RESEARCH STATEMENT
TOWARDS ROBUST DISTRIBUTED DYNAMIC ALGORITHMS

AMITABH TREHAN
LECTURER, COMPUTER SCIENCE
LOUGHBOROUGH UNIVERSITY, UK

INTRODUCTION

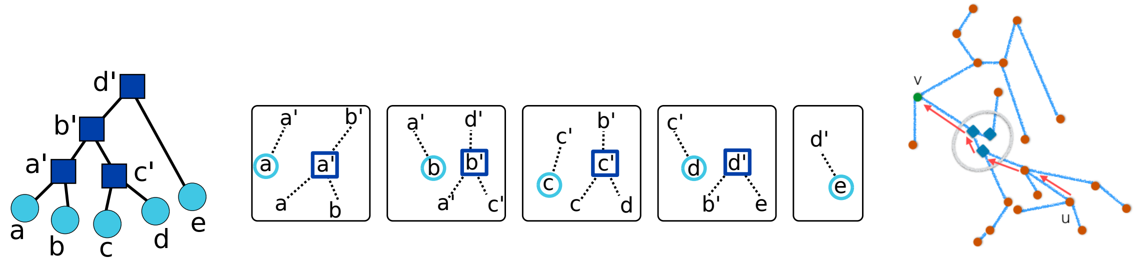
The discipline of distributed computing and algorithms is concerned with what can be achieved by interacting agents with limited local knowledge. This is a vital question in the upcoming era of ubiquitous networks, robotics and algorithms permeating every aspect of our life. The complexity of this field and the parameters involved (communication, memory, dynamicity, reliability and security) lend themselves to an almost unlimited number of mathematical models and interesting, important questions. My work seeks to address some of these questions.

For a large network of low-memory devices (such as the *Internet of Things* [1]), what is a good model of computation and how can we develop efficient, robust and secure algorithms in this model? Section 1 discusses our proposed program for addressing this question introducing the notions of compact memory and local streaming. Can we devise better algorithms and prove fundamental lower bounds for classic problems such as *Leader Election* and *Byzantine Agreement*. Can we apply the established theory of *fixed parameter tractability* to develop parametrised distributed algorithms? Section 3 addresses such questions and related research plans. Can we build autonomous networks that recover by themselves (self-heal/self-stabilise) in the face of adversarial attacks (fail-stop/hard, transient/soft, byzantine)? We address this in Section 2 by proposing the self-healing model [37] and highlighting the long line of research we have done in developing self-healing algorithms for overlay networks topology maintenance and critical processes such as network routing. Self-healing primarily addresses resource availability which is an essential aspect of security and an essential component for an autonomic system [15]. Are there distributed algorithms to design distributed systems and does game theory provide insight into such processes? Our work on algorithmic game theory and future directions are discussed in Section 4. My work explores a wide range of systems and scenarios including overlay and P2P networks (Section 3), Blockchains (Section 3), Exascale computers (vis the EU H2020 grant AllScale [43]), the *Internet of Things* (Section 1) and *Software Defined Networking* [6] (vis the *Newton International Fellowship* [36]), Fog and Edge networks [2], and Databases [32].

My main methods of research have involved **mathematical modelling** of the problems involved and bringing the various beautiful tools of combinatorial (and other) mathematics to design and prove the **efficiency of algorithms** and/or to elicit **fundamental understanding** of the problem at hand in terms of lower bounds. However, I am happy to explore algorithmic problems across a broad spectrum of Computer Science.

