Modeling and Analysis of Random Walk Search Algorithms in P2P Networks

Nabhendra Bisnik, Alhussein Abouzeid

ECSE, Rensselaer Polytechnic Institute

Contributions

- Analytical expressions for performance metrics of random walk search in decentralized P2P networks
- An algorithm, called equation based adaptive search (***EBAS***), that uses analytical results to set the parameters of random walk
- Feedback based mechanism for maintaining popularity[1] estimates

[1] Popularity of an item equals the fraction of nodes in the network that have the item

Outline

- Search mechanisms and trade-offs involved
- Analytical results
- EBAS
- Simulation results
- Conclusion and future work

- ☑ Search mechanisms and trade-offs involved
- ☐ Analytical results
- ☐ EBAS
- ☐ Simulation Results
- ☐ Conclusion and future Work

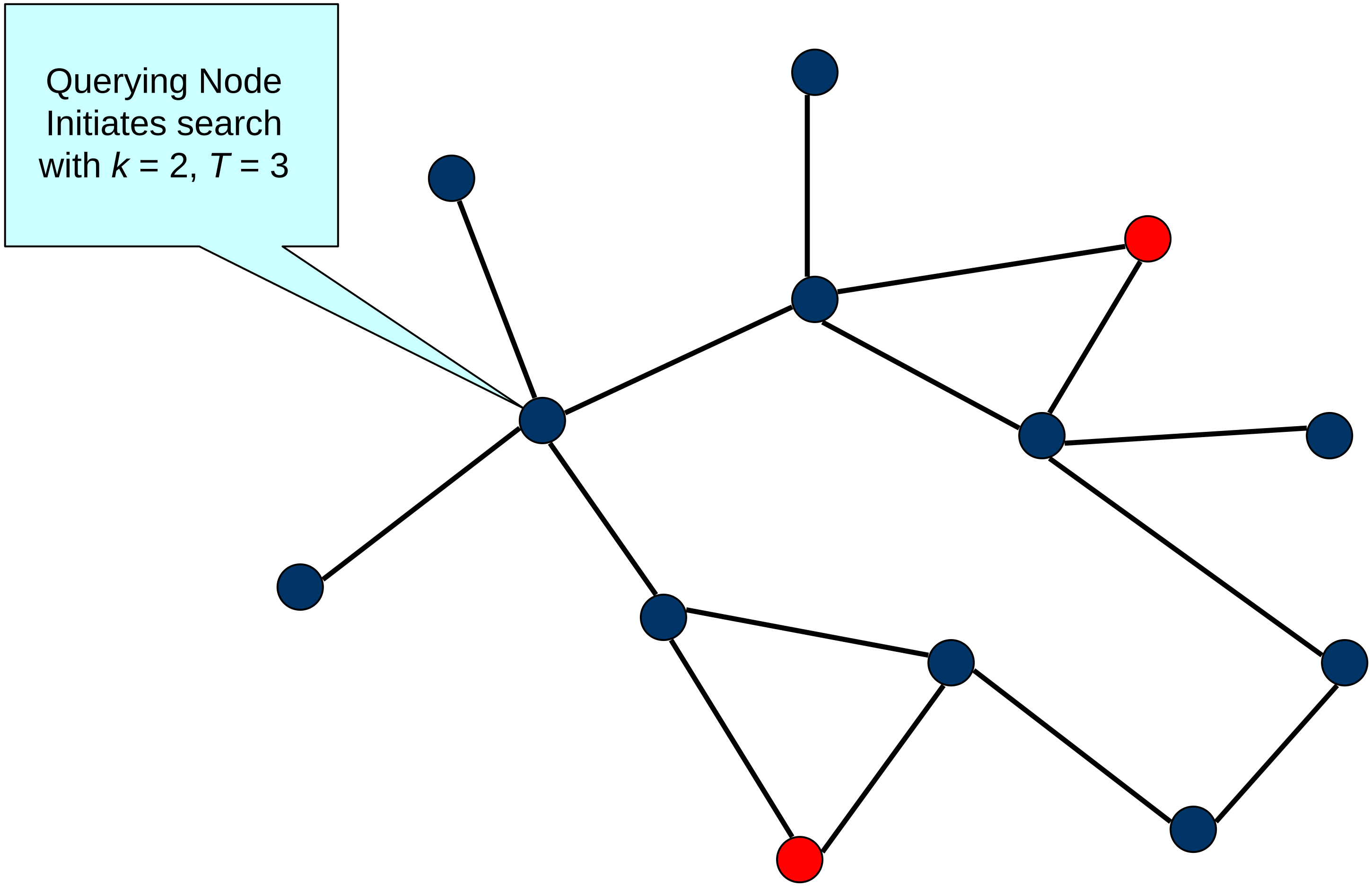
Search Algorithms

- Stateless search algorithms
  - No state information about network or links maintained
  - e.g. Flooding, Iterative Deepening, Random Walk, k-Random Walk
- State-full search algorithms
  - Improvement in performance by maintaining state information (goodness of neighbors, resource indices)
  - Better performance but more complex
  - e.g. Directed BFS, Local Indices, APS

Search Algorithms (cont.)

- Desired performance
  - Low **overhead**
  - Low **delay**
  - High **success rate**
- Trade-offs
  - Overhead and success rate
  - Overhead and delay

