PREFERENCE DRIVEN SERVER SELECTION IN PEER-2-PEER DATA SHARING SYSTEMS

AbdelHamid Elwaer, Ian Taylor and Omer Rana
Cardiff School of Computer Science
Cardiff University, UK

Presented by Ian Taylor
Reader @ Cardiff

OUTLINE

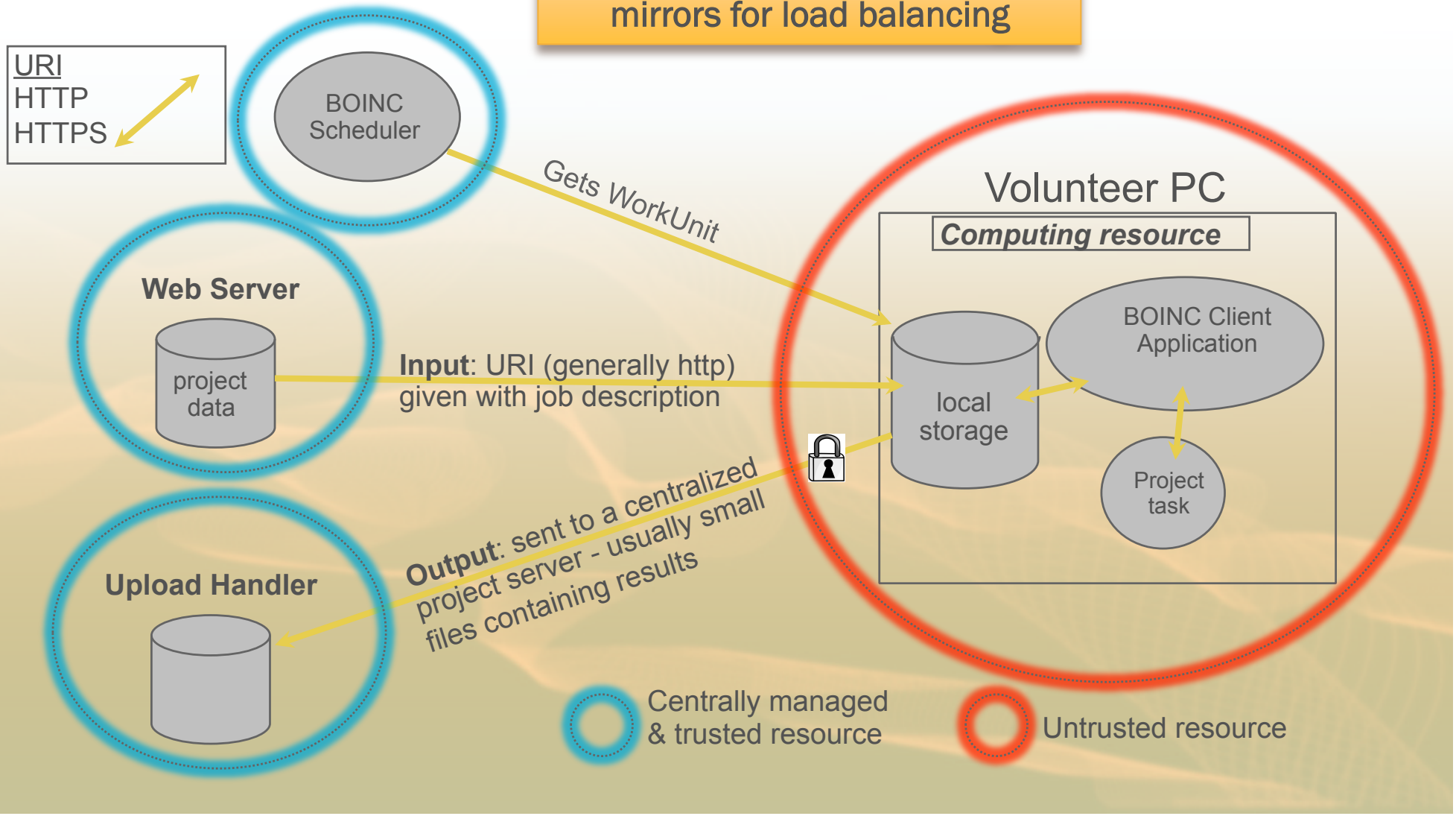
- Desktop Grids, Data Management and P2P Systems
- Attic File System
- Baseline Results – compare Attic with BOINC
- Trust Framework, background into trust
- Experimental Environment
- Performance Results
- Conclusion

DATA DISTRIBUTION IN VOLUNTEER OR DESKTOP GRID COMPUTING

- Projects such as Einstein@HOME and the previous SETI@HOME, people currently donate spare CPU cycles
  - But why not *have them also donate network bandwidth and share data with one another?*
- Community has shown support for such ideas
  - environment issues are critical in this space
  - The potential impact could be great, lowering administrative costs and overheads of running projects
- But the environment is more dynamic
  - there is more transient connectivity and such fluctuations in server availability and its more prone to attack
- We address these issues here through our self-adaptive trust framework.

