-prone environments with streaming data analytics as our main demonstration application. Streaming data analytics is an application domain which is really important at the moment due to the amount of data (big data) being generated and analysed in real time e.g. in social networks.

ROBUST DISTRIBUTED COMPUTATION AND ALGORITHMS

A vitally important question is carrying out distributed computation in different adversarial and dynamic settings:

- *Efficient Leader Election Algorithms:* We have developed message and time efficient algorithms for leader election and proved fundamental lower bounds. For certain topologies, if the nodes do not need to know the identity of the leader, we show that we can solve leader election with sublinear number of messages and low (constant) time (our work has won the **best paper award** at *ICDCN 2013* [23] (Journal version: [24])). In further work the same year we showed several fundamental lower bounds for randomised and distributed leader election on general graphs (many of which were almost folklore but never proven). We also gave what we suspect is the best deterministic LE algorithm that simultaneously bounds time and messages. This paper was presented at PODC [22] and was only one of two invited directly to the *journal of the ACM* where it has been accepted and will be published in 2015.

- *Scalable Byzantine Agreement through quorum building:* In [13], we addressed the problem of designing distributed algorithms for large scale networks that suffer from byzantine faults. Byzantine Agreement is a basic building block for such algorithms. Following on from earlier works of King, Saia and co-authors [14, 15, 16], we devised the first scalable load balanced byzantine agreement algorithm in the full information model. An important technique we invented is a quick and load balanced method of developing a *good quorum*, which is a set of $O(\log n)$ processors that contains a representative fraction of good processors. We believe we have developed a powerful tool and are now looking at applying it for addressing other problems.
- *Approximate densest graphs in an edge-dynamic model:* In an edge dynamic model (edges are inserted or deleted), we developed an algorithm [4], in which nodes continually estimate the graph density to determine if they belong to a dense subgraph (or to an at-least-k-dense subgraph for some k) (i.e the nodes are **self-aware**). We also give the first distributed algorithms for dense subgraph approximation for these problems (the at-least-k-dense subgraph problem is known to be NP-hard in the centralized setting). We hope to extend our work to design better self-awareness and estimation algorithms in dynamic scenarios.
- *Verification and Distributed Complexity:* While complexity theory (and the related complexity classes) is very well established for centralized algorithms, its development is still at a very early stage in Distributed Computing. Recent work with labeling schemes [19] and LOCAL model complexity [8] offers a promising direction. We are looking at further applications of such verification schemes, and using them to develop a sound distributed complexity theory.
- *Robust Analytic Queries in in-memory databases:* We are investigating (with a Ph.D. student of mine) the possibility of developing a robust ('self-healing') analytic query system for in-memory databases using a novel idea of query checkpoints. This work is potentially commercially important as companies like SAP are heavily invested into in-memory databases and analytic queries is a major part of their business. This research aims to minimise the loss that occurs if a failure happens during analytic queries which are queries which run for a long period of time and, therefore, a failure can cause loss of a major amount of work and costly re-execution.

GAME THEORY, ECONOMICS AND COMPUTATION

What game theoretic protocols can lead rational and selfish agents to form 'high quality' consortiums (groups) (say, in distributed systems)? Can protocols be developed for better allocation of funding grants?

- *EU grant games:* In our AAAI paper [21], a work which was supported by the *Technion-Microsoft Electronic-Commerce Center*, we have a setting where agents (each with an individual value) form consortiums in order to compete for grants from a funding agency. Our work suggests a direction towards addressing important real world questions such as the best way to allocate research funds. Here, we would like to paraphrase the following comment about our work (by Prof. S. Muthukrishnan, of Rutgers, in his blog [28]:

  > *Big problems, eg., can we provide guidance on how science budget should be allocated among various disciplines, or NSF CS budget among different areas? Given a subset of researchers, say we can estimate their impact on the society when funded. Given this oracle, can we allocate funds to people to maximize social welfare? Can we model people switching teams in second round or open bid systems for reallocating funds? Q: Why doesn't NSF give $'s to 2 teams for the same project and get them to compete? For some recent work, see the work of Shay Kutten, Ron Lavi and Amitabh Trehan.*

  We believe that this simple setting can also model various other real life situations such as peer-to-peer systems, and can be viewed as a step in the direction of research into the process of people teaming up to construct distributed systems. Over our past period of research, we have delved into the nuances of this setting trying to come up with the best protocols that will minimize the Strong

Price of Anarchy (SPOA) i.e. give solutions that maximize social welfare. We have shown protocols where agreement between agents but also a process for appealing rejections improves price of anarchy, especially with constraints on collaboration between agents.

