Delay Tolerant Bulk Data Transfers on the Internet

Nikolaos Laoutaris, Georgios Smaragdakis, Pablo Rodriguez, Ravi Sundaram

Presented by Petko Georgiev

5 March 2013

Motivation

- There are important applications requiring exchange of Delay Tolerant Bulk data (TBytes):
  - CERN's Large Hadron Collider
  - Large data centers
  - Rich media transfers

- Current solutions:
  - Expensive dedicated networks
  - Postal service

Goals

- Use existing commercial ISP capacity infrastructure
- Avoid increasing transit cost
- Avoid reducing QoS for interactive traffic

Contributions

- Ideas:
  - Transmit during off-peak hours
  - Do not impact charged volume of sender's/receiver's access ISP
- Two situational policies that perform DTB data transfers for free (or at minimized cost)
- Performance analysis
- Cost analysis

Network Model

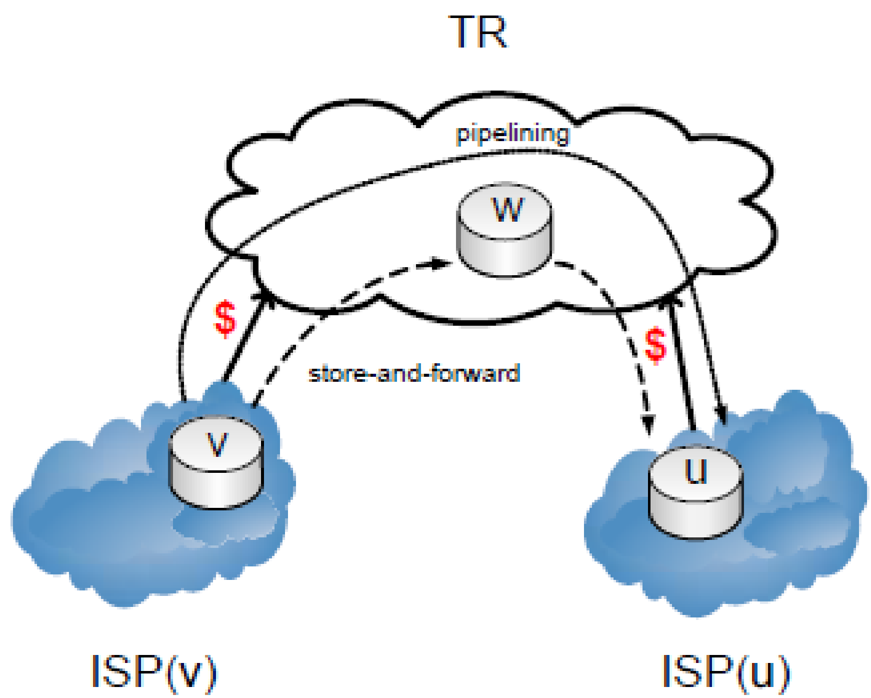
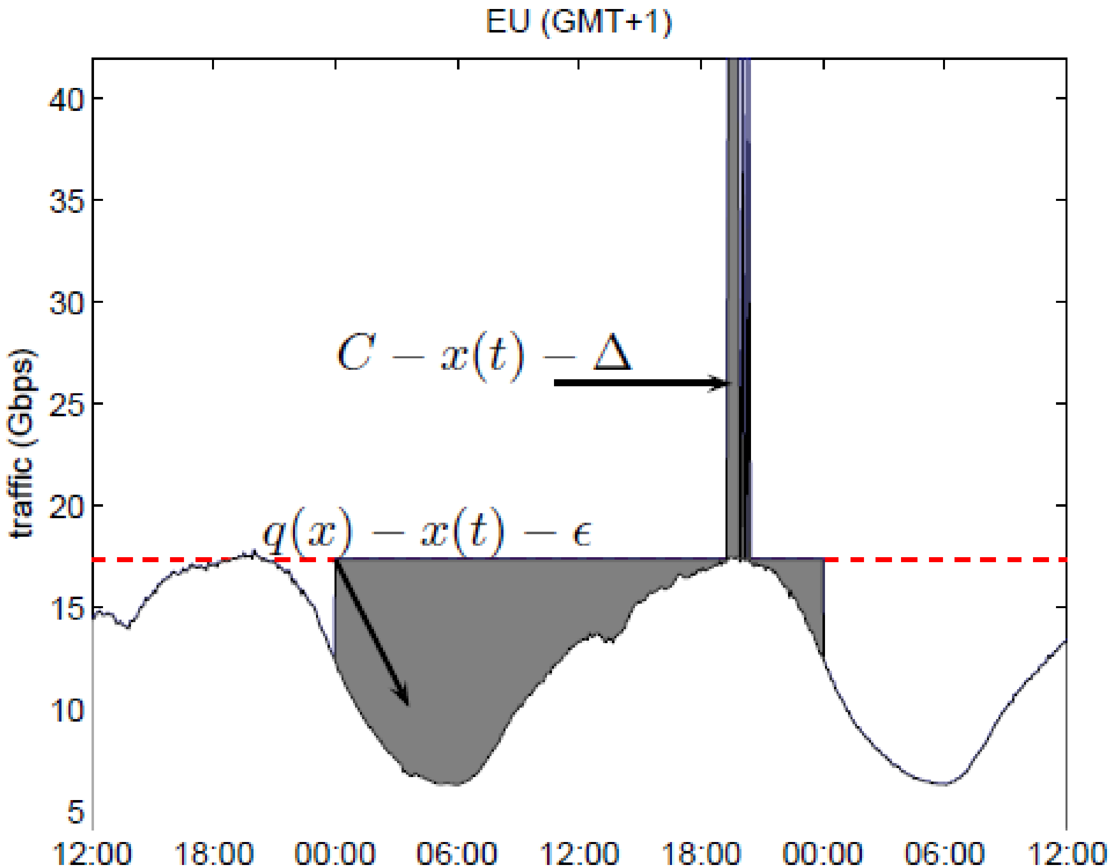


