# Design Space Analysis for Modeling Incentives in Distributed Systems

by Rameez Rahman, Tamas Vinko, David Hales, Johan Pouwelse, and Henk Sips

**Delft University of Technology**

# Incentives in Distributed Systems

Consider a P2P file sharing system, such as BitTorrent:

- *Collective interest*: upload to others so everyone gets the file quickly
- *Individual interest*: save bandwidth by only downloading and hence free-riding on others
  - Need to tackle freeriding in some way

→ Requires an incentive scheme.

- How do we evaluate how good the incentive scheme is?

# Traditional vs Our Approach

Brainstorm /Intuition

Protocol Variant

Game Theoretic Analysis

Predicted Outcomes

Specify Design Space

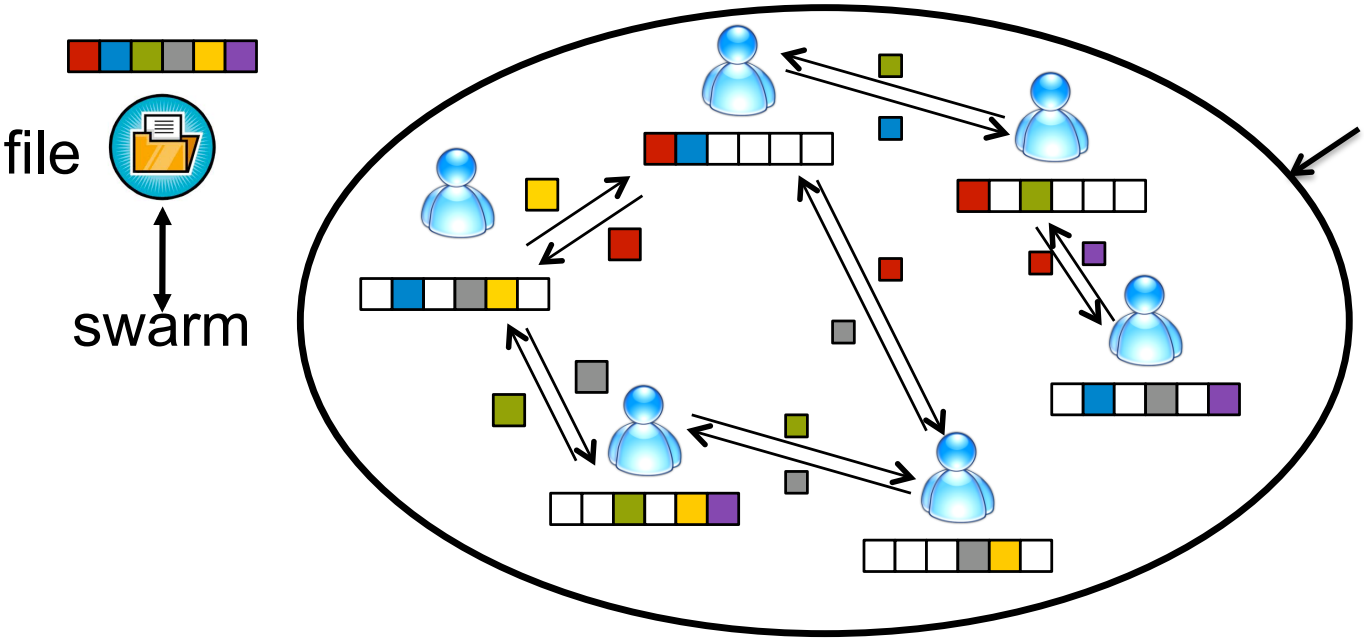
Many Protocols

Simulation Based Analysis

Predicted Outcomes

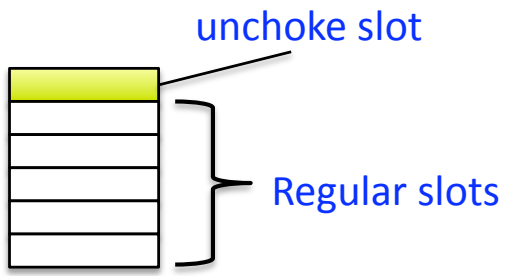
# We consider BitTorrent like file swarming systems