Many variations of the previous game are of interest: different appeal mechanisms, grant size as a function of certain game parameters, uneven sharing of the grant, and negotiations between agents for their share. In future, one could also study existing 'natural' games (i.e. not just mechanisms) and look at dynamic environments (where researchers join and leave the system). Since starting in my faculty position, I have engaged in starting an informal reading group where students and interested faculty discuss Algorithmic Game Theory with the specific aim of extending this line of research.

I have a wide range of research interests and a varied background which equips me well to explore them. I am a good collaborator and am lucky to have excellent collaborators and advisors. I conclude by highlighting some more of the open problems that are of interest:

(1) There are many distributed dynamic network models broadly classifiable as edge-dynamic [20, 7, 25] and node-dynamic(such as our self-healing model). Can we devise a general theory that addresses (and integrates) various dynamic models? Can we use previous work on centralized algorithms for dynamic networks?

(2) There are many open 'self-healing' questions: Can we extend our model and algorithms .e.g byzantine faults, multiple failures, higher churn, load-balanced, edge-weighted graphs? Can we heal non-topological invariants e.g. a computation (we can call this *functional self-healing*)? Can we use proof labeling schemes [19] as a tool for newer self-healing algorithms e.g. as methods to verify invariants, or detect if some action is required? Can we go beyond Self-healing to other self-* properties such as self-stabilization [5]? Can we derive theoretical relationships between various self-* properties?

(3) *Algorithmic Game Theory in Distributed Systems and vice versa:* Our 'EU Games' line of research began a question which can be roughly stated as follows: *'How do certain distributed system form by themselves in a distributed manner?'* - this question applies to all the large scale networks we have seen evolve in our modern age and, I believe, game theory can vastly contribute to understanding of such processes. At the same time, many questions involving computation and communication in game theoretic settings can benefit from distributed algorithms. In our line of research, we are trying to make such connections e.g. by seeing, how the equilibria will be impacted if communication among agents is limited. In a way, game theory and distributed algorithms look at different aspects of similar multi-agent scenarios and intuitively, there should be deep connections which can be explored.

REFERENCES

[1] Yehuda Afek, Baruch Awerbuch, Serge A. Plotkin, and Michael E. Saks. Local management of a global resource in a communication network. *J. ACM*, 43(1):1–19, 1996.

[2] Armando Castañeda, Danny Dolev, and Amitabh Trehan. Compact routing messages in self-healing trees. *CoRR*, abs/1508.04234, 2015. Accepted to ICDCN 2016.

[3] Gregory Chockler and Amitabh Trehan. Towards self-healing sdn. In *Distributed Software Defined Net- works (DSDN) workshop, Principles of Distributed Computing (PODC) 2014*, 2014.

[4] Atish Das Sarma, Ashwin Lall, Danupon Nanongkai, and Amitabh Trehan. Dense subgraphs on dynamic networks. In MarcosK. Aguilera, editor, *Distributed Computing*, volume 7611 of *Lecture Notes in Computer Science*, pages 151–165. Springer Berlin Heidelberg, 2012.

[5] Edsger W. Dijkstra. Self-stabilizing systems in spite of distributed control. *Commun. ACM*, 17(11):643–644, November 1974.

[6] Shlomi Dolev and Nir Tzachar. Spanders: distributed spanning expanders. In *SAC*, pages 1309–1314, 2010.

[7] Michael Elkin. A near-optimal distributed fully dynamic algorithm for maintaining sparse spanners. In *Proceedings of the twenty-sixth annual ACM symposium on Principles of distributed computing*, PODC '07, pages 185–194, New York, NY, USA, 2007. ACM.

[8] Pierre Fraigniaud, Amos Korman, and David Peleg. Local distributed decision. In Rafail Ostrovsky, editor, *FOCS*, pages 708–717. IEEE, 2011.

[9] C. Gkantsidis, M. Mihail, and A. Saberi. Random walks in peer-to-peer networks: Algorithms and evaluation. *Performance Evaluation*, 63(3):241–263, 2006.

[10] Thomas P. Hayes, Jared Saia, and Amitabh Trehan. The forgiving graph: a distributed data structure for low stretch under adversarial attack. In *PODC '09: Proceedings of the 28th ACM symposium on Principles of distributed computing*, pages 121–130, New York, NY, USA, 2009. ACM.

[11] Thomas P. Hayes, Jared Saia, and Amitabh Trehan. The forgiving graph: a distributed data structure for low stretch under adversarial attack. *Distributed Computing*, pages 1–18, 2012.

[12] Tom Hayes, Navin Rustagi, Jared Saia, and Amitabh Trehan. The forgiving tree: a self-healing distributed data structure. In *PODC '08: Proceedings of the twenty-seventh ACM symposium on Principles of distributed computing*, pages 203–212, New York, NY, USA, 2008. ACM.

[13] Valerie King, Steven Lonargan, Jared Saia, and Amitabh Trehan. Load balanced scalable byzantine agreement through quorum building, with full information. In *ICDCN'11*, pages 203–214, 2011.