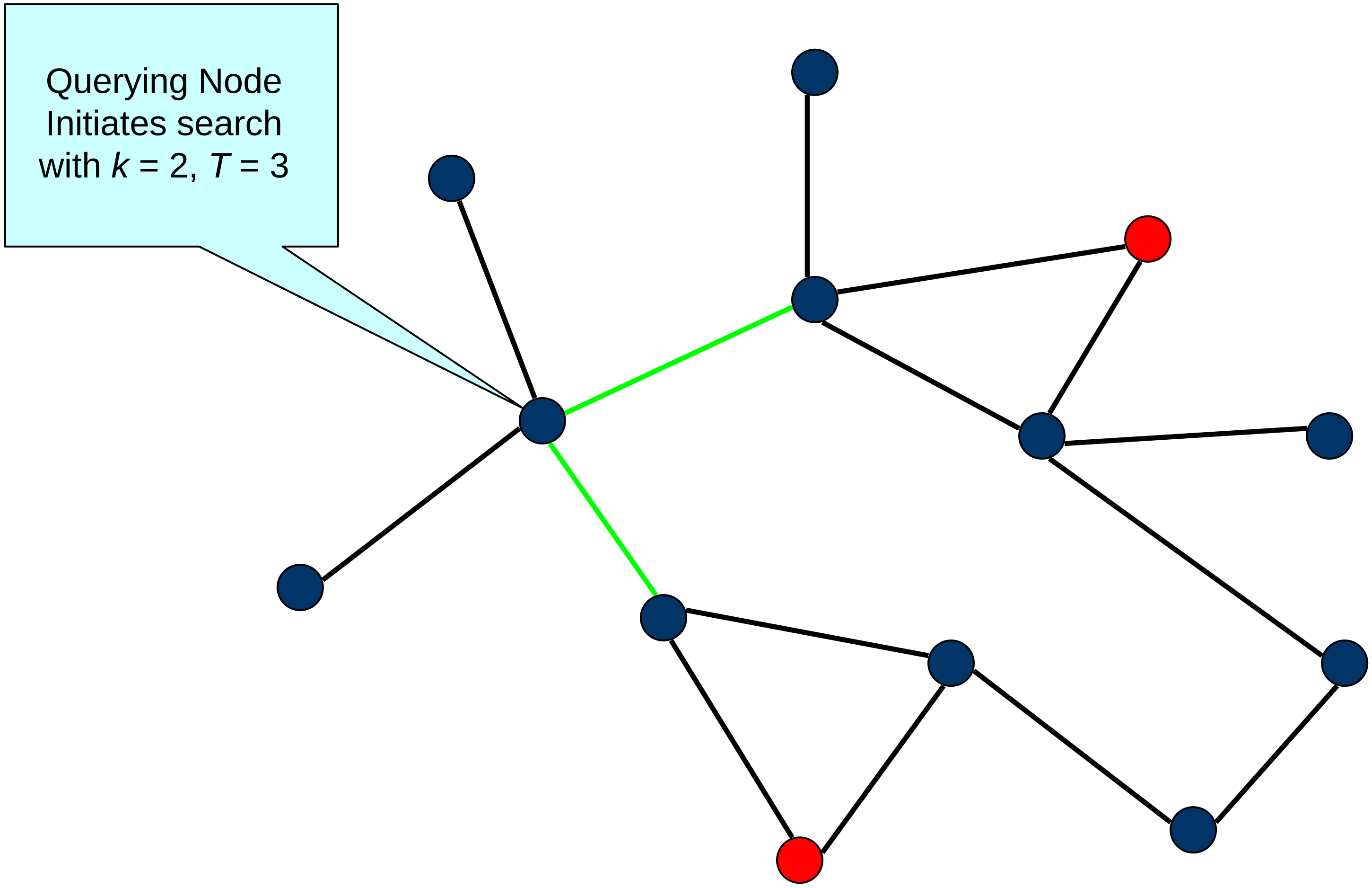
The *k*-Random Walk



Nodes With the Resources

Nodes Without the Resources