Figure 1: Sender (u) and receiver (v) of DTB traffic

Idea

- 95-percentile pricing
  - x ≡ time series of 5-minute transfer volumes
  - Charged volume: $q(x)$ ≡ 95-percentile value of x
- Pricing is determined by peak transfers and NOT total volume!
- Use off-peak hours to fill capacity with DTB

Approach: Water Filling



C ≡ capacity

x ≡ time series of volume transfers

t ≡ time slot

f(C, x, t) ≡ DTB data

$$f(C, x, t) = \begin{cases} q(x) - x(t) - \epsilon, & \text{if} \quad x(t) < q(x) \\ C - x(t) - \Delta, & \text{o.w.} \end{cases}$$

Approach: Water Filling (cont'd)

$$F(C, x, t_0, T) = \sum_{t=t_0}^{t_0+T-1} f(C, x, t)$$

- Volume of DTB traffic pushed through a charged link of capacity *C* carrying background traffic *x* in the interval $[t_0, t_0 + T)$ without increasing its charged volume *q(x)*

Policy: End-to-End with Source Scheduling

- Policy is essentially pipelining
- Respect both sender's and receiver's charged volume

$$F(\text{E2E-Sched}) = \sum_{t=t_0}^{t_0+T-1} \min\left(f(C_v, x_v, t), f(C_u, x_u, t) \right)$$

Policy: Store-and-Forward

- Independent water-fillings in the two charged links:
  - *ISP(v)* → *TR* (sender uplink)
  - *TR* → *ISP(u)* (receiver downlink)

$$F(\mathrm{SnF}, t) = F(\mathrm{SnF}, t-1) + f(t), \quad t_0 \leq t < T$$

$$f(t) = \begin{cases} f(C_u, x_u, t), \text{ if } \ f(C_u, x_u, t) < f(C_v, x_v, t) \\ f(C_v, x_v, t) + \min(f(C_u, x_u, t) - f(C_v, x_v, t), b_w(t-1)), \\ \qquad \text{o.w.} \end{cases}$$

