

Minimum Makespan Workload Dissemination in DTNs: Making Full Utilization of Computational Surplus Around

Sheng Zhang, Nanjing University

Jie Wu, Temple University

Sanglu Lu, Nanjing University

Motivations

- Wireless devices around us: laptops, smartphones, ...
 - Intermittent and opportunistic connectivity
- Most of them are **idle** during most of the time.



Why not utilize these computational surplus around us?

Advantages of Utilizing Nearby Devices

- compared with mobile cloud computing

➤ Low cost for users

- Users do not need to pay any service fee to cloud providers for offloading computation to remote clouds.

➤ Ad-hoc

- The approach can be used in cases where there are no infrastructure-based services, e.g., in a disaster.

➤ Friendly to the Internet (wide-area network)

- The Internet is relieved, since, otherwise, uploading tasks and data to clouds may take up a large amount of bandwidth.

Problem Statement

➤ Task

- Output is small, thus, we focus on workload disseminating phase, and do not care about output gathering phase.
- Workload is fine-grained: the total workloads can be redistributed in any ratio between two nodes in a contact.

➤ Network

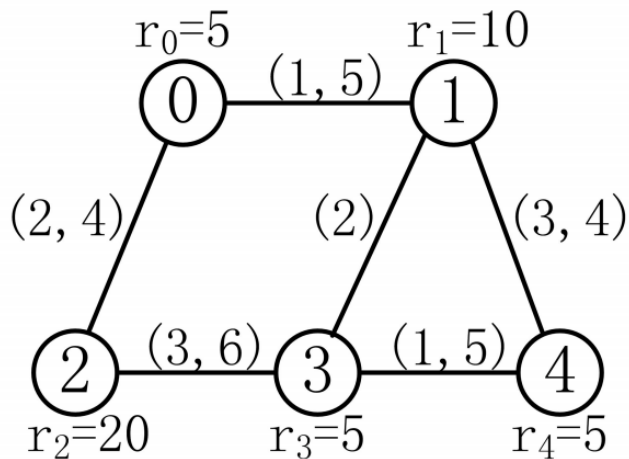
- Each node has a processing rate.
 - Inter-contact time is exponentially distributed.
 - We do not consider storage and bandwidth constraints.
- As a starting point, we only consider a single task in an “empty” DTN.



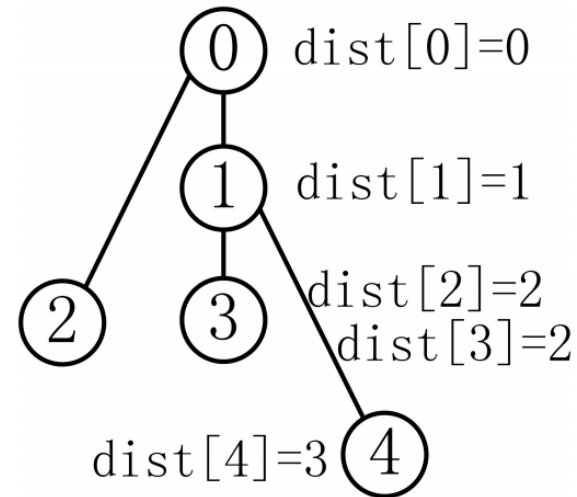
Given a task that originates at some node in a DTN, how are we to disseminate the workload during pairwise contacts, so as to minimize the makespan?

The Oracle Case with Global and Future Knowledge

- When we have global and future information, we can derive the shortest communication delays from the source to all of the other nodes.
- To minimize the makespan, we just have to utilize each node as soon as possible, and make sure that all participating nodes finish their respective workloads at the same time.



(a) Discrete contact graph



(b) Shortest delay tree

The Distributed Case

- Workload queue: manages operations on actual workload.
- The r-hop manager: maintains contact rate, opportunistic path, and workload of each node that is r-hop away.
- Finish time estimator: estimates the expected finish time, given a workload.
- Future workload estimator: estimates the expected future workload of itself

The total workloads are split between two nodes in a contact such that the expected finish time of their respective workloads are equal.