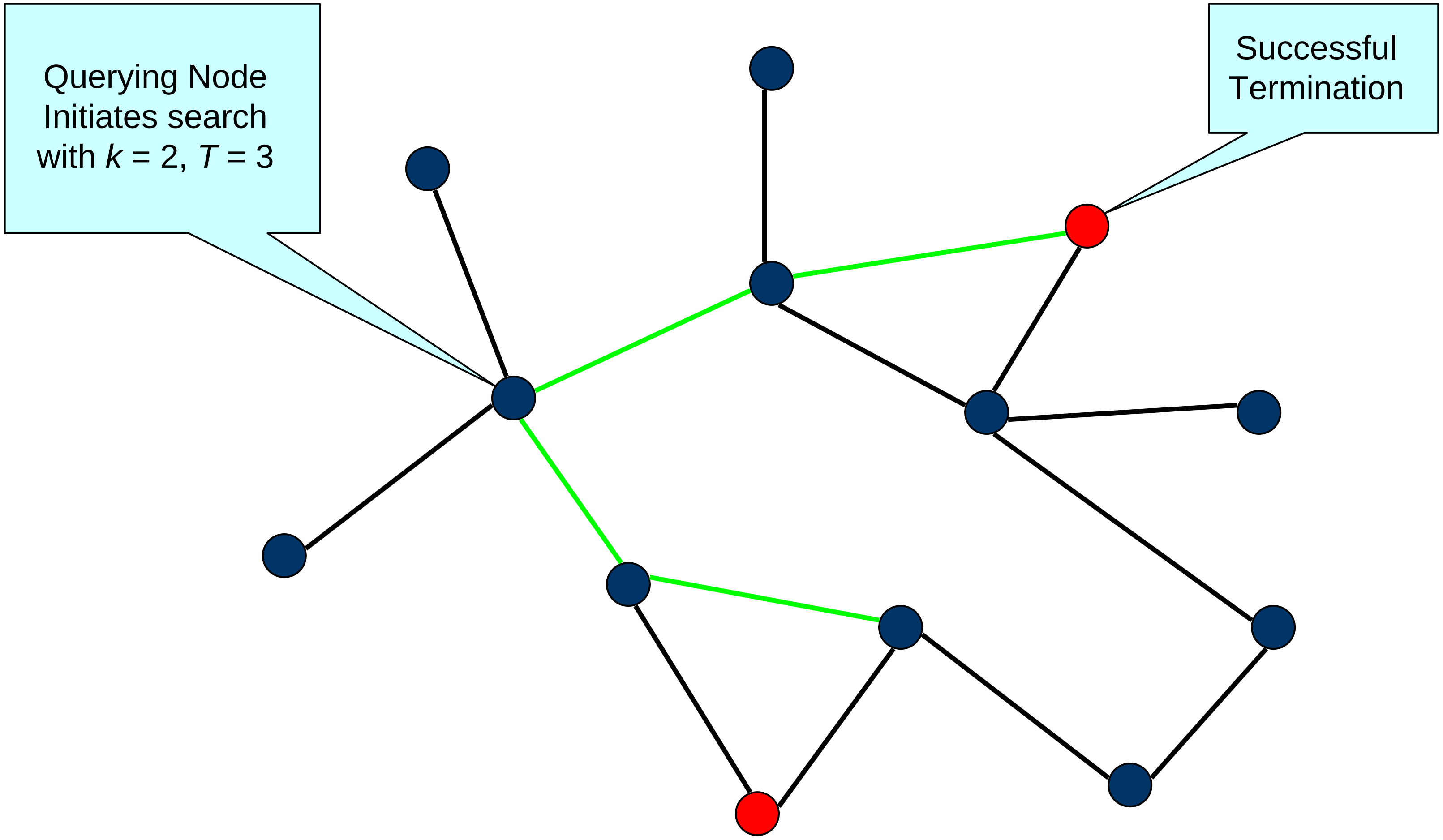
The *k*-Random Walk



Nodes With the Resources

Nodes Without the Resources