BOINC DATA ACCESS

Data is generally provided centrally or by using managed mirrors for load balancing



SCALABILITY AND DATA ISSUES

1. Cost
   - volunteer paradigm should be upheld i.e. zero cost in hardware and admin
2. Security
   - Protect home user's machines when serving data
     - Need an opt-out system - compulsory open ports on all workers is not possible
     - Need a way or creating policies e.g. specifying trusted data centers
   - Protect the project's data
     - may want limited caching on a peer to limit exposure
     - need to ensure data integrity

EXISTING SYSTEMS

- There are obviously a number of commercial system e.g. Amazon's S3, which fail on cost.
- There are also a number of free P2P systems e.g. BitTorrent, Gnutella etc
  - they do not provide an opt out policy and authentication for specific nodes
- Hadoops HDFS , it is an open source counterpart of google's GFS
  - Already integrated with Condor and there are on-going discussions with BOINC. However
    - to date, no such integration exists
    - No framework for an opt out policy and authentication for specific nodes
- AtticFS addresses these concerns by
  - Creating a framework for specifying a trusted data server peers.
  - Verifying integrity of data
  - Plugs into existing systems e.g. BOINC and XtremWeb
  - Zero administration or additional hardware costs.

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

EPSRC
Engineering and Physical Sciences Research Council

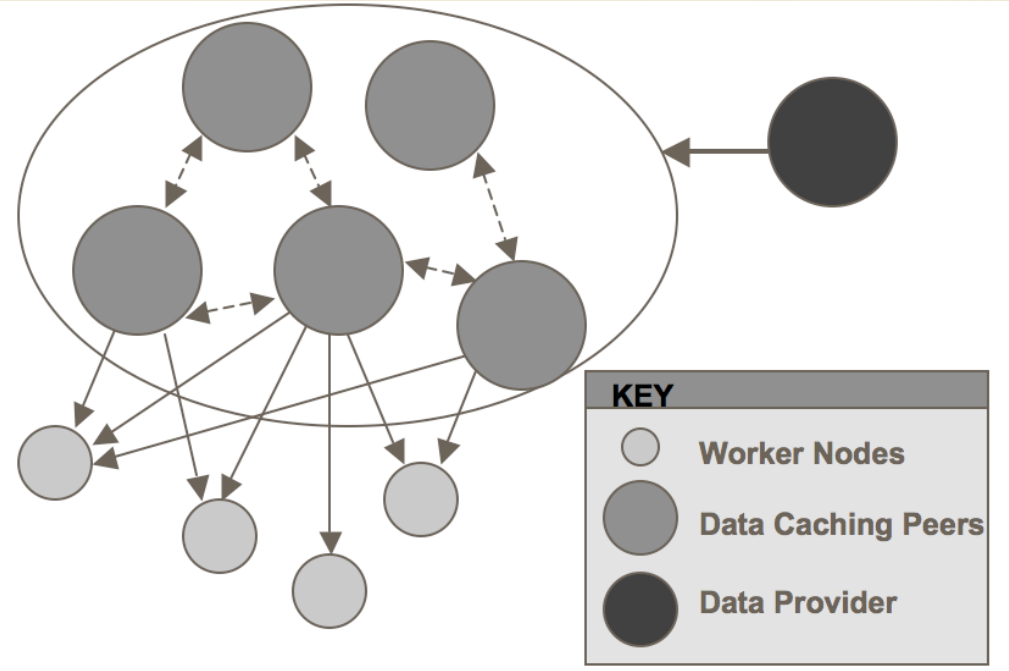
ATTIC

EDGeS

EDGI

Project Website: *http://www.atticfs.org*

- Started as part of a UK EPSRC proposal in 2005
  - User scenarios provided by Einstein@home
- Continued under EU FP7 EDGeS and EDGI projects



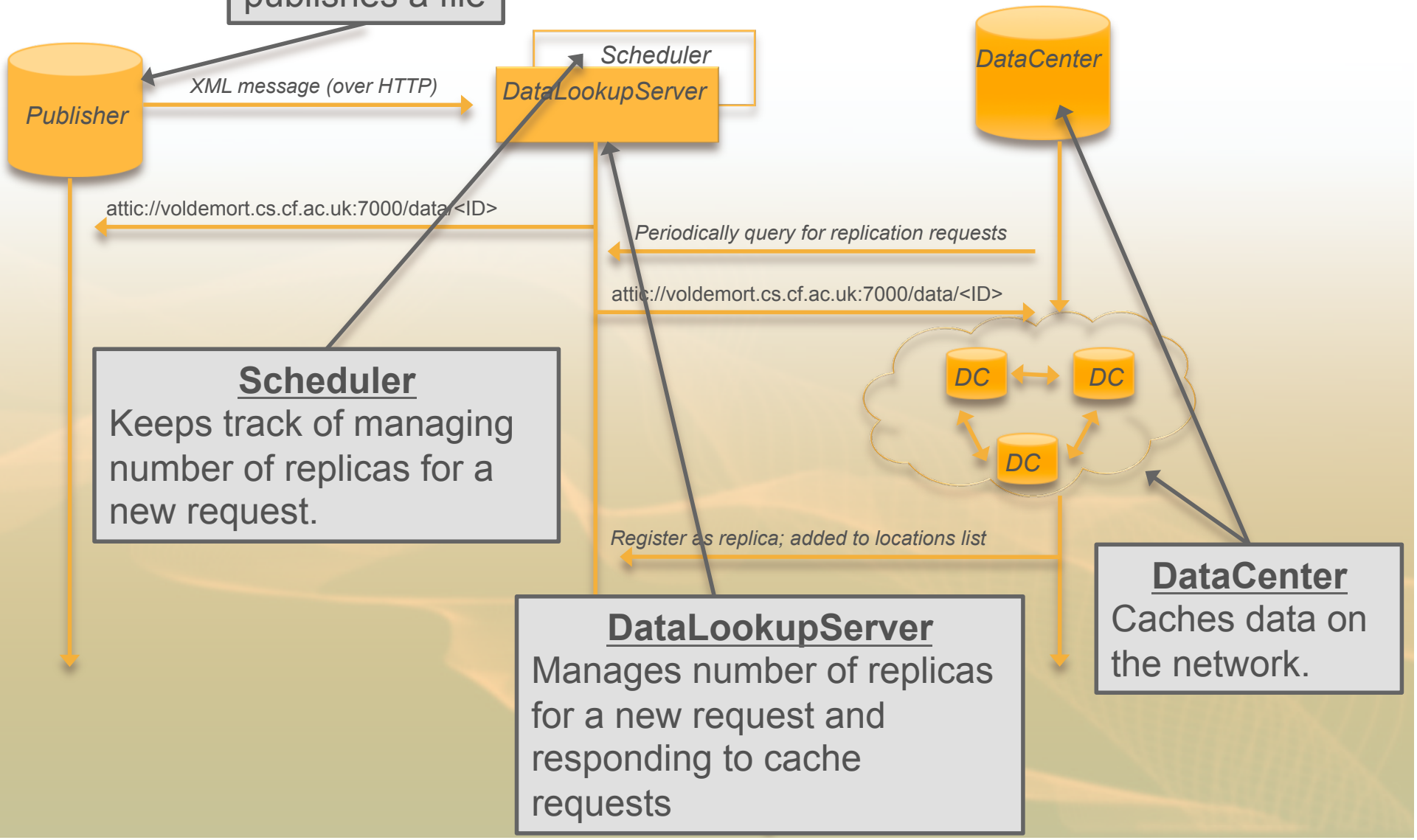
- Provides a dynamic layer of HTTPS-based data centers
- Data Caching peers exchange data amongst themselves and serve client machines

