# Proactive Data Dissemination to Mission Sites

Fangfei Chen, Matthew P. Johnson, Amotz Bar-Noy, Iris Fermin and Thomas F. La Porta

We consider a specialized content distribution application. When a mission is created, for example when an alarm for a fire is reported, data is pushed to storage nodes at the mission site where it may be retrieved locally by responding personnel (e.g., police, firefighters, paramedics, government officials, and the media). It is important that when the information is requested or *pulled* by the personnel, it is available with low latency. We define an ILP to pick optimal storage nodes to minimize the latency-based cost and then extend the ILP to explicitly account for congestion. We show that the problem is NP-hard and develop distributed protocols that achieve close to the ILP bound of minimum cost.

## ILP formulation

$$\min \sum_{i,j} c_{i,j} x_{i,j}$$

All items placed

$$s.t. \sum_{j} x_{i,j} \geq 1 \quad , \quad \sum_{i} w_{i,j} x_{i,j} \leq b_j$$

Capacity of SN

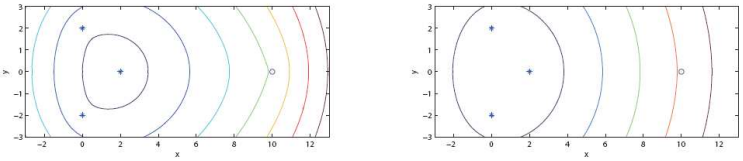
Size of item

$$x_{i,j} \in \{0,1\}$$

$$c_{i,j} = \alpha \cdot s_{i,j} + (1-\alpha) \cdot r_{i,j}$$

## Cost Contours

We approximate the number of hops by distance. The cost function becomes: $c_{ij} = \alpha \cdot d(D_i, S_j) w_i + (1-\alpha) \sum_{k \in R(D_i)} d(S_j, U_k) w_i$



(a) Weight 0.5 on pulling

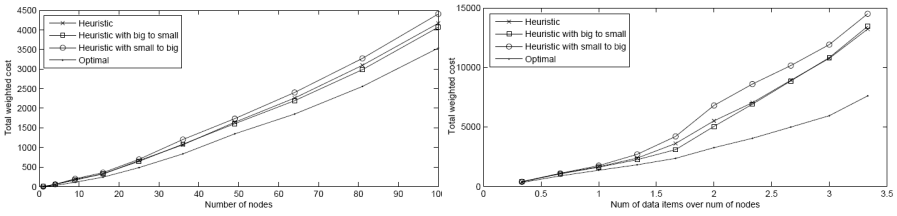
(b) Weight 0.9 on pulling

## Heuristics

**Algorithm 1** Greedy Algorithm

1: **for** each data item $D_i$ in arbitrary order **do**
2: **if** $D_i$ currently fits in at least one node having a smaller cost than that of $D_i$ staying at the provider **then**
3: place $D_i$ in a node $j$ minimizing $c_{ij}$
4: **else**
5: $D_i$ stays at its provider
6: **end if**
7: **end for**

## Performance



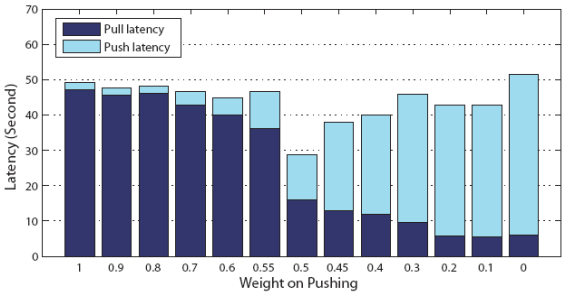

(a) Cost versus problem size (varying # data items and # nodes)

(b) Cost versus problem hardness (varying # data items only)

## Push Pull Trade-off

The latency $T_r$ in retrieving data as seen by a user may be expressed as:

$$T_r = \begin{cases} T_{ps} - T_u + T_{pl} & \text{if } T_{ps} > T_u \\ T_{pl} & \text{otherwise} \end{cases}$$

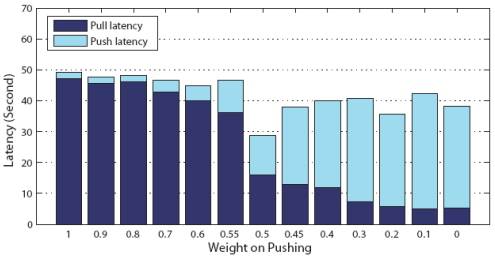


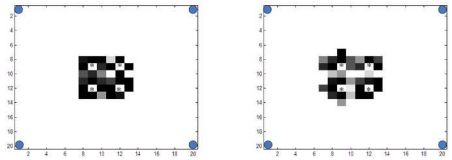

## Virtual Memory Occupancy

We add constraints (in ILP) to the one-hop neighborhood of a node:

$$\sum_{k} \sum_{i} w_i x_{ik} \leq e$$

And we introduce *virtual memory occupancy* for heuristics. When a storage node accepts data, it occupies some "virtual" memory in its neighboring nodes.



(a) Placing without VM

(b) Placing with VM

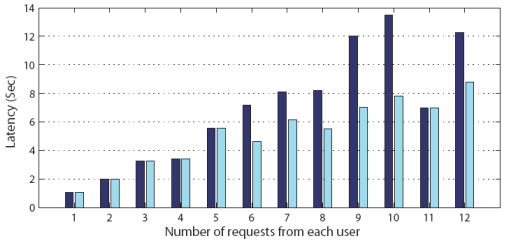



Fig. 5. Pull latency with/without VM

This research was sponsored by US Army Research laboratory and the UK Ministry of Defence