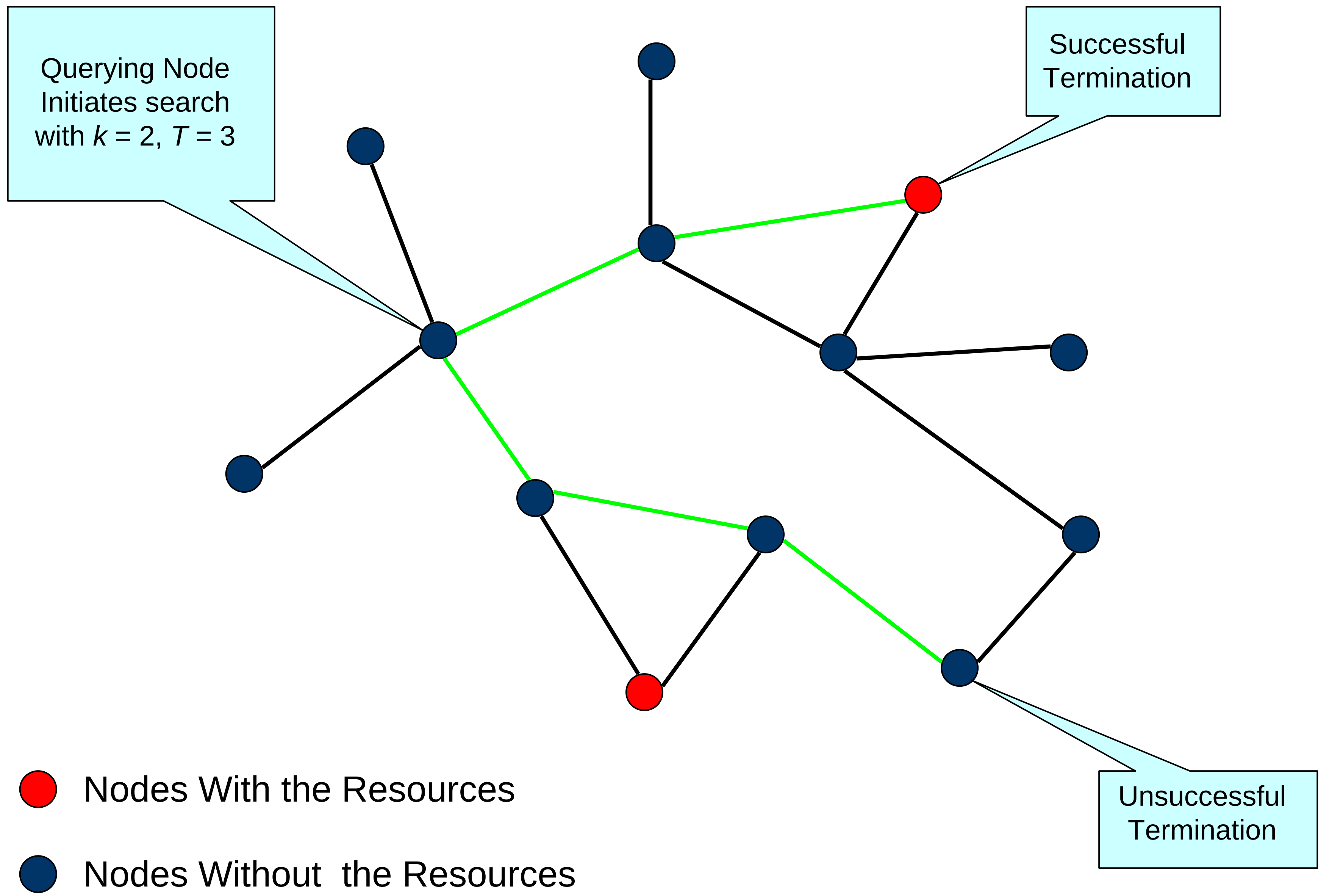
The *k*-Random Walk



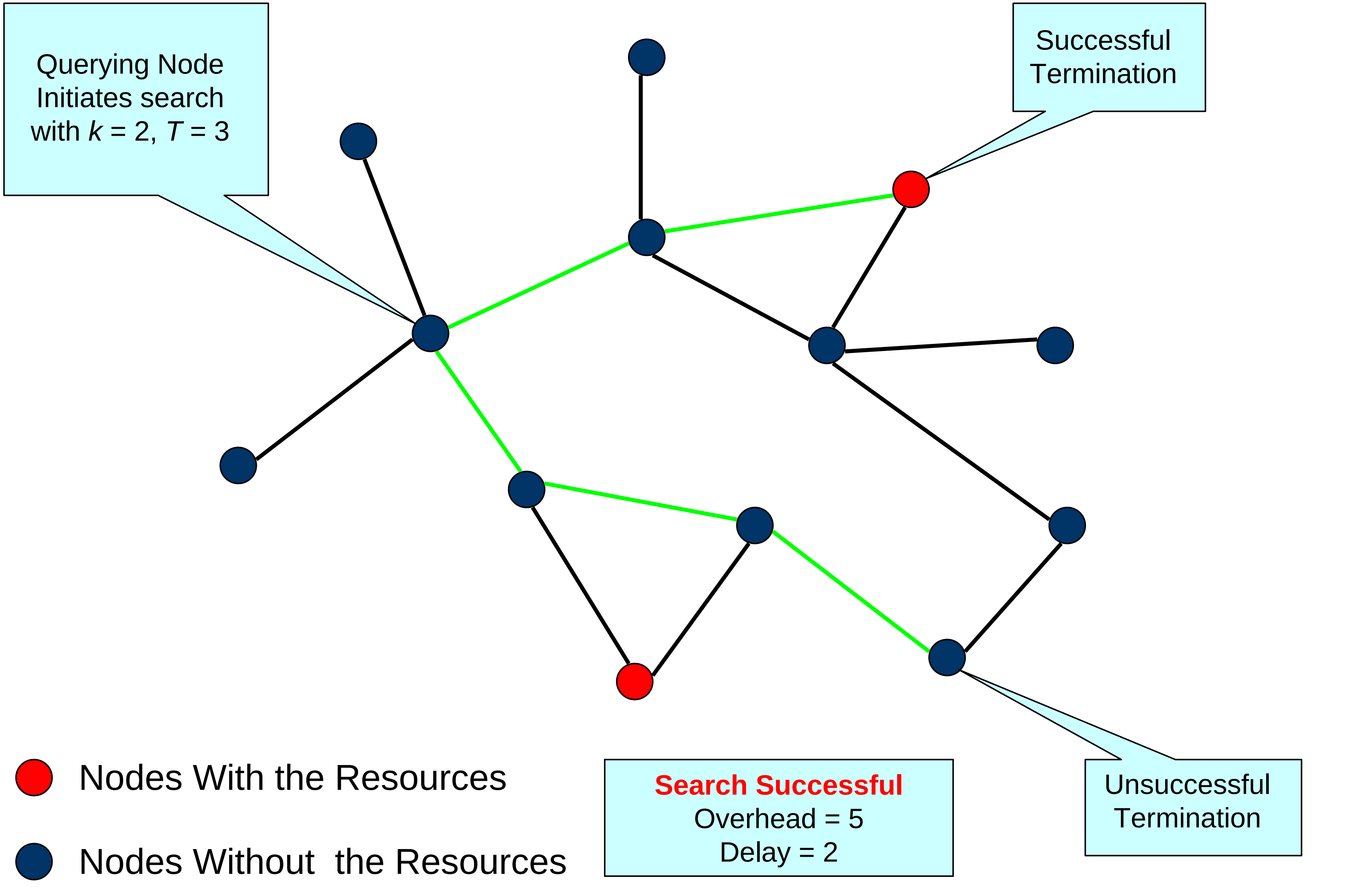
Nodes With the Resources

Nodes Without the Resources

The *k*-Random Walk



The *k*-Random Walk



The *k*-Random Walk (cont.)