ATTIC

- Data can be published to data centers
- Files can be split into individual chunks for distribution
- Clients download from multiple data centers (like bittorrent) or can download different files from different data centers - scenario dependent.
- Each data center can have a security policy e.g. X.509 trusted entities - static
- Or you can override this as we have to **automate the assigning of trust** - dynamic

WHAT'S IN THE ATTIC?

Publisher
Any entity that publishes a file



Scheduler
Keeps track of managing number of replicas for a new request.

DataLookupServer
Manages number of replicas for a new request and responding to cache requests

DataCenter
Caches data on the network.

A TRUST MODEL

- If users now support data provisioning then that **data can become corrupted**
  - with or without the intent of the volunteered resource owner
- Data centers can also have **different upload speeds** and their **availability can change** over time.
- The **key research question** is to enable a client to decide which data centre to download data from given the dynamic nature of the network
- We propose here the use of a **trust model** to assist clients in the selection of the data center most aligned with their preferences.

TRUST BACKGROUND

Previous work on trust:

- Using prior interaction experience
  - Use prior interaction history to rate other providers
  - Witkowski et al., Sabater et al. – e.g. the "REGRET" system
- Information gathered from others (aka. "Recommendation")
  - Based on ratings provided by others
    - Need to also account for "unreliable" recommendations
    - Use of a connectivity graph between recommenders and hash functions to chose multiple raters
  - EigenTrust, Google PageRank, PowerTrust are examples

We also make use of historical data and consider community feedback to assess trust (i.e. a recommendation from others). We use a particular feedback format that can be re-used in a number of other contexts.

