Internship LIP6 SU UPMC

On Byzantine fault tolerance in dynamic networks

Contacts: Binh-Minh Bui-Xuan <buixuan@lip6.fr>
Sébastien Tixeuil <sebastien.tixeuil@lip6.fr>

Venue: NPA (Networks and Performance Analysis), LIP6 SU UPMC, Paris, France.

Keywords: algorithm, graph mining, temporal data.

Maintaining connectivity is fundamental for network studies [15], especially in dynamic settings [10, 14]. Pieces of information can be exchanged in interconnected networks that may evolve over time. In this context, applications using time series databases (TSDB) and graph oriented databases (such as Neo4j) naturally capture the modeling of such dynamic exchanges of information, see e.g. [4] for an extensive overview. This kind of data arises from as many topics as IP routing [2, 22], public transportation [7, 11], road management [16, 21], e-marketing [3], social networks [17], and so on.

Furthermore, redundant connectivity is instrumental for designing communication primitives that can withstand Byzantine attackers [8, 9] (that is, attackers with unbounded computational power whose behavior may be arbitrary, and in particular malicious). For the reliable communication problem in dynamic networks in the presence of Byzantine entities, it is known [18] that a dynamic minimal cut of $2k + 1$ is necessary and sufficient to tolerate k Byzantine participants.

The current internship proposal addresses such input. We aim at further results optimizing solutions for $\textsc{Journey}$ and k-$\textsc{Journeys}$, two optimisation problems defined below. The former problem helps modeling connectivity in dynamic networks, while the latter problem addresses Byzantine fault tolerant routing through them.

In classical graph theory, $\textsc{ShortestPath}$ asks for the computation of a path joining two given vertices with the least number of edges in a (static) graph. It can be solved by popular polynomial time algorithms such as Dijkstra, Bellman-Ford, Floyd-Warshall, and so on, see *e.g.* [5].

Given a temporal graph, two timestamped source-target vertices "s at moment μ_s" and "t at moment μ_t", and a cost function $c : \mathbb{N} \times \binom{V}{2} \to \mathbb{R}$, we define $\textsc{Journey}$ as the problem of computing a list

$$J = [(\mu_1, v_1v_2), (\mu_2, v_2v_3), \ldots, (\mu_{p-1}, v_{p-1}v_p)] \text{ with } v_1 = s, \mu_1 = \mu_s, v_p = t, \text{ and } \mu_{p-1} = \mu_t,$$

such that $c(J) = c(\mu_1, v_1v_2) + c(\mu_2, v_2v_3) + \ldots + c(\mu_{p-1}, v_{p-1}v_p)$ is not only minimized, but also what is called *realizable*, in the following sense.

The most common realizability criteria is *timely monotone*: $\mu_1 < \mu_2 < \ldots < \mu_{p-1}$. It is proved that topological sorting helps in extending popular $\textsc{ShortestPath}$ to solve $\textsc{Journey}$ when under *timely monotone* constraint in polynomial time [2]. A stronger constraint is called *timed transit* w.r.t. a given transit function $\delta : \binom{V}{2} \to \mathbb{R}$ denoting the time it takes for sailing from one vertex to another vertex. Here, we must have $\mu_{i+1} = \mu_i + \delta(v_i, v_{i+1})$, for every $1 \leq i < p$.

$\textsc{Journey}$ under *timed transit* has many applications in industries. For instance, on planning taxiway usage in an airport, a least cost plan is a set of k temporal journeys for moving aircraft. Such a journey answers to a pilot request of moving from one point to another at a given time window. In this sense, each journey can be viewed as a list of timetable checkpoints with expected passing time instants for each aircraft. A principal constraint for the k journeys is to be vertex-disjoint at all time, in order to prevent collision between aircraft. The overall goal is to minimize the expected global fuel consumption for the aircraft to realize the k journeys.