- A popular P2P file sharing system
- Hundreds of millions of users, and a large fraction of Internet traffic
- A key of BitTorrent's success: Tit-For-Tat (TFT) incentive policy

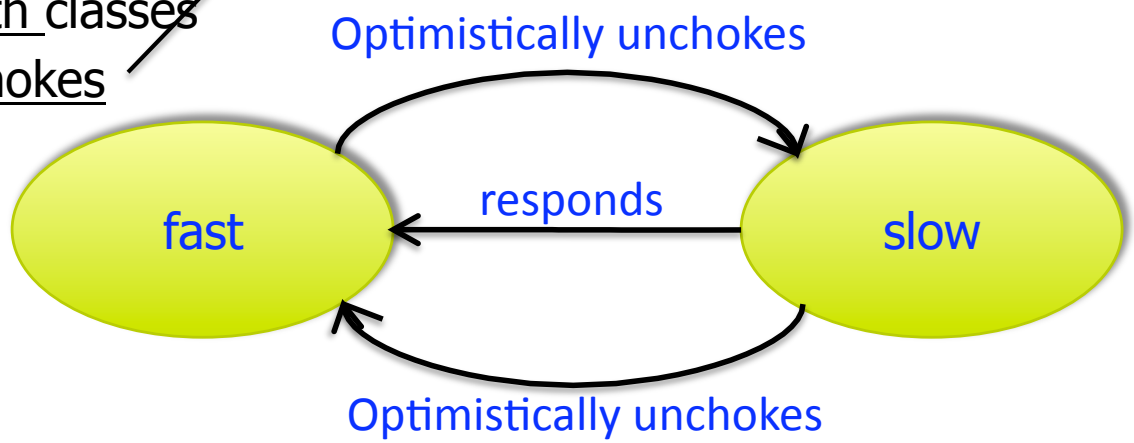



Peers exchanging file pieces with each other using a rate based TFT approach

# Our Approach



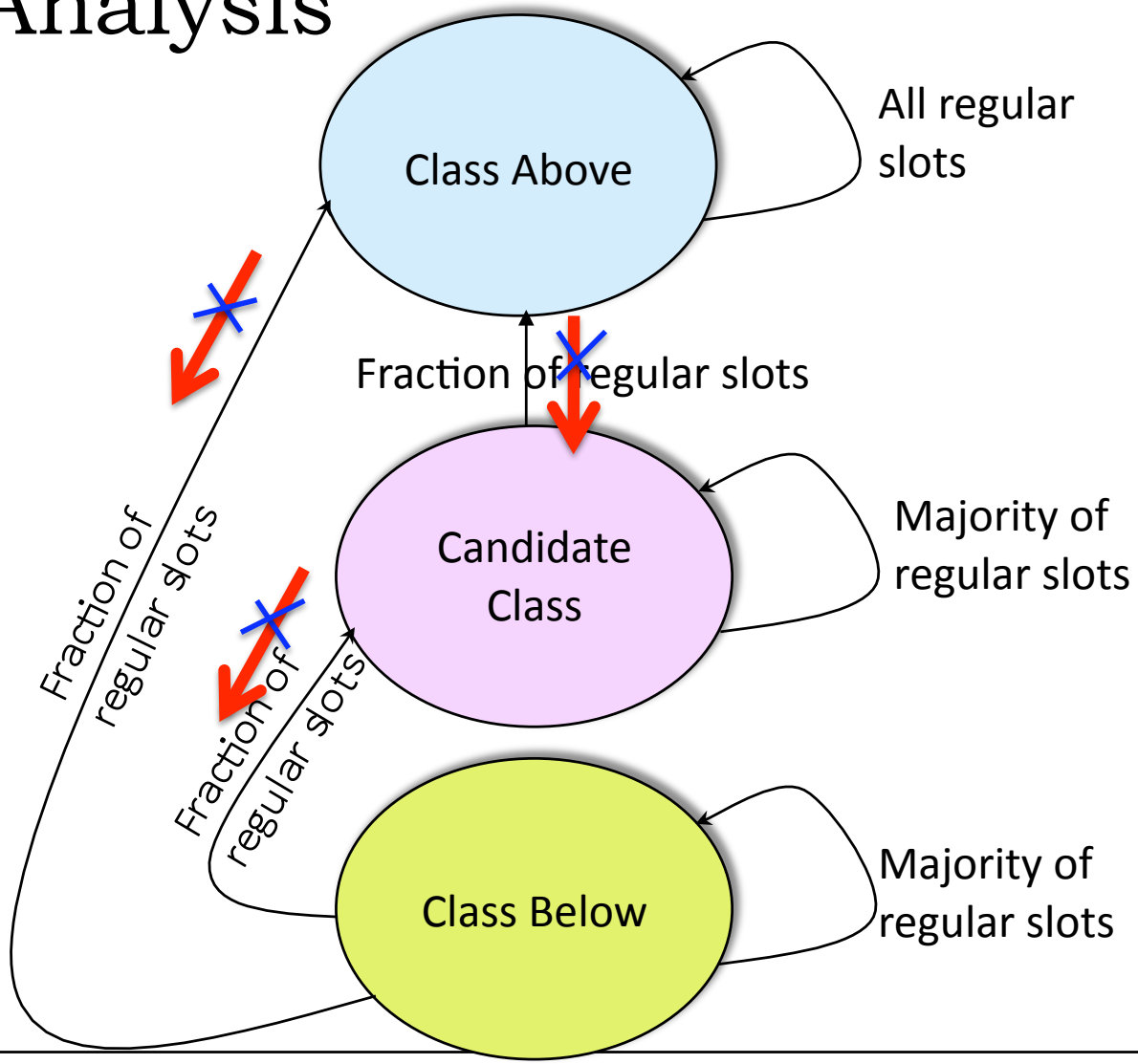

- First, a game theoretic analysis of BitTorrent, based on heterogeneous bandwidth classes
- We model the repeated aspects of the protocol. Also, we use different abstractions than in previous work
  - heterogeneous bandwidth classes
  - modeling optimistic unchokes



# Three Class Analysis

- **Optimistic unchokes (not shown in the figure) are nearly uniformly distributed over all classes**

**Higher classes do not reciprocate to the "Fraction of regular Slots**



# Results

- BT is not a Nash Equilibrium (unlike previous findings)
- Considering BT as a strategy in a game allows us to build a robust BT variant called **Birds**
  - Birds sorts on the basis of proximity to its own upload speed
  - Birds is a Nash Equilibrium
  - A recently released BT client called BitMate is very similar to Birds

# And now?

- Game theoretic analysis (like most modeling techniques) needs a high level of abstraction