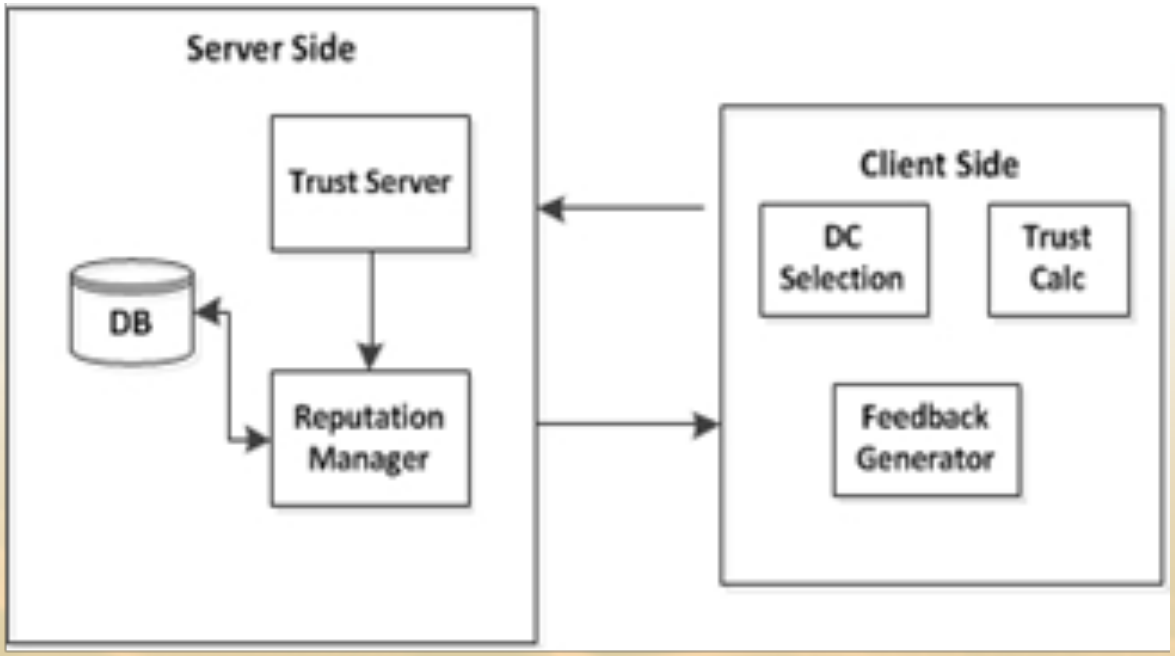
THE TRUST FRAMEWORK

- Trust is a metric to guide an Agent in deciding how, when, and who to interact with.
  - Where an agent can be either a user or service provider.
- To establish trust, an agent must gather data about their counterparts - this can be achieved in three ways:
  1. Using prior interaction experience.
  2. Information gathered from other agents.
  3. Socio-cognitive trust.

- This work focuses on characteristics 1 and 2.

TRUST FRAMEWORK

- The trust framework has client and server components .



- The clients generates feedback, processes trust values and selects data centers based on its preferences.
- The server collects clients feedback, updates the reputation database and provides this data to the clients.

UPDATING THE TRUST VALUES

- After a client completes downloading data, it provides a subjective assessment of each of the three metrics
-
  - Honesty: data integrity and quality, storage reliability and malicious data modification in-transit or at source
  - Availability: uptime, failure rate and resilience
  - Speed: access time, latency and effective bandwidth for each data center that has been used by this client.
- This public feedback can then subsequently be used by other clients, to support their decision about which data centers to trust, using the following equations

APPLYING BETA DISTRIBUTION

The assessment is calculated using an iterative beta distribution (see opposite) calculation. The equation calculates the degree of satisfaction (satisfied (r) or not satisfied (s))

$$E(p) = \mathbf{(r+1)/(r+s+2)}$$

Which is used by the client to calculate the three metrics. The total trust value is then calculated using:

$$T = a.TAvailability + b.THonesty + c.TSpeed$$

$$where\ a + b + c = 1.$$

The three weights are used to fine tune the clients preferences for which metric applies the most importance to them.

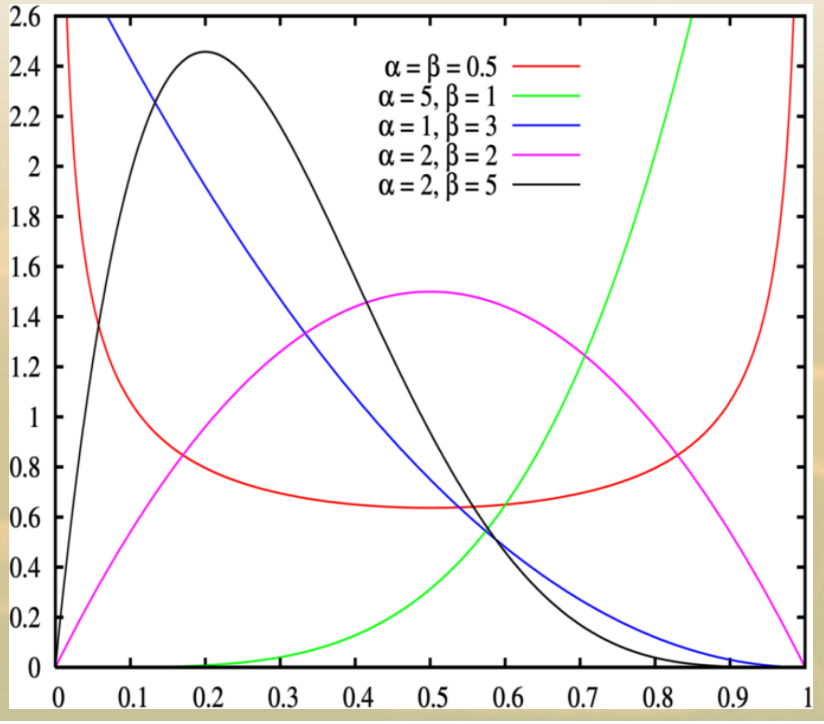
The Beta distribution:

$$\int (p|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\, p^{\alpha-1}\,(1-p)^{\beta-1},$$

$$where\ \ 0 \le p \le 1, \alpha \le 0, \beta > 0$$

The probability expectation value of the beta distribution is given by:

$$E(p) = {}^{\alpha}/_{(\alpha+\beta)} \qquad (1)$$



INTEGRATION IN BOINC

- We generate work units using attic url instead of http e.g. **`attic://dls.org/1234`**
- The BOINC client has been modified to use Attic worker when the download url of the input file starts with <attic>.
- The BOINC clients uses AtticFS when the url includes <attic>
- The BOINC clients will contact the lookup server to get a list of data centers.