- Popular alternative to flooding [3]
- So far focus is on adaptively forwarding queries to “good” neighbors
- Performance depends on parameters *k* and *T* and popularity of resource (*p*)
  - low k and T => high delay and low success rate
  - high k and T => high overhead
- Problem: Number of nodes queried are either more or less than what is required
- Solution: Adaptively set parameters of random walk according to popularity of resource

- ☐ Search mechanisms and trade-offs involved
- ☑ Analytical results
- ☐ EBAS
- ☐ Simulation Results
- ☐ Conclusion and future Work

Analytical results

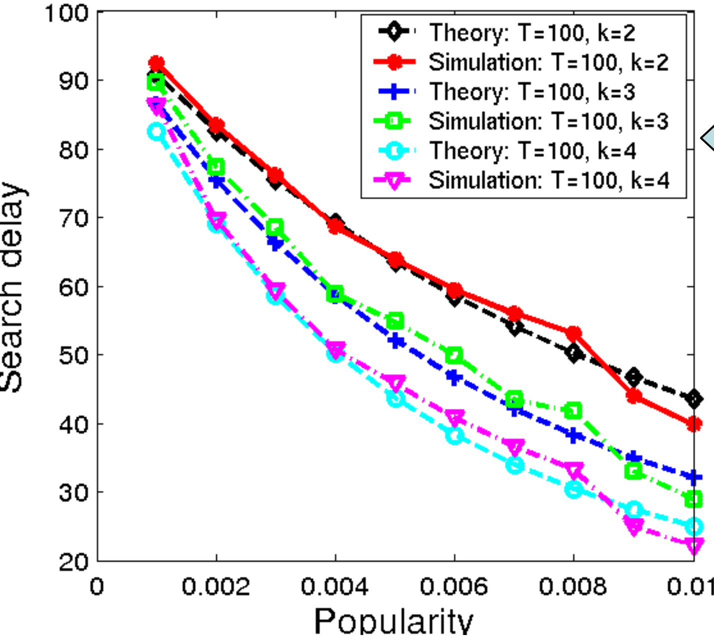
- Random walk has statistical properties similar to sampling from uniform distribution [1]
- Using above we found analytical expressions of success rate, overhead and delay in terms of **number of random walkers** (*k*), **TTL** (*T*) and **popularity** (*p*)

$$p_s = 1-(1-p)^{kT}$$

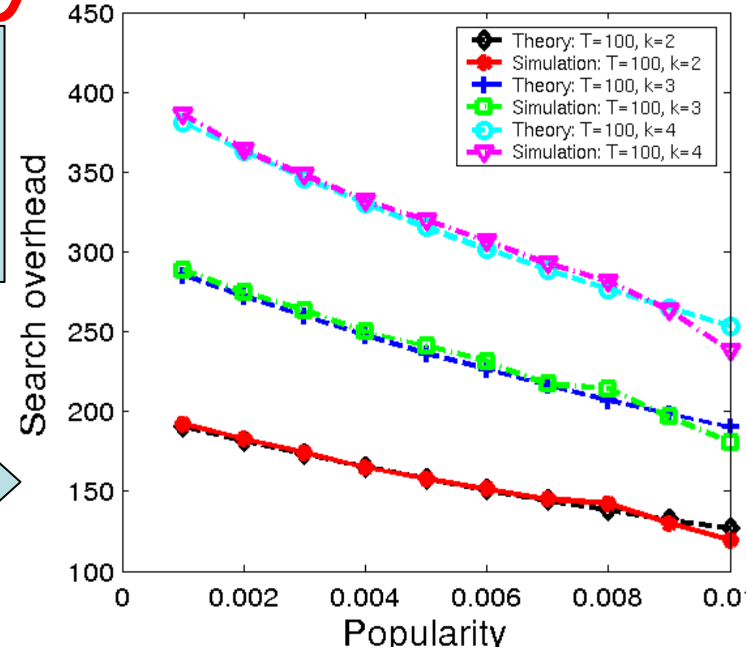
$$E[O] = k \cdot \left(\frac{1-(1-p)^{T-1}}{p} + (1-p)^{T-1} \right)$$