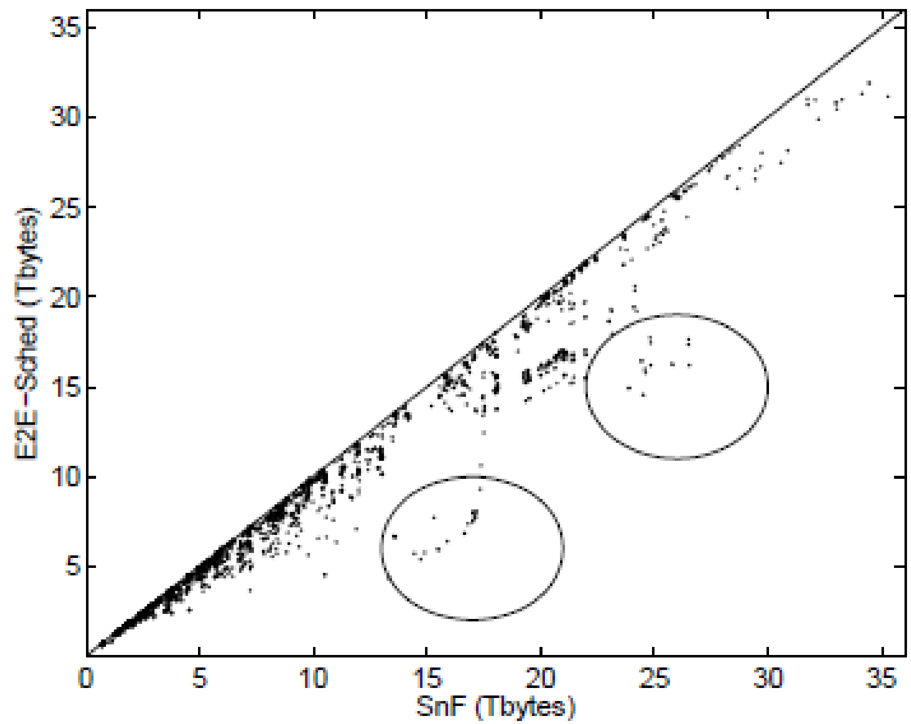
Policies: Meeting Deadlines

- **Problem:** The target volume B cannot be sent for free within time T

- **Solution:** Solve an optimization problem
  - Find charged volumes $q_v > q(x_v)$ and $q_u > q(x_u)$ to minimize the extra transit cost $c_v(q_v) – c_v(q(x_v)) + c_u(q_u) – c_u(q(x_u))$

  - Predict x based on daily traffic patterns of previous week

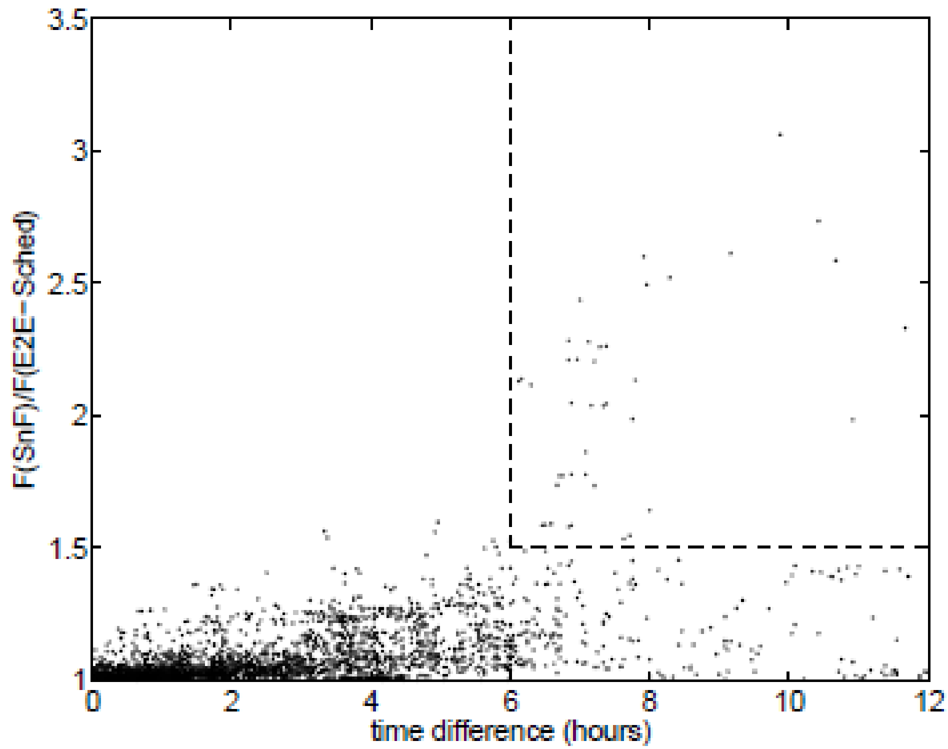
Results: Free DTB Transfers



(a) Daily free volume, E2E-Sched, SnF.