1. COMPACT MEMORY, ROUTING AND LOCAL STREAMING

A traditional assumption in message passing distributed computing is to assume unlimited computation at individual nodes - this is to concentrate on communication aspects. However, in reality, computation is limited, and in particular, constrained by resources such as memory. This is even more relevant now with the advent of small devices expected to connect in ad-hoc and unexpected ways into large networks (such as IOT and social networks) - this has presented unprcedented challenges vis technology, availability, usability and security. In [5], we present the *Compact Message Passing / Distributed Streaming (CMP/DS)* model which looks at designing distributed algorithms when the local memory is much smaller ($O(polylog\ n)$) than even



(a) Compact function f to query labeled BST trees/half-full trees: On the left is such a tree with 5 leaves. $f(5, b)$ will return the second box (having the $O(\log n)$ sized subtrees of b and b')

(b) CompactFTZ routes messages between low memory nodes (u and v) despite node failures

FIGURE 1. Algorithms in CMP/DS: Compact functions and Self-Healing Routing

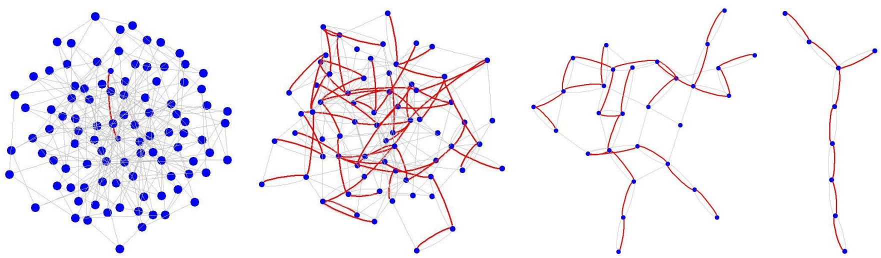
the number of neighbours a node may have ($O(n)$) - a consequence of this is that our solutions rely on a streaming style of computation, hence, this model combines both the streaming and distributed paradigms. The limited local memory also limits the network bandwidth, thus our model is stronger than the $CONGEST$ model [31]. We show some problems are memory agnostic whereas others are memory sensitive requiring no overhead or necessary overhead respectively, as compared to standard CONGEST solutions. For these solutions, we also introduce a solution concept around *local compact functions* and *compact protocols* (refer Figure 1).

One active line of research which has always recognised the memory issue (both from the algorithmic theory and networks in practice perspective) is *Compact Routing* which seeks to keep sublinear (in network size n) routing tables on nodes. This has spawned outstanding and award-winning research. Our research addresses two important shortcomings: compact routing is very sensitive to topological changes and the initial setup is usually centralised (decentralising this has been the topic of much recent research e.g. [10]). In [3, 4] (**Best paper runner up**, ICDCN 2016), we developed *CompactFTZ*, one of the few 'dynamic' compact routing schemes (in fact, a self-healing (discussed in Section 2) scheme) and in [5] we made the scheme fully decentralised in the CMP/DS model. Our solutions were based on the well-known routing of Thorup-Zwick [35] and a compact version of our self-healing algorithm ForgivingTree [14].

This work has been funded by my *EPSRC First Grant* COSHER [41], a Newton Fund AMC mobility grant to visit UNAM, Mexico [36] and an I-CORE fellowship [39] - we have also developed extensions such as fully dynamic (handling both deletions and insertions) self-healing schemes. Building on these results further, I plan a research program to develop comprehensive solutions and theoretical foundations for low-memory distributed algorithms.

2. SELF-HEALING AND DYNAMIC NETWORKS

In my Ph.D. dissertation [37], 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 4 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.

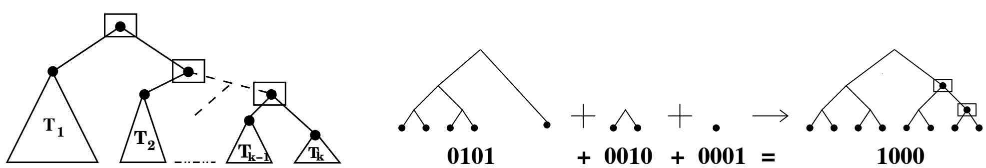


(a) single deletion (b) 30 deletions (c) 70 deletions (d) 90 deletions

FIGURE 2. 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.