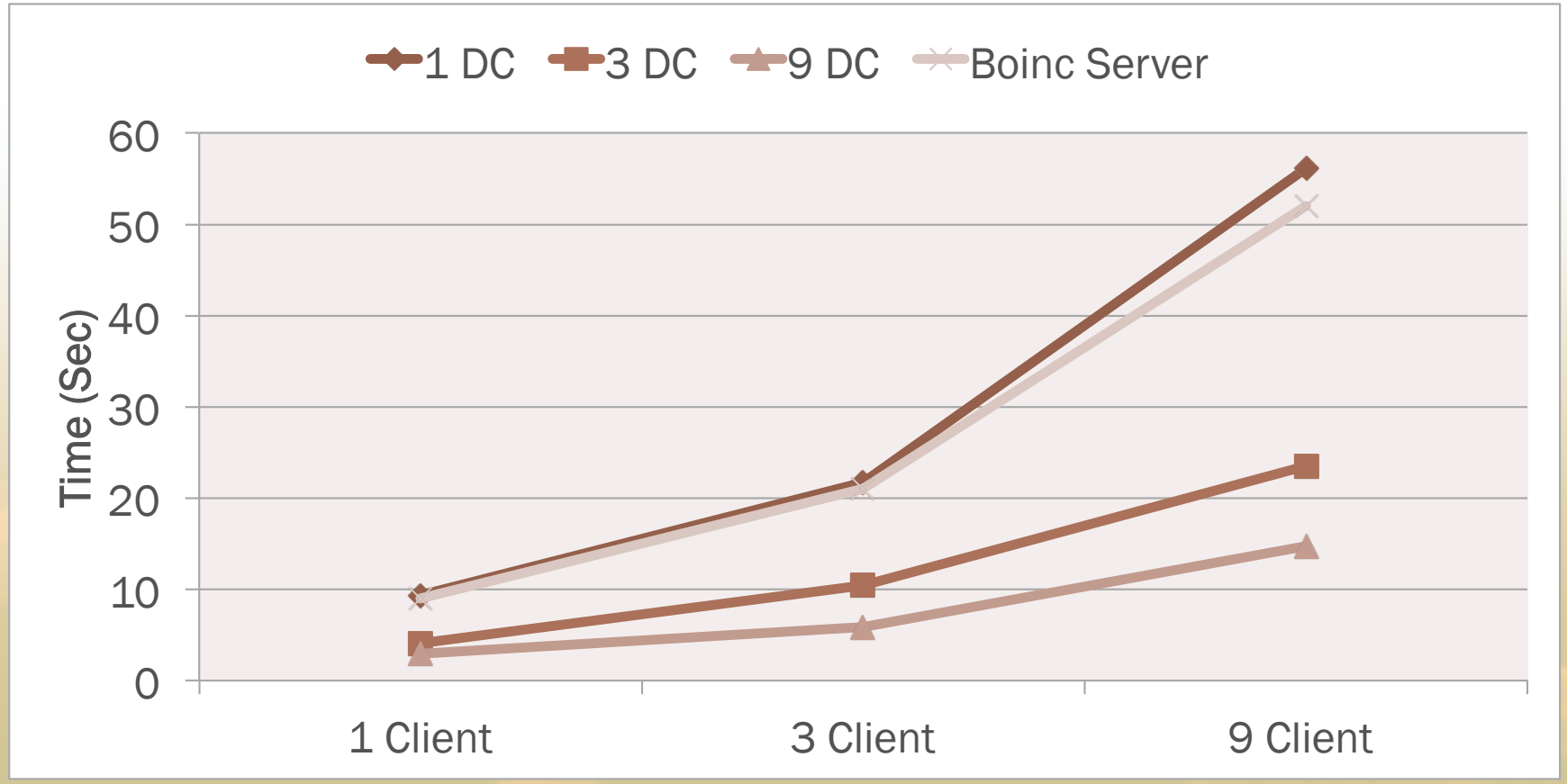
EXPERIMENT ENVIRONMENT OVERVIEW

- 33 Linux machines were used to run various combinations of clients and data centers.
- The network connection speed of a subset of the machines were set to 10 Mbps and others to 100 Mbps
  - We switched the networks by changing the speed of the the sockets that the machines were connected to the network with (admin utility for the school)
  - We wanted to limit the bandwidth to emulate the restricted bandwidths of a home user on the internet and the different between download and upload speeds e.g. typically most for ISPs, you can download more than 10 times faster than you can upload.
- A Poisson Distribution was used to simulate the availability of data centers.

BASELINE COMPARISON WITH BOINC

- This experiment makes a comparison between using BOINC server and the AtticFS to download a 10 MB file.
- The file is downloaded 3, 6 and 9 clients concurrently
- Using 3, 6 and 9 data centers.
- The same clients (3,6 and 9) were used to download concurrently the 10 MB file using the BOINC server.
- Servers had a 10Mbit max download speed

BASELINE COMPARISON WITH BOINC:

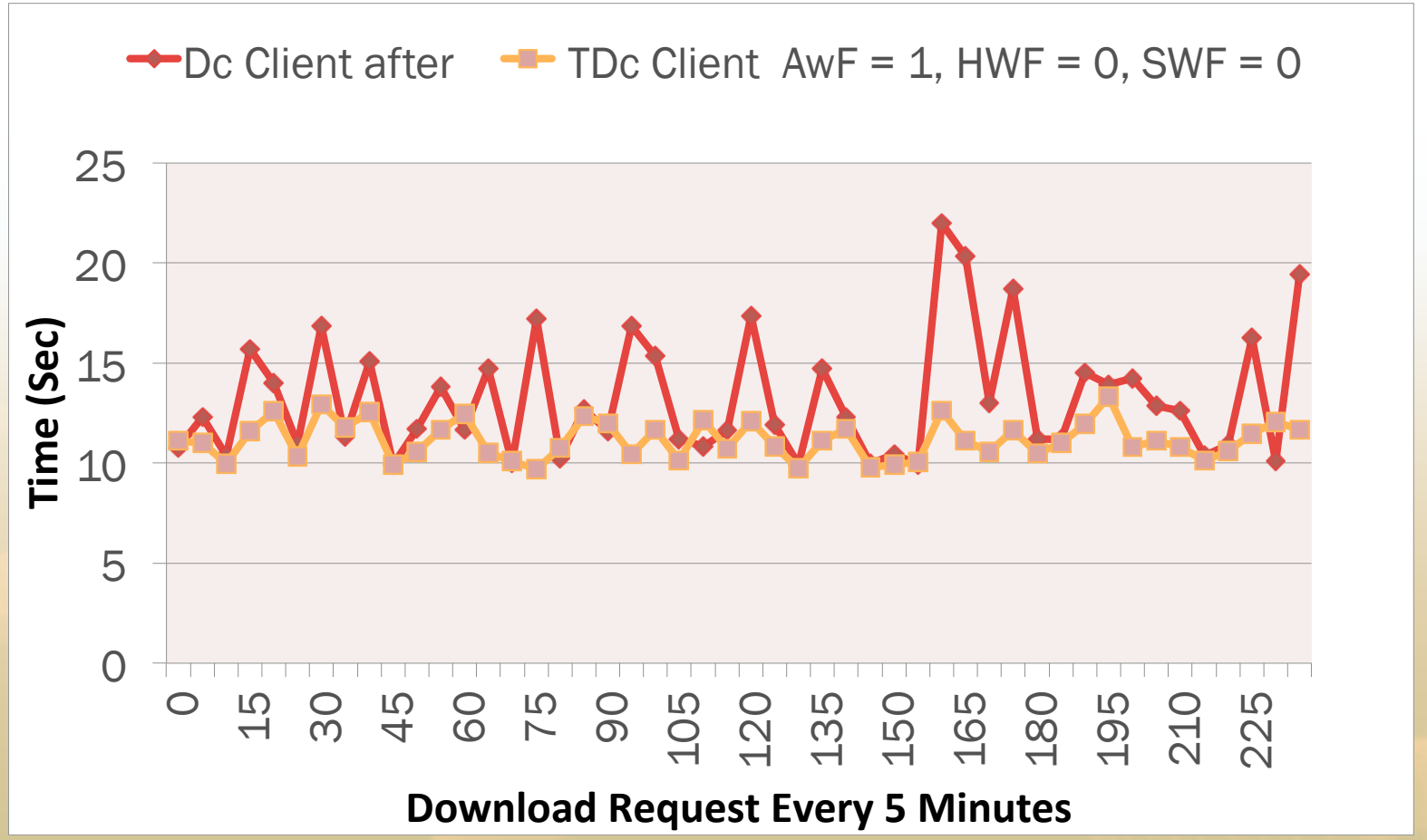


Comparing BOINC Server with the Attic file system

EXPERIMENT 1: *DATA AVAILABILITY*

- Shows the effect of **data center availability** on download efficiency.
- 10 MB file was published to AtticFS (10 data centers).
- The 10 data centers have 10Mb/s connections and are all honest peers.
- A comparison was made between the AtticFS client with and without the trust framework.
- The experiment lasted eight hours.

EXPERIMENT 1:



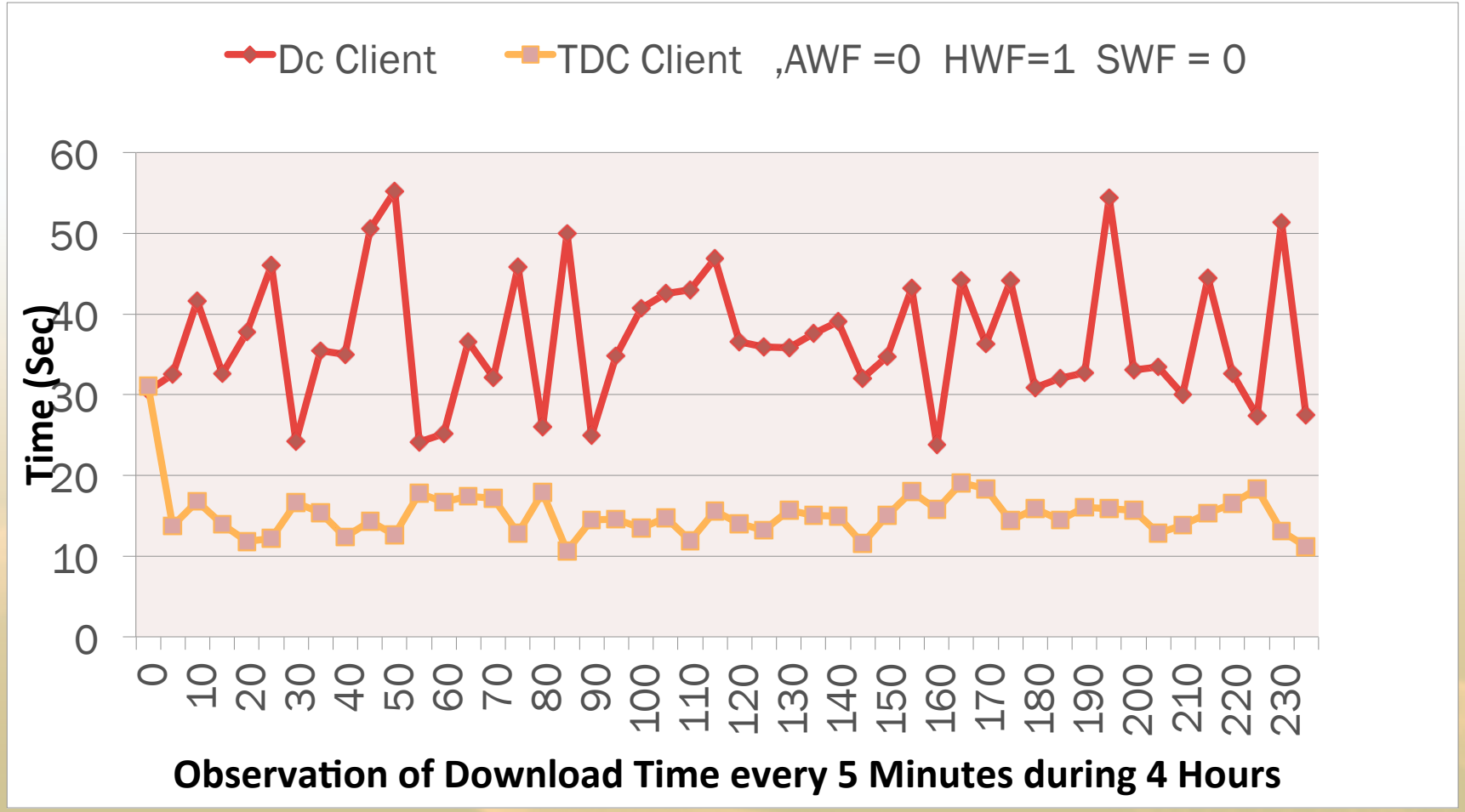
Shows an improvement on the download time in the next four times using our trust model.