- Different abstractions may lead to different and even contradictory results.
- We should remember that the BT variants BitThief, BitTyrant came only after it had been proved that BT is a Nash!

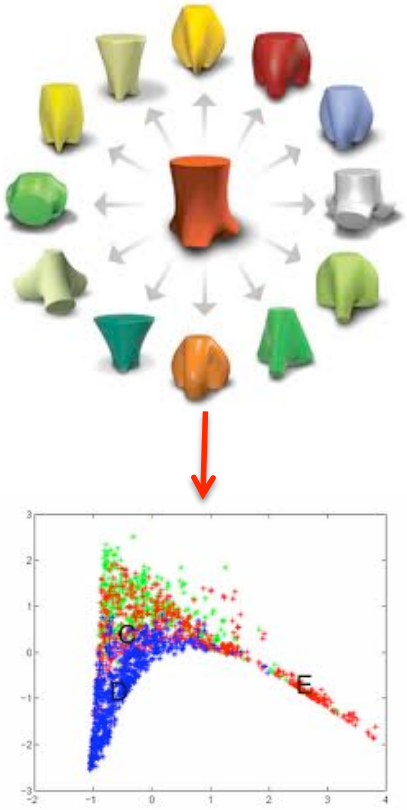
# Open Questions

- If we would include more details, would our Birds analysis still hold? Would we come up a variant "***Bird Flu***", that aims to exploit Birds.
- How robust is Birds anyway, or any protocol that one might devise?
- Did we model everything? What did we not model? Resource allocation, Candidate list, different Selection functions…

*Maybe it is time for an approach that augments/complements game theoretic approach?*

# Our Approach: Design Space Analysis (DSA)

- Apply Axelrod-like tournament approach to evaluate realistic P2P protocol variants
- Interesting bit is:
  - Break down of protocols into a design space
  - Evaluation of protocol variants (PRA)
- Specific application to BitTorrent protocol variants



# The Three Elements of DSA

1) Flexible behavioral assumptions

2) Specification of the Design Space
   - Parameterization
   - Actualization

3) Systematic analysis of the Design Space

# Flexible Behavioral Assumptions

In DSA, protocols may, in the words of Axelrod:

> "*simply reflect standard operating procedures, rules of thumb, instincts, habits, or imitation*".