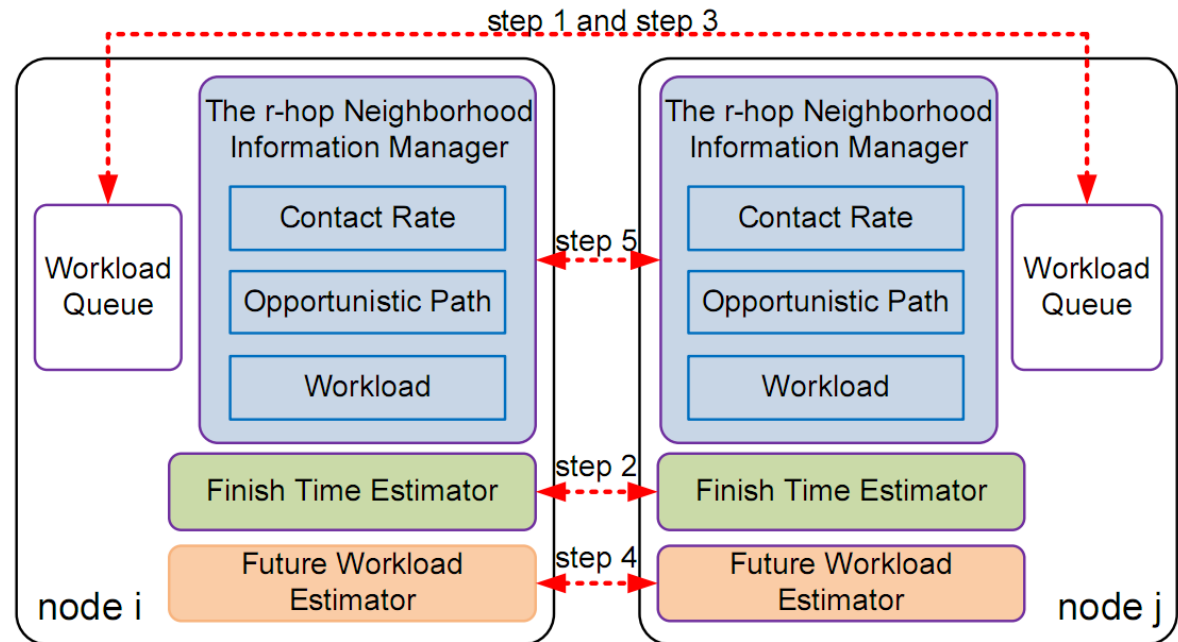
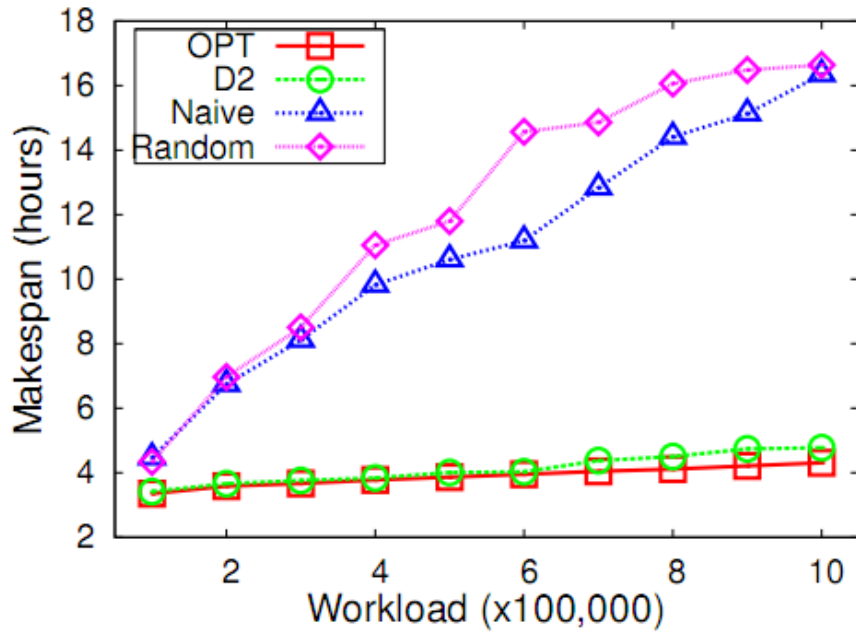
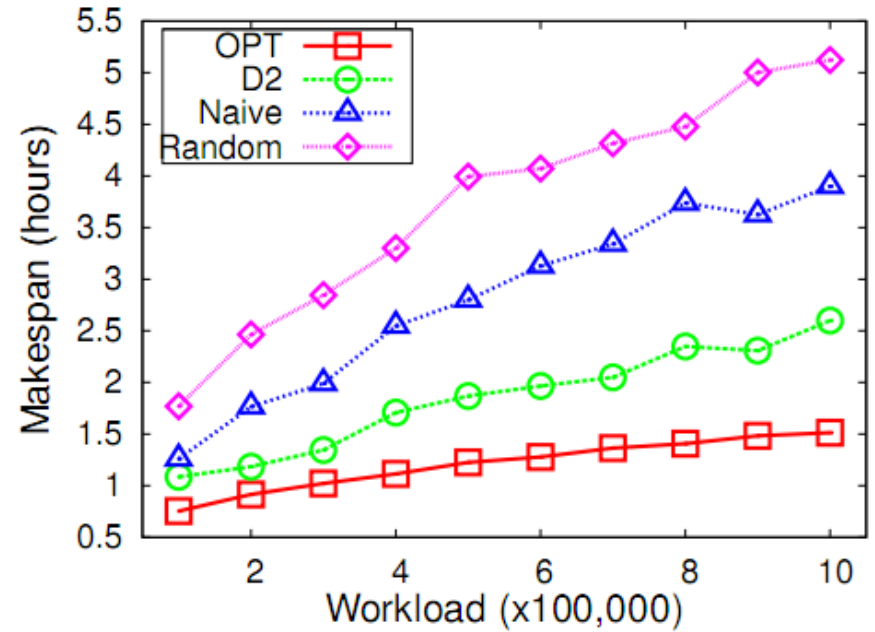


Figure 4: The architecture of D2.