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 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 [38]. Outlines of some of our algorithms follows:

- FORGIVINGGRAPH [12, 13]: 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 3(a) and 3(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 3. Half-full trees (hafts)

- *DEX and DConstructor: Resilient Distributed Expander Constructions* [29, 33]*:* 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 [28]. Having such an algorithm not only has immediate impact on many algorithms that use expander construction as a building block such as Xheal [30], but also extends additional functionality to existing popular overlay networks such as Chord [34] and Re-Chord [17]. Only a few, and that too, randomized distributed construction of expander graphs are known [11, 8, 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.

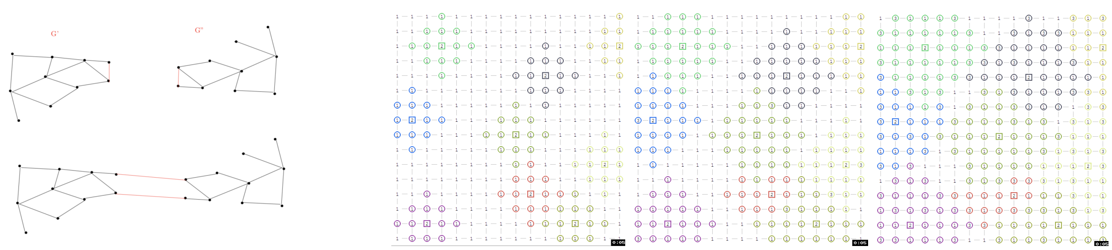
[1]here, by deterministic expanders, we imply that the expansion properties hold deterministically, not just *with high probability*

We believe our models and work so far has a lot to contribute in the near future: i) *A unified understanding of dynamicity:* 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. There are only a few models addressing dynamicity such as Kuhn, Lynch and Oshman's *Dynamic Graph* model [19] and our self-healing model but it will be good to get a unified model or deeper understanding of dynamism. ii) *Reliable high performance computing:* The immense scale of high performance systems such as exascale computing systems reveals sporadic faults; in [43], we investigated applying the concept of self-healing towards exascale systems. iii) *Self-healing Software Defined Networks and Streaming Data Analytics:* SDN is a new and commercially important paradigm that provides an important use case for self-healing for which we developed initial ideas [6] and won the prestigious **Newton Incoming Fellowship** of the Royal Society for a proposal based on developing self-healing SDN.

3. FUNDAMENTAL DISTRIBUTED ALGORITHMS

Here, we discuss our research work and plans involving some fundamental distributed computing problems.

- *Leader election algorithms - Efficiency and Complexity:* In our *J. ACM* [24] paper (invited as one of the two best papers from PODC 2013 [21]), we proved the folklore lower bounds of $O(D)$ time and $O(m)$ messages for even high probability randomised algorithms and gave efficient algorithms that try to match both bounds simultaneously. This includes the most efficient simultaneously time-messages deterministic LE algorithm ($O(D \log n)$ time and $O(m \log n)$ messages) - we (with my current Ph.D. student) am trying to bridge this gap. In earlier work, for *implicit Leader Election* (all nodes need not know leader's identity), we developed the first sublinear (in messages) LE for graphs with good expansion [22] (Journal version: [23]) (this work won the **best paper award** at *ICDCN 2013*).



(a) The *Bridge Crossing* gadgets used to prove $\Omega(|\#Edges|)$ messages lower bound

(b) Stages of deterministic Growing Kingdom Leader Election. Various candidates (different colours) grow their kingdoms clashing at boundaries.

FIGURE 4. Lower and Upper Bounds for Leader Elections from [24]

- *Scalable Byzantine Agreement, Blockchain and Security:* In [16], we addressed the problem of designing distributed algorithms for large scale networks under byzantine attacks. Byzantine Agreement is a building block for such algorithms and for modern technologies such as *blockchains* and *cryptocurrency*. This built on earlier groundbreaking work of King and Saia while introducing an important technique of developing a *good quorum* (a set of $O(\log n)$ processors that contains a representative fraction of good processors). This has influenced much further work. I am also investigating using blockchain technology e.g. in securing edge networks (ref. EPSRC grant application PREEMPT [42].
- *Distributed Complexity and Parametrized Distributed Algorithms:* Unlike centralized algorithms, complexity theory is rather poorly established for distributed algorithms (partly due to the variety of models). There is research e.g. with proof labeling schemes [18] - I would like to explore these directions. *Fixed Parameter Tractability (FPT)* is an established paradigm to develop practical solutions for NP-hard problems; we are applying techniques and adapting parametrized algorithms towards distributed computing. and already have results e.g. for *K-feedback vertex set*.