Putting aircraft and taxiways aside, vertex-disjoint journeys can benefit to Byzantine tolerance too. Here, the notion of vertex-disjoint must be extended to be independent of times. A formal definition of these two kind of vertex-disjoint journeys is given in the sequel, under the name of k-$\textsc{Journeys}1$-2. Informally, k-$\textsc{Journeys}1$-2 can be seen as the temporal extension of k-$\textsc{Paths}$, where, given a (static) graph G and k pairs $(s_i, t_i) \in V(G)^2$ of source-target vertices, for $1 \le i \le k$, the goal is to compute k vertex disjoint paths joining each source s_i to its corresponding target t_i while minimizing the total distance of the k paths. When $k = 1$, the problem is reduced downto $\textsc{ShortestPath}$ and $\textsc{Journey}$ as defined previously.

Formally, we define k-$\textsc{Journeys}1$ to be the problem of joining k pairs of timestamped source-target vertices "s_i at moment μ_{s_i}" and "t_i at moment μ_{t_i}", each by a journey J_i solving $\textsc{Journey}$ on $s_i, \mu_{s_i}, t_i, \mu_{t_i}$, such that for every (v, μ), there are no J_i and $J_{i'}$ where $(v, \mu) \in J_i$ and $(v, \mu) \in J_{i'}$, and such that the total sum $c(J_1) + c(J_2) + \ldots + c(J_k)$ of cost function over the k journeys is minimized.

We also define k-$\textsc{Journeys}2$ to be the problem of joining k pairs of timestamped source-target vertices "s_i at moment μ_{s_i}" and "t_i at moment μ_{t_i}", each by a journey J_i solving $\textsc{Journey}$ on $s_i, \mu_{s_i}, t_i, \mu_{t_i}$, such that for every v, there are no μ, μ', J_i and $J_{i'}$ where $(v, \mu) \in J_i$ and $(v, \mu') \in J_{i'}$, and such that the total sum $c(J_1) + c(J_2) + \ldots + c(J_k)$ of cost function over the k journeys is minimized.

Related works: For the general case when $k > 1$, $k-\textsc{Paths}$ is an NP-hard optimization problem [20] with interesting solutions including both matrix approach [6] and dynamic programming (DP) [13]. Matrix and DP approaches are important for practical applications in the sense that implementations usually require only a few for-loops filling some array, hence, less likely to contain bugs. Both matrix and DP can be used to provide implementations of sublinear optimization and polynomial time approximation scheme (PTAS) as what has been done in other graph mining problems, *e.g.* with the use of matrix splitters in [3] and geometric graph DP in [19], respectively. For k-$\textsc{Paths}$, a standard heuristics is to precompute shortest path and distance matrices, *e.g.* with Floyd-Warshall algorithm, then dynamically attribute paths joining s_i to t_i in a greedy manner, recomputing reroute options when a vertex is already in use by an earlier path. The more involved DP given in above cited Ref. [13] use vertex-separator properties of tree-width decomposition in order to provide fixed parameter tractable (FPT) algorithm solving optimally k-$\textsc{Paths}$ with a small polynomial for the term that is independent from parameter k. Such a technique is particularly interesting for a practical application to k-$\textsc{Journeys}1$-2.

Internship proposal: We propose to investigate either the case of k-$\textsc{Journeys}1$ or k-$\textsc{Journeys}2$. Moved by the example of aircraft, a solution to this kind of problems must be selected from a double combinatorial parameter. Firstly, as with cars, fuel consumption here is subject to the distance between the checkpoints. Moreover, aircraft consumption highly depends on its speed, where stopping and restarting the jet engines is critical. Ideally, solutions to our problems should also consider such practical matters. Besides, we note that equivalent terminologies of k-$\textsc{Journeys}1$-2 exist in the literature, such as Byzantine fault tolerance routes or Menger's theorem for temporal graphs [18, 1]. We stress that even when $k = 2$, several definitions of the problem exist, leading to both polynomial and NP-complete cases [12].

Main question: *Under timed transit, how efficient Byzantine fault tolerant journeys can be found in dynamic networks? How about the same question without timed transit?*

References

[1] Silvia Bonomi, Giovanni Farina, and Sébastien Tixeuil. Reliable broadcast in dynamic networks with locally bounded byzantine failures. In Taisuke Izumi and Petr Kuznetsov, editors, *Stabilization, Safety, and Security of Distributed Systems - 20th International Symposium, SSS 2018, Tokyo, Japan, November 4-7, 2018, Proceedings*, volume 11201 of *Lecture Notes in Computer Science*, pages 170–185. Springer, 2018.