This in contrast to the usual rational framework assumption of traditional game theoretic analysis

# Design Space Specification (1)

**Parameterization:** identify salient dimensions

E.g. for gossip protocols:

1) Selection function for choosing partners
2) Periodicity of data exchange
3) Filtering function for data to exchange
4) Record maintenance policy in local db

# Design Space Specification (2)

**Actualization:** specify values for the identified dimensions

E.g. for ‘selection function’ for gossip Protocols:

1) Choose partners randomly
2) Choose partners based on similarity
3) Choose partners who have given best service
4) Choose loyal partners…

And so on…

# PRA characterization of a protocol $\pi$

- ***Performance*** - the overall performance of the system when all peers execute $\pi$ (where performance is determined by the designer)
- ***Robustness*** - the ability of a majority of the population executing $\pi$ to outperform a minority executing a protocol other than $\pi$
- ***Aggressiveness*** - the ability of a minority of the population executing $\pi$ to outperform a majority executing a protocol other than $\pi$

# More detail on PRA

- ***P*** = average download time
- ***R*** = number of "wins" in round robin tournaments against all other protocol variants
- ***A*** = number of "wins" in round robin tournaments against all other protocol variants
- *P,R,A* values are normalized over the space

# Parameterizing of a P2P protocol

- Peer Discovery
  - *Timing and nature of the peer discovery policy*
- Stranger Policy
  - *How to treat newcomers*
- Selection Function of known peers
  - *E.g .past behavior (through direct experience or reputation system), service availability, and liveness criteria*
- Resource Allocation
  - *The way a peer divides its resources among the selected peers*

# Actualizing BT like file-swarming protocols

- Stranger policy (10 variants)
- Selection function:
  - Candidate list - peers to consider (2 variants)
  - Ranking function - order list (6 variants)
  - Selection - number of peers to select (9 variants)
- Resource allocation (3 variants)