EXPERIMENT 2: TRUSTED PEERS

- Show how **malicious behaviour** of data centers can be avoided.
- 10 MB file was published in the AtticFS .
- Six data centers are used in this experiment.
- Three of them honest data centers and the other three are malicious.

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

EXPERIMENT 2:

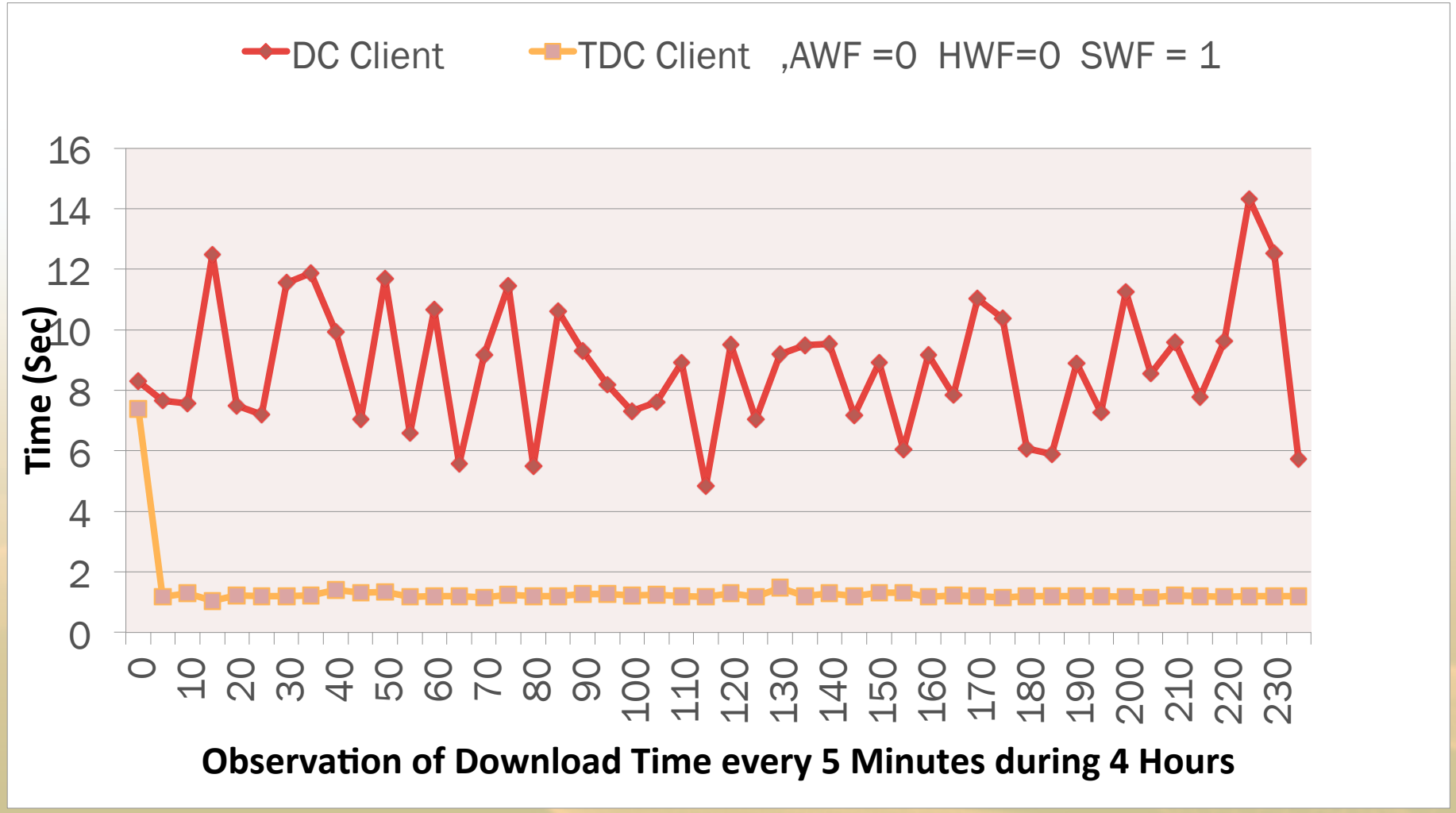


Shows that our trusted clients has significantly better download time.