$$E[D] = \frac{1-(1-p)^{k(T-1)}}{1-(1-p)^{k}} + (1-p)^{k(T-1)}$$

Verification of Analytical Results

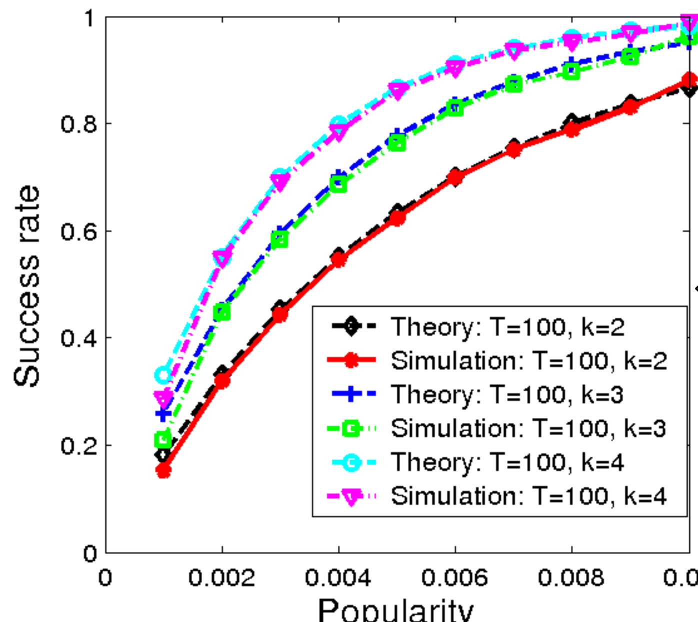


Search delay vs. popularity



Search Overhead vs. popularity

Analytical results agree closely with the simulation results



Success rate vs. popularity

- ❑ Search mechanisms and trade-offs involved
- ❑ Analytical results
- ☑ EBAS
- ❑ Simulation Results
- ❑ Conclusion and future Work

EBAS

- Objective: Set parameters (k,T) of random walk such that

$$p_s \geq 1-\varepsilon(p), \ \ 0<\varepsilon(p)\ll 1 \qquad (1)$$

$$E[O] \leq \alpha(p), \ \ \alpha(p) \geq 1 \qquad (2)$$

$$E[D] \leq \delta(p), \ \ \delta(p) \geq 1 \qquad (3)$$

- EBAS consists of two components:
  - Popularity estimation module
  - Parameter selection module

Popularity Estimator Module

- Popularity estimator is based on exponentially weighted moving average
- Uses the fraction of successful searches in an update interval ($r_i(j)$) to obtain current estimate according to

$$\hat{q}_i(j) = 1 - \exp\left(\frac{\log(1 - r_i(j))}{k_j T_j}\right) \tag{4}$$

- Popularity estimate for next update interval is updated according to

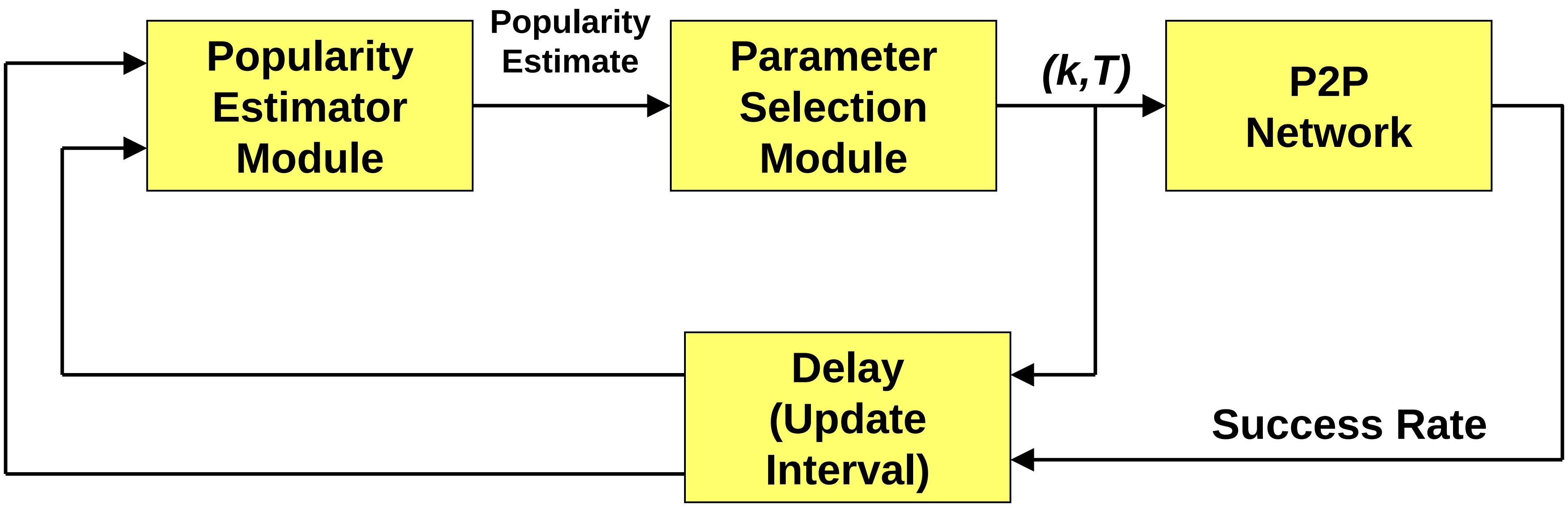
$$\hat{p}_{new} = \beta \cdot \hat{p}_{old} + (1 - \beta) \cdot \hat{q}_i(j), 0 < \beta < 1 \tag{5}$$