Figure 3a: *F(E2E-Sched)* vs. *F(SnF)*
280 links with *C* > 1 Gbps

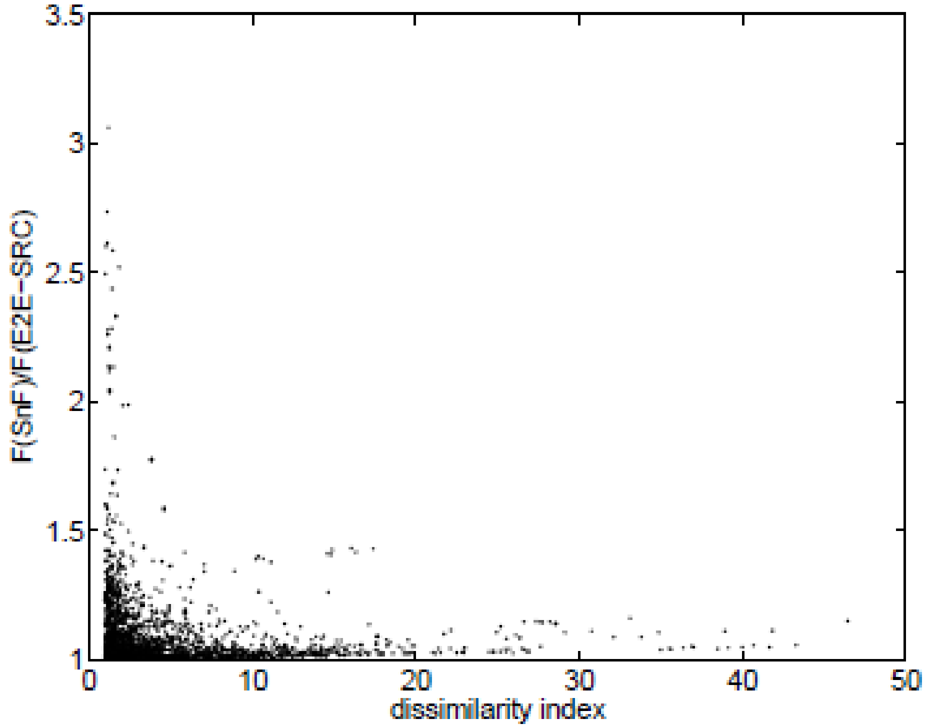
Results: Time Zone Difference



(b) Free volume ratio vs time-zone diff.

Figure 3b: *F(E2E-Sched)/F(SnF)* ratio as a function of the time difference between sender and receiver

Results: Off-peak Capacity Differences



(c) Free volume ratio vs dissimilarity

Figure 3c: Influence of off-peak capacity dissimilarity on SnF performance

Results: Advantage of SnF Policy

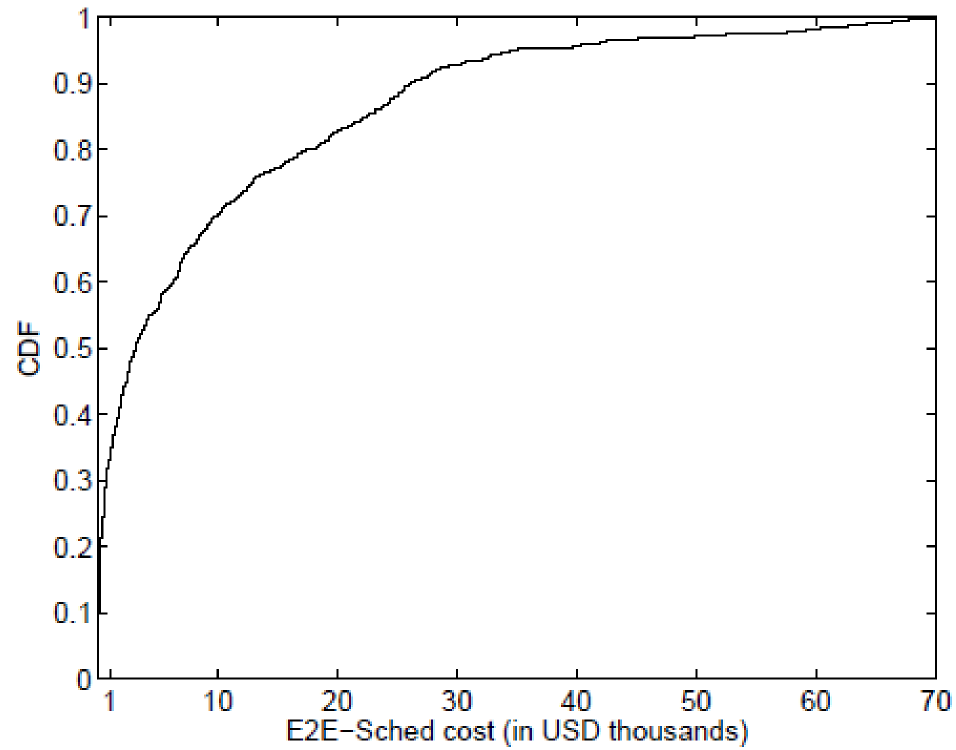


Figure 6: Transit cost paid by **E2E-Sched** to match the volume that **SnF** delivers for free.