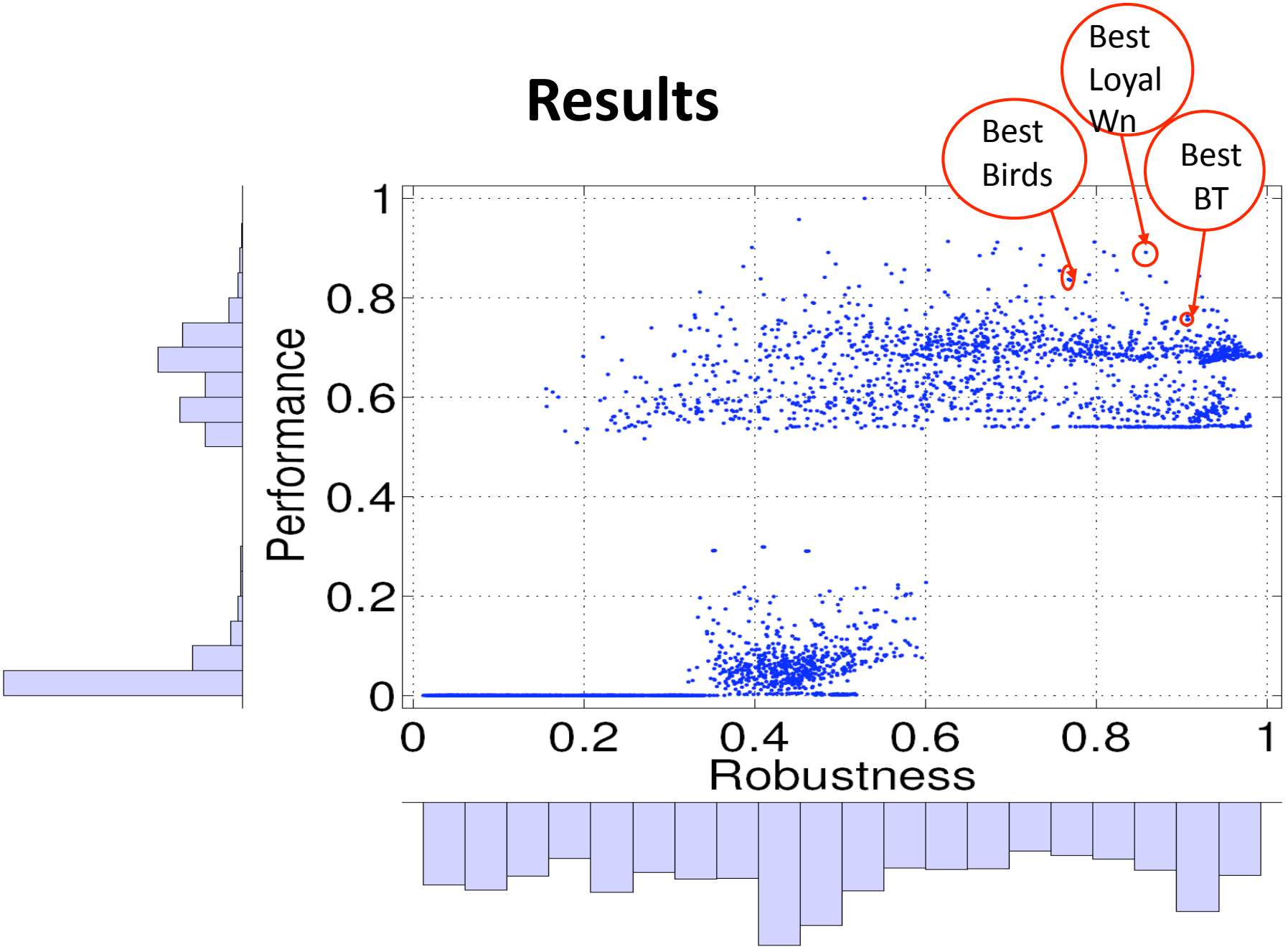
*Gives a space of* **3270** *unique protocols*