4. 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 [20], 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. We paraphrase the following comment on our work (by Prof. S. Muthukrishnan, of Rutgers, in his blog [27]:

  > *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 and we are actively exploring this further (e.g. with a LMS CS Scheme 7 grant [40]).

I have a wide range of research interests and a varied background which equips me well to explore them. I believe I am a good collaborator and am lucky to have excellent collaborators and advisors. I have highlighted a number of research areas and problems in this statement and am always keen to explore algorithmic questions in scenarios of interest.

REFERENCES

[1] What is the internet of things? wired explains. `https://www.wired.co.uk/article/internet-of-things-what-is-explained-iot`. Accessed: 2018-12-03.

[2] Iain Phillips Lin Cui Amitabh Trehan, Posco Tso. Workshop co-chairs, 2018. `https://synnetsys.github.io/topic2018/`.

[3] Armando Castañeda, Danny Dolev, and Amitabh Trehan. Compact routing messages in self-healing trees. In *Proceedings of the 17th International Conference on Distributed Computing and Networking, Singapore, January 4-7, 2016*, pages 23:1–23:10. ACM, 2016.

[4] Armando Castañeda, Danny Dolev, and Amitabh Trehan. Compact routing messages in self-healing trees. *Theor. Comput. Sci.*, 709:2–19, 2018.

[5] Armando Castañeda, Jonas Lefèvre, and Amitabh Trehan. Self-healing routing and other problems in compact memory. *CoRR*, abs/1803.03042, 2018.

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

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

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

[9] 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.

[10] Michael Elkin and Ofer Neiman. On efficient distributed construction of near optimal routing schemes. *Distributed Computing*, 31(2):119–137, 2018.

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

[12] 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.

[13] 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.

[14] 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.

[15] IBM. http://www.research.ibm.com/autonomic/manifesto/autonomic computing.pdf.

[16] 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.

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

[19] 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.

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

[21] 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.

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

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

[24] Shay Kutten, Gopal Pandurangan, David Peleg, Peter Robinson, and Amitabh Trehan. On the complexity of universal leader election. *J. ACM*, 62(1):7:1–7:27, 2015.

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

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

[29] 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.

[30] 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.

[31] David Peleg. *Distributed Computing: A Locality Sensitive Approach*. SIAM, 2000.

[32] Daniel Playfair, Amitabh Trehan, Barry McLarnon, and Dimitrios S. Nikolopoulos. Big data availability: Selective partial checkpointing for in-memory database queries. In James Joshi, George Karypis, Ling Liu, Xiaohua Hu, Ronay Ak, Yinglong Xia, Weijia Xu, Aki-Hiro Sato, Sudarsan Rachuri, Lyle H. Ungar, Philip S. Yu, Rama Govindaraju, and Toyotaro Suzumura, editors, *2016 IEEE International Conference on Big Data, BigData 2016, Washington DC, USA, December 5-8, 2016*, pages 2785–2794. IEEE, 2016.

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

[34] 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.

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

[36] Amitabh Trehan. Newton fund international collaboration programme amc mobility grant, 08/2016 - 09/2016. £2,000 plus travel.

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

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

[39] Amitabh Trehan. Postdoctoral fellow (hosted by prof. danny dolev huji), 2013. NIS 120,000 p.a.

[40] Amitabh Trehan. Advanced grant games and their analysis, 2017. £500.

[41] Amitabh Trehan. Ep/p021247/1: Compact self-healing and routing over low memory nodes (cosher), 2017-19. £125,000.

[42] Amitabh Trehan. Preempt: Cross-domain collaboration network for early edge-based cyber threat detection and defence, 2018. £420,000.

[43] 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).