[2] B.-M. Bui-Xuan, A. Ferreira, and A. Jarry. Computing shortest, fastest, and foremost journeys in dynamic networks. *International Journal of Foundations of Computer Science*, 14(2):267–285, 2003.

[3] B.M. Bui-Xuan, H. Hourcade, and C. Miachon. Computing small temporal modules in time logarithmic in history length. *Social Network Analysis and Mining*, 12(1):19, 2022.

[4] A. Casteigts, P. Flocchini, E. Godard, N. Santoro, and M. Yamashita. Expressivity of time-varying graphs. In *19th International Symposium on Fundamentals of Computation Theory*, pages 95–106, 2013.

[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 1989.

[6] S. Datta, S. Iyer, R. Kulkarni, and A. Mukherjee. Shortest k-disjoint paths via determinants. In *38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 122 of *LIPIcs*, pages 19:1–19:21, 2018.

[7] J. Dibbelt, T. Pajor, B. Strasser, and D. Wagner. Connection scan algorithm. *ACM Journal of Experimental Algorithmics*, 23, 2018.

[8] Danny Dolev. Unanimity in an unknown and unreliable environment. In *22nd Annual Symposium on Foundations of Computer Science, Nashville, Tennessee, USA, 28-30 October 1981*, pages 159–168. IEEE Computer Society, 1981.

[9] Danny Dolev. The byzantine generals strike again. *J. Algorithms*, 3(1):14–30, 1982.

[10] L. Foschini, J. Hershberger, and S. Suri. On the complexity of time-dependent shortest paths. *Algorithmica*, 68(4):1075–1097, 2014.

[11] R. Geisberger, P. Sanders, D. Schultes, and C. Vetter. Exact routing in large road networks using contraction hierarchies. *Transportation Science*, 46(3):388–404, 2012.

[12] A. Ibiapina and A. Silva. Snapshot disjointness in temporal graphs. In *2nd Symposium on Algorithmic Foundations of Dynamic Networks*, volume 257 of *LIPIcs*, pages 1:1–1:20, 2023.

[13] K. Kawarabayashi, Y. Kobayashi, and B. Reed. The disjoint paths problem in quadratic time. *Journal of Combinatorial Theory, Series B*, 102(2):424–435, 2012.

[14] D. Kempe, J. Kleinberg, and A. Kumar. Connectivity and inference problems for temporal networks. *Journal of Computer and System Sciences*, 64(4):820–842, 2002.

[15] J. Kleinberg and E. Tardos. *Algorithm Design*. Pearson/Addison-Wesley, 2005.

[16] A. Kosowski and L. Viennot. Beyond highway dimension: Small distance labels using tree skeletons. In *28th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1462–1478. SIAM, 2017.

[17] M. Latapy, T. Viard, and C. Magnien. Stream graphs and link streams for the modeling of interactions over time. *Social Network Analysis and Mining*, 8(61), 2018.

[18] Alexandre Maurer, Sébastien Tixeuil, and Xavier Défago. Communicating reliably in multihop dynamic networks despite byzantine failures. In *34th IEEE Symposium on Reliable Distributed Systems, SRDS 2015, Montreal, QC, Canada, September 28 - October 1, 2015*, pages 238–245. IEEE Computer Society, 2015.

[19] T. Picavet, N.T. Nguyen, and B.M. Bui-Xuan. Temporal matching on geometric graph data. In *12th International Conference on Algorithms and Complexity*, volume 12701 of *LNCS and LNTCS*, pages 394–408, 2021.

[20] N. Robertson and P.D. Seymour. Graph minors. XIII. the disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63(1):65–110, 1995.

[21] F.J. Ros, P.M. Ruiz, and I. Stojmenovic. Acknowledgment-based broadcast protocol for reliable and efficient data dissemination in vehicular ad-hoc networks. *IEEE Transactions on Mobile Computing*, 11(1):33–46, 2012.

[22] J. Villacis-Llobet, B.M. Bui-Xuan, and M. Potop-Butucaru. Foremost non-stop journey arrival in linear time. In *29th International Colloquium on Structural Information and Communication Complexity*, volume 13298 of *LNCS*, pages 283–301, 2022.