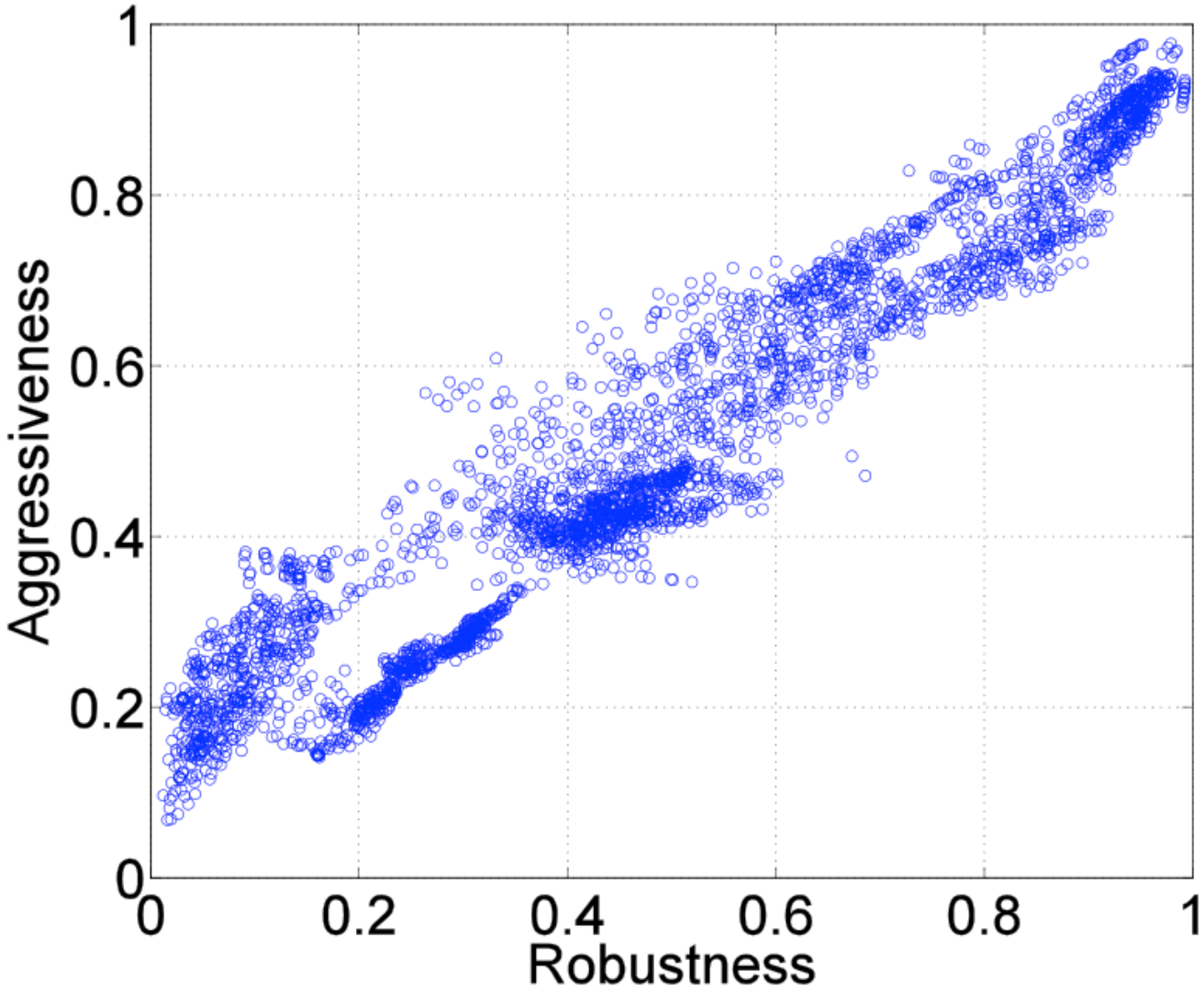
# Methodology of conducting DSA

- 50 peers, that interact with each other for 500 rounds.
  - Bandwidth distribution taken from Piatek et al. [NSDI 2007]
- For Performance, 100 runs for each protocol $\pi$.
- For Robustness, each protocol $\pi$ against all other 3269 protocols. 10 runs for each such encounter. 0.5 $\pi$ and 0.5 $\pi^*$
- For Aggressiveness, same as above. But with 0.1 $\pi$ and 0.9 $\pi^*$

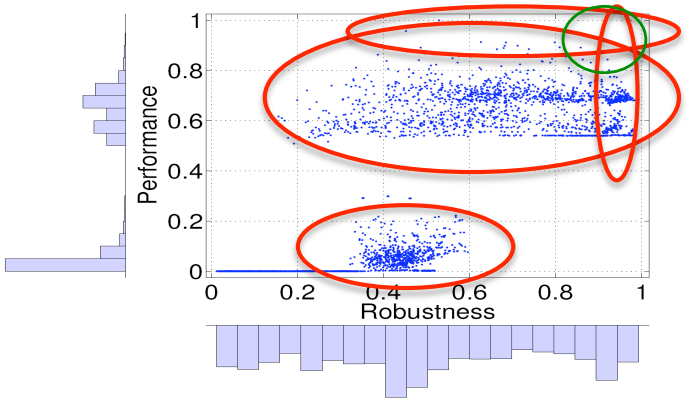
*This comes to* **107 million** *runs* ➔ *25 hours on a 50 dual node cluster*

# Results


Best
Loyal
Wn
Best
Birds
Best
BT
Performance
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
Robustness


Aggressiveness
Robustness
0
0.2
0.4
0.6
0.8
1

# Salient Observations (1)




- Lower cluster (low P) all free rider variants who do not reciprocate with partners
- Upper cluster (high P) do reciprocate with partners but some defect with strangers
- Top P, low number of partners (1,2), Sort Loyal, When Needed
- Top R, high number of partners (6-9), Sort Fastest, When Needed, Prop. Share
- Sweet spot (P,R>0.8): Sort Loyal

