iRoute: An IoT Fire Escape Application

Aditya, Ovia, Phani, Samuel

Indian Institute of Technology (Delhi and Madras)

25th June 2016

Overview

Problem Motivation

- Current fire escape solutions only indicate the presence or absence of fire. We intend to provide a framework which, in the event of a fire, helps the endangered by giving him the most appropriate escape path.
- We present this as an IoT framework which leverages only temperature sensors and the endangered person's relative location in the building (using perhaps a smartphone) in the case of a fire catastrophe.

Sensing and Alerting

- We place temperature sensors at critical locations on the pathways in a building, and send the temperature values to the cloud.
- On a potential fire outbreak, the endangered send's his location to the cloud.
- The cloud quickly responds with the safest escape path - a path with the most agreeable temperature values.

Path-Finding Algorithm: Description-1

- Let $G = (V, E)$ be the temperature point graph. For each $(u, v) \in E$ with vertex temperature labels t_u, t_v respectively, the edge weight is $w(u, v) = t_u - t_v$.
- Define path-index $\rho(p) = \sum_{(u,v)\in p} w(u, v)$.
- We have to find a simple (directed) path with maximum path index from the endangered to one of the fire exits.
- At the first step, we take a hop from the endangered to all his adjacent nodes.
- At any subsequent step, we take a hop from all terminal nodes in each sub-path to their adjacent nodes.

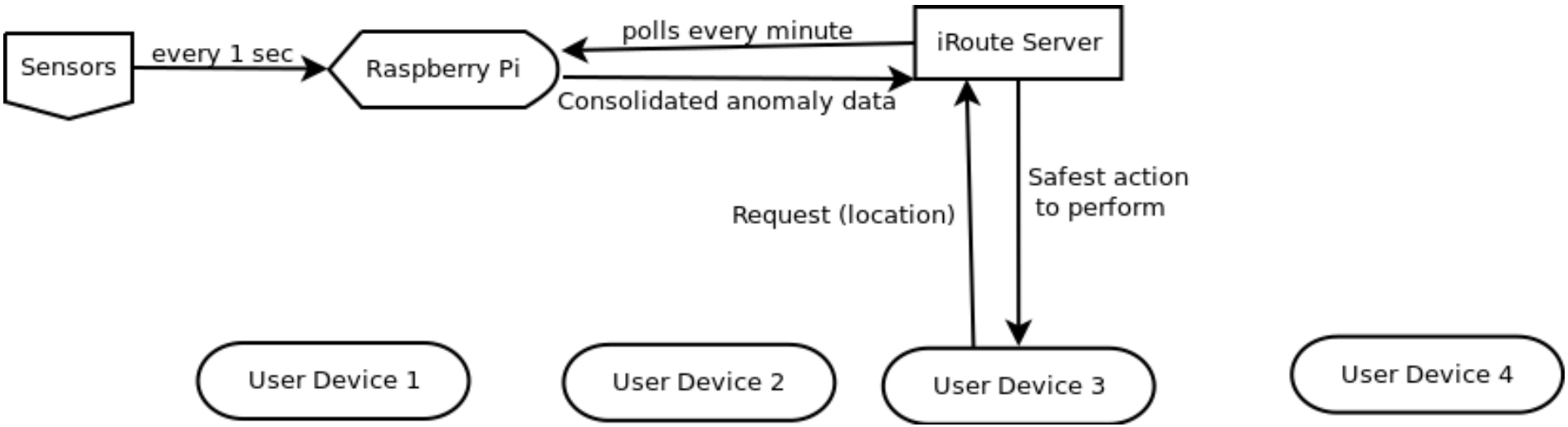
Path-Finding Algorithm: Description-2

- We maintain the following invariant: *after i iterations, we identify all i-length sub-paths from the endangered towards the exit*.
- If in any iteration, $\exists \geq 2$ sub-paths with the same terminal vertex, we only retain the sub-path with maximum path index (and drop the rest).
- On termination, we return the fire-exit path with maximum path index.

Algorithm Correctness and Time Complexity

- A simple proof by contradiction would show, that for the max path-index path, every sub-path also has max path-index.
- As the max length of any path is $|E|$, due to our invariant, our algorithm runs in linear-time (assuming the bookkeeping takes constant time per iteration).

Implementation Architecture



Future Extensions

- We can augment our solution by using Indoor Localization [1, 2].
- We can use temporal locality of temperature.
- We can use the same solution for cold regions (by negating the edge weights).

References

Martin, Eladio, Oriol Vinyals, Gerald Friedland, and Ruzena Bajcsy. "Precise indoor localization using smart phones." In Proceedings of the 18th ACM international conference on Multimedia, pp. 787-790. ACM, 2010.

Lim, Chin-Heng, Yahong Wan, Boon-Poh Ng, and Chong-Meng Samson See. "A real-time indoor WiFi localization system utilizing smart antennas." IEEE Transactions on Consumer Electronics 53, no. 2 (2007): 618-622.

Thanks. Questions?