Results: Storage Node Cost

- Cost of storage server and maintenance amortized to less than 1K $ per month

- Cost of E2E Scheduling around 60K $ per month

Results: SnF vs. Courier Services

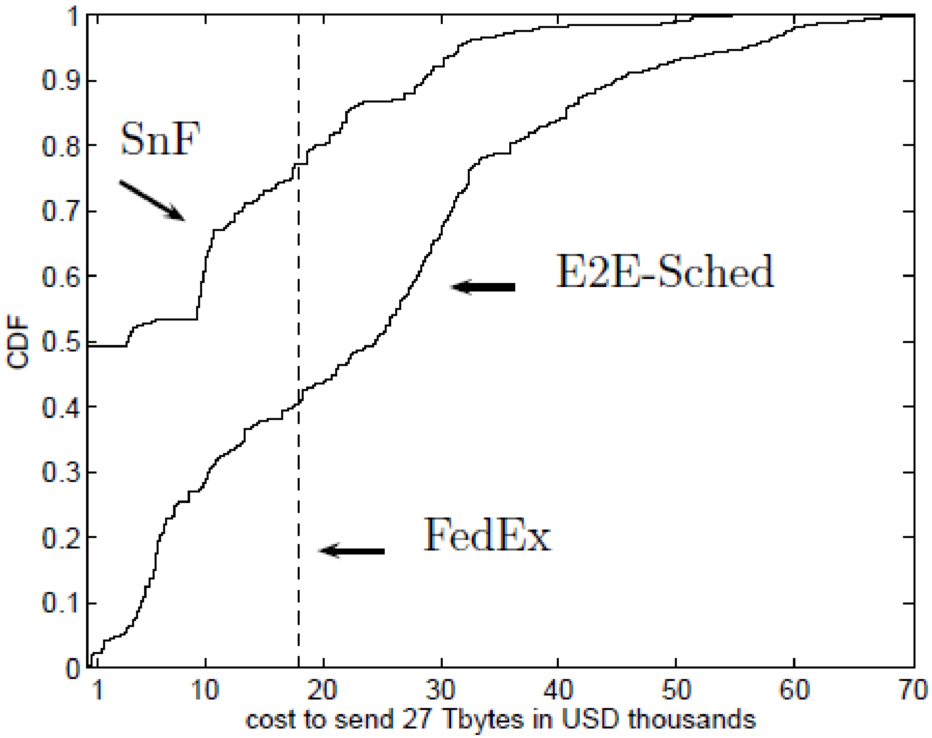


Figure 9: The cost of sending 27 Tbytes.

Compute amortized daily cost for all sender-receiver pairs

Compute daily cost of shipping hard drives (FedEx services)

Summary

- If *B* is the target DTB data volume, then
  - *B < F(E2E-Sched)*: transfer for free
  - *F(E2E-Sched) < B < F(SnF)*: transfer at zero *transit* cost (but pay for storage)
  - *F(SnF) < B*: SnF can minimize transit cost

Related Work

- QBone Scavenger Service
  - does not protect from high transit cost
  - does not guarantee delivery under deadlines
- Slurpie protocol (application layer)
  - suitable for one-to-many distribution
- Mobile networks
  - scheduling differs because cost is not considered
  - nodes are mobile and not static

Future Work

- Data encoding
- Error recovery
- Multiplexing concurrent DTB jobs
- Utilizing multiple up/down links for transfer
- Survey of how changing market policies will affect the applicability of the model

Criticism

- Advantage of SnF seems situational
  - Time zone differences of > 5 hours
  - Comparable off-peak capacities

- Storage node deployment
  - Is it really *never* a bottleneck?
  - How is it positioned to avoid triangular routing?
  - Single point of failure

Criticism

- Very few details on the “load valleys”
- More example DTB transfer volumes needed
- Not all pricing information is transparent
  - E.g. server and maintenance cost estimation

Overall impressions

- A nice simple idea of water-filling
- It is hacking the traffic volume charging
- A lot of evaluation scenarios covered
- Needs future work to be production ready