[14] Valerie King and Jared Saia. From almost everywhere to everywhere: Byzantine agreement with $\tilde{o}(n^{3/2})$ bits. In *DISC*, pages 464–478, 2009.

[15] Valerie King and Jared Saia. Breaking the $O(n^2)$ bit barrier: Scalable byzantine agreement with an adaptive adversary. In *PODC*, 2010.

[16] Valerie King and Jared Saia. Breaking the $o(n^2)$ bit barrier: Scalable byzantine agreement with an adaptive adversary, 2010. http://arxiv.org/abs/1002.4561.

[17] Sebastian Kniesburges, Andreas Koutsopoulos, and Christian Scheideler. Re-chord: a self-stabilizing chord overlay network. In *Proceedings of the 23rd ACM symposium on Parallelism in algorithms and architectures*, SPAA '11, pages 235–244, New York, NY, USA, 2011. ACM.

[18] Amos Korman and Shay Kutten. Controller and estimator for dynamic networks. In Indranil Gupta and Roger Wattenhofer, editors, *PODC*, pages 175–184. ACM, 2007.

[19] Amos Korman, Shay Kutten, and David Peleg. Proof labeling schemes. *Distributed Computing*, 22(4):215–233, 2010.

[20] Fabian Kuhn, Nancy Lynch, and Rotem Oshman. Distributed computation in dynamic networks. In *Proceedings of the 42nd ACM symposium on Theory of computing*, STOC '10, pages 513–522, New York, NY, USA, 2010. ACM.

[21] Shay Kutten, Ron Lavi, and Amitabh Trehan. Composition games for distributed systems: The eu grant games, 2013.

[22] Shay Kutten, Gopal Pandurangan, David Peleg, Peter Robinson, and Amitabh Trehan. On the complexity of universal leader election. In *Proceedings of the 2013 ACM Symposium on Principles of Distributed Computing*, PODC '13, pages 100–109, New York, NY, USA, 2013. ACM.

[23] Shay Kutten, Gopal Pandurangan, David Peleg, Peter Robinson, and Amitabh Trehan. Sublinear bounds for randomized leader election. In *ICDCN'13*, pages 348–362, 2013.

[24] Shay Kutten, Gopal Pandurangan, David Peleg, Peter Robinson, and Amitabh Trehan. Sublinear bounds for randomized leader election. *Theoretical Computer Science*, (0):–, 2014.

[25] Shay Kutten and Avner Porat. Maintenance of a spanning tree in dynamic networks. In Prasad Jayanti, editor, *Distributed Computing*, volume 1693 of *Lecture Notes in Computer Science*, pages 846–846. Springer Berlin / Heidelberg, 1999.

[26] C. Law and K. Y. Siu. Distributed construction of random expander networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, volume 3, pages 2133–2143 vol.3, 2003.

[27] Rebecca Linke. Spotlight on software-defined networks. http://www.itworld.com/print/323240.

[28] S. Muthukrishnan. http://www.mysliceofpizza.blogspot.com.

[29] Gopal Pandurangan, Peter Robinson, and Amitabh Trehan. Distributed construction and maintenance of deterministic expander networks(working title), 2012. In preparation.

[30] Gopal Pandurangan, Peter Robinson, and Amitabh Trehan. Dex: Self-healing expanders. In *Proceedings of the 2014 IEEE 28th International Parallel and Distributed Processing Symposium*, IPDPS '14, pages 702–711, Washington, DC, USA, 2014. IEEE Computer Society.

[31] Gopal Pandurangan and Amitabh Trehan. Xheal: localized self-healing using expanders. In *Proceedings of the 30th annual ACM SIGACT-SIGOPS symposium on Principles of distributed computing*, PODC '11, pages 301–310, New York, NY, USA, 2011. ACM.

[32] Peter Robinson Amitabh Trehan Seth Gilbert, Gopal Pandurangan. Dconstructor: Network construction with polylogarithmic overhead. Under Review.

[33] Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek, and Hari Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Trans. Netw.*, 11(1):17–32, 2003.

[34] Mikkel Thorup and Uri Zwick. Compact routing schemes. In *SPAA*, pages 1–10, 2001.

[35] Amitabh Trehan. *Algorithms for self-healing networks*. Dissertation, University of New Mexico, 2010.

[36] Amitabh Trehan. Self-healing using virtual structures. *CoRR*, abs/1202.2466, 2012.

[37] UIBK, FAU, QUB, KTH, NUMECA, and IBM. Allscale: An exascale program- ming, multi-objective optimisation and resilience management environment based on nested recursive parallelism, 2014. €4.3 Mi (QUB €450K).

[38] Wikipedia. Software-defined networking. https://en.wikipedia.org/wiki/Software-defined_networking.