EXPERIMENT 3: MAXIMISING BANDWIDTH

- Shows how the trust framework can be used to choose data centers with the highest bandwidth connections.
- 10 data centers are used.
- Six data centers have 10 Mb/s connections and four have 100 Mb/s connections.

EXPERIMENT 3:

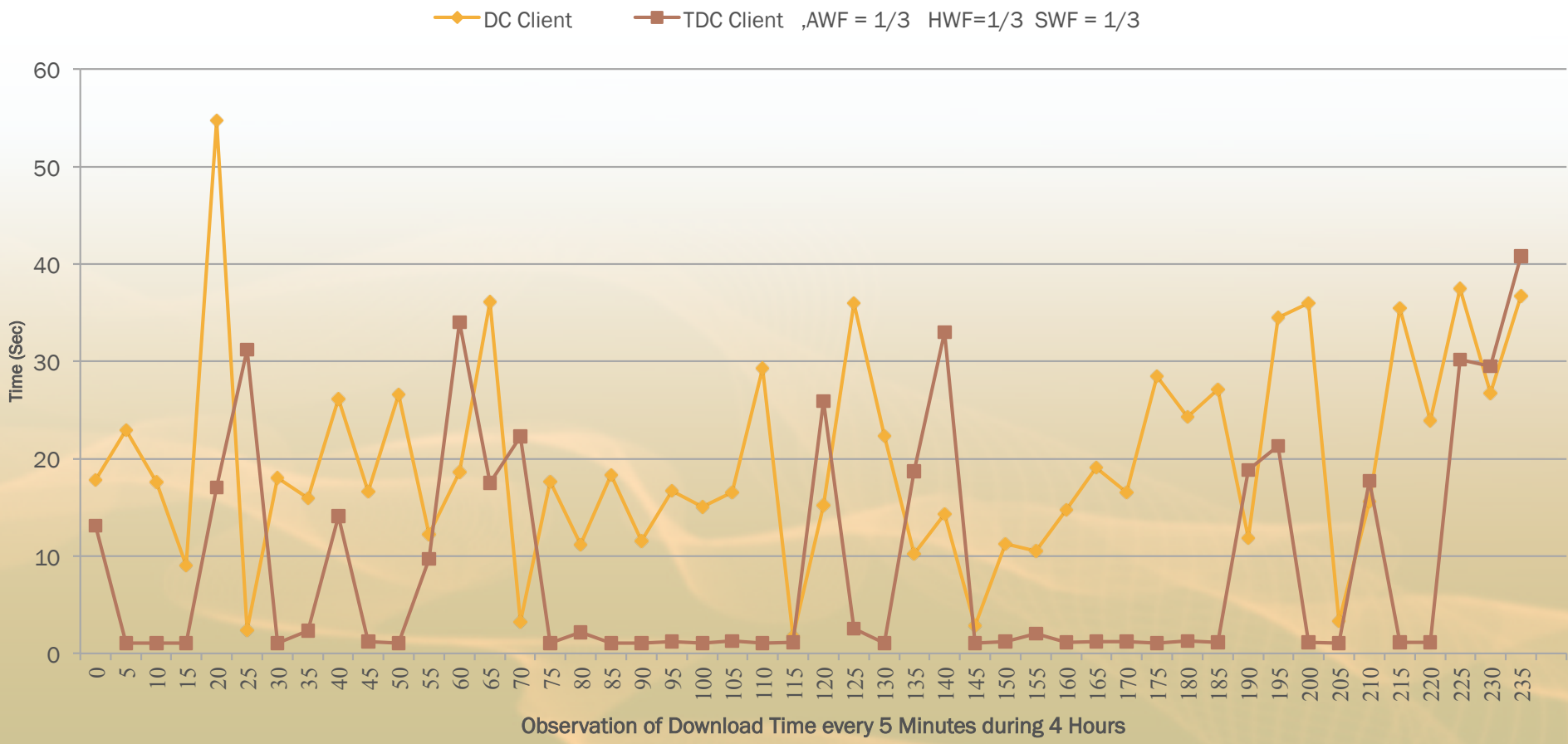


Fig(5) shows that the download time improves by clients making use of our the trust framework.

EXPERIMENT 4: USING ALL FACTORS

- Shows effect of the three factors combined (speed, honest and availability)
- 10 data centers are used
  - six data centers 10MB and four data centers having 100 Mb.
- Three of the data centers act maliciously
- Availability of all data centers changes over time according to a Poisson distribution.

EXPERIMENT 4:



CONCLUSION

- The results shows a significant improvements to a client's download time when the trust framework is applied.
- The trust framework offers to the clients the possibility to choose the data centers based on their preferences.
- This framework shows how data centers reputations can be used to calculate their trust value.
- Future work includes empirical analysis of the effect of user preferences for maximum efficiency
- Thanks to
  - Andrew Harrison for Attic development
  - Ian Kelley from Cardiff EDGI for Attic Support
  - Funding from EPSRC, EDGeS and EDGI