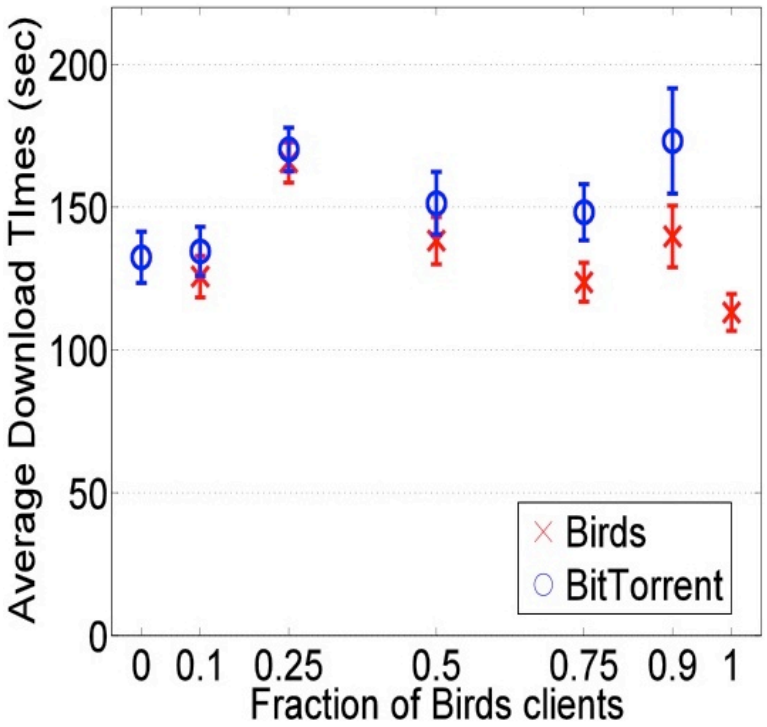
# Salient Observations (2)

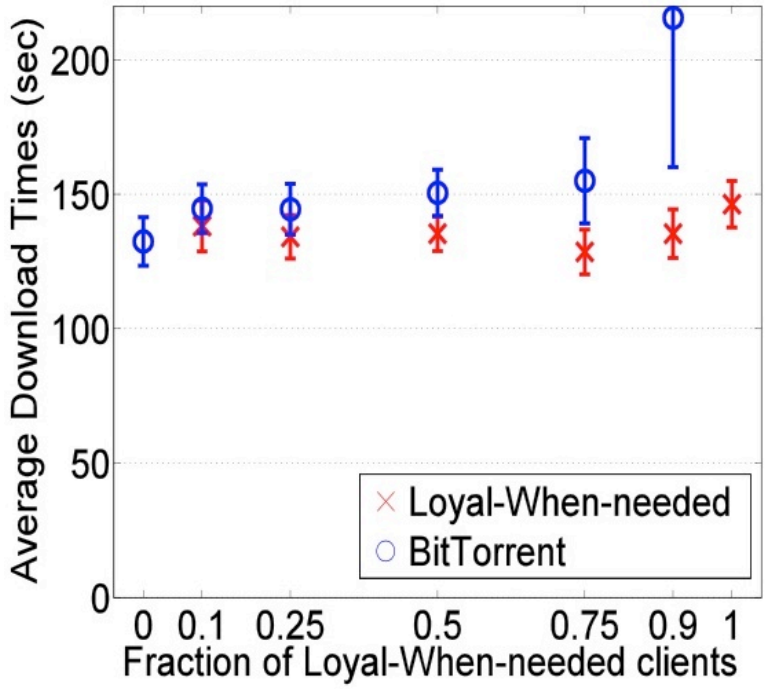
- Highest performing protocols:
  - *Defect on strangers*
  - *Sort Slowest!*
  - *Low number of regular partners (1-2)*