Trace-driven Simulations



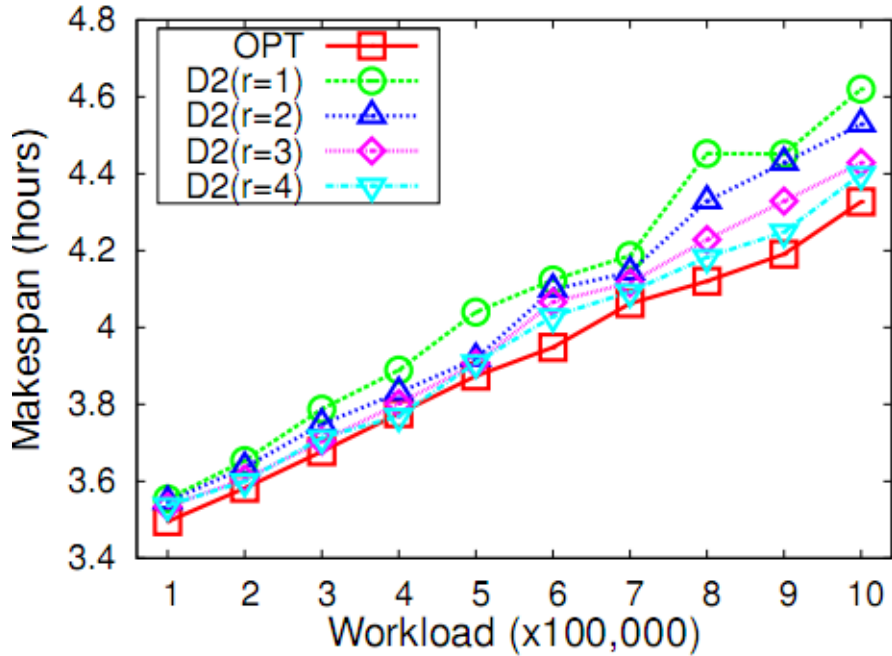
(a) Infocom06 trace



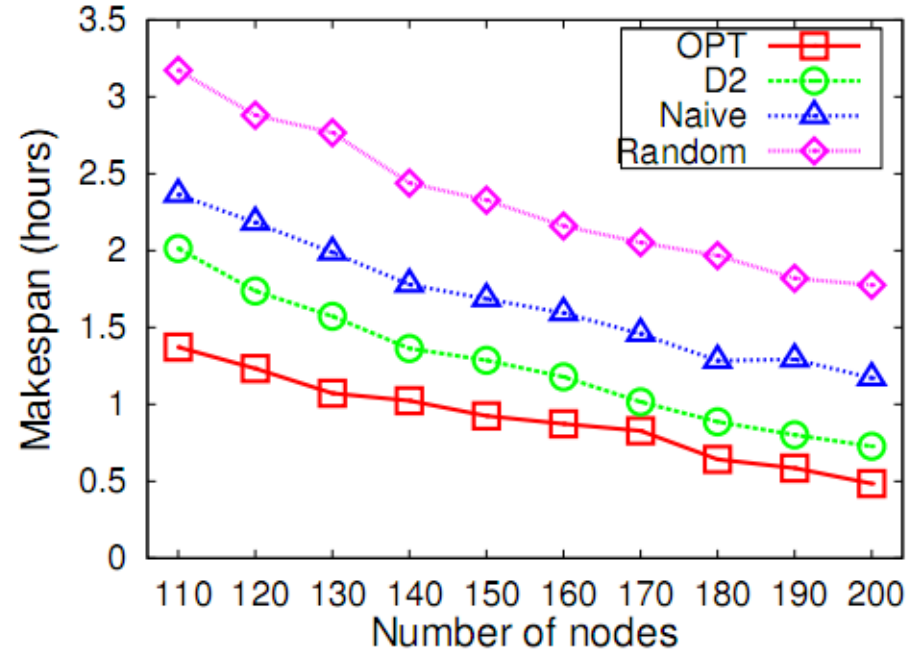
(b) Synthetic trace

Figure 5: Performance comparison under different workloads while keeping $MaxCapacity = 10$.

Trace-driven Simulations



(a) Impact of r (Infocom06)



(b) Impact of N (Synthetic)

Figure 6: Sensitivity results.

Conclusion

- In this paper, we propose making full utilization of computational surplus around us.
- We design a centralized optimal algorithm, OPT, and a distributed protocol, D2, for the minimum makespan workload disseminating problem.
- Future work
 - Multiple simultaneous tasks
 - Storage and bandwidth constraints

Thanks for your attention.