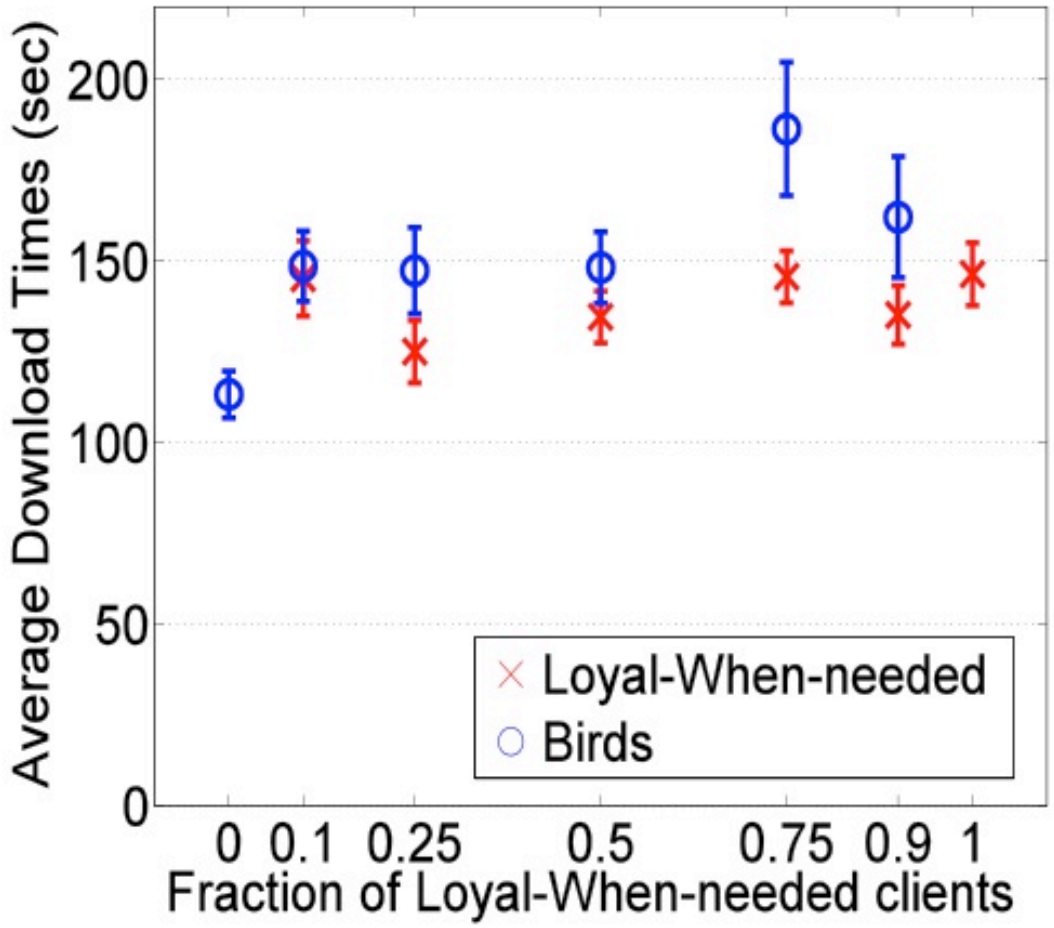
- Highly robust protocols
  - *Use Propshare*
  - *Sort Fastest*
  - *Use When_needed stranger policy*

# Validation of Results with instrumented BitTorrent Clients

Based on client from Legout et al [Sigmetrics2007]




Average Download Times (sec)
200
150
100
50
0
0 0.1 0.25 0.5 0.75 0.9 1
Fraction of Loyal-When-needed clients
Loyal-When-needed
Birds

# Related Work

- Mechanism design [Feigenbaum/Shenker 2002; Dash/Jennings/ Parks 2003]
- Game theory for system design [Majahan/Rodrig/Wetherall/ Zahorjan 2004]
- Evolutionary game theory to p2p [Feldman/Lai/Stoica/Chuang 2004]
- BitTorrent is a Nash [Qiu/Srikant, 2004]
- BitTorrent is an Auction [Levin/LaCurts/Sring/Bhattacharjee, 2008]

# Conclusions

- Standard BT is not a Nash; Birds is a Nash
- Game theoretic models are focused on a single protocol and do not cover all aspects of a protocol
- DSA is a complementary simulation based approach that explores a larger protocol design space
- Future research
  - Other DSA dimensions: Fairness?
  - Other protocols than p2p
  - Heuristics to prune search space

# Thanks for listening!

# Performance of Various Protocols