Parameter selection module

- Uses $\hat{p}_i(j)$ in order to set k_j and T_j such that (1), (2) and (3) are satisfied
- Inequality (1) is satisfied if

$$k_j \cdot T_j \geq \frac{\log(\varepsilon)}{\log(1-\hat{p}_i(j))} \qquad (6)$$

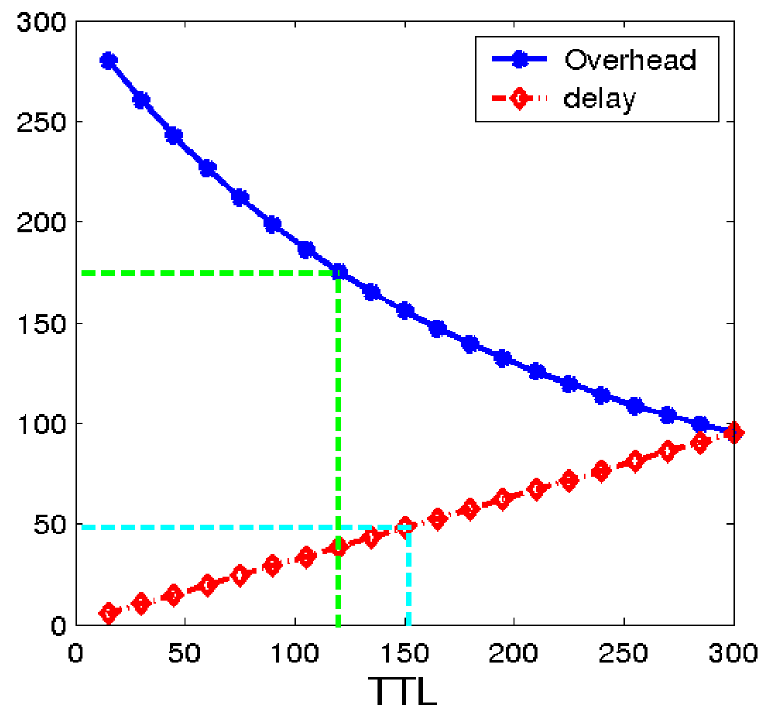
- Inequalities (2) and (3) may be solved numerically in order to obtain a the range of feasible *k* and *T*
- The parameter selection module may be implemented by means of a *parameter selection table* in which feasible values of k and T corresponding to various range of popularities



Block diagram showing main components of EBAS

An Example

- Consider a case where $p = 0.01, \varepsilon(0.01) = 0.05, \alpha(0.01) = 175$ and $\delta(0.01) = 50$
- Inequality (1) is satisfied if $k \cdot T \geq 298$
- *E[O]* and *E[D]* as a function of T are plotted in the figure, for *kT = 300*
- According to the figure $k = 2$ and $T = 150$ is a feasible *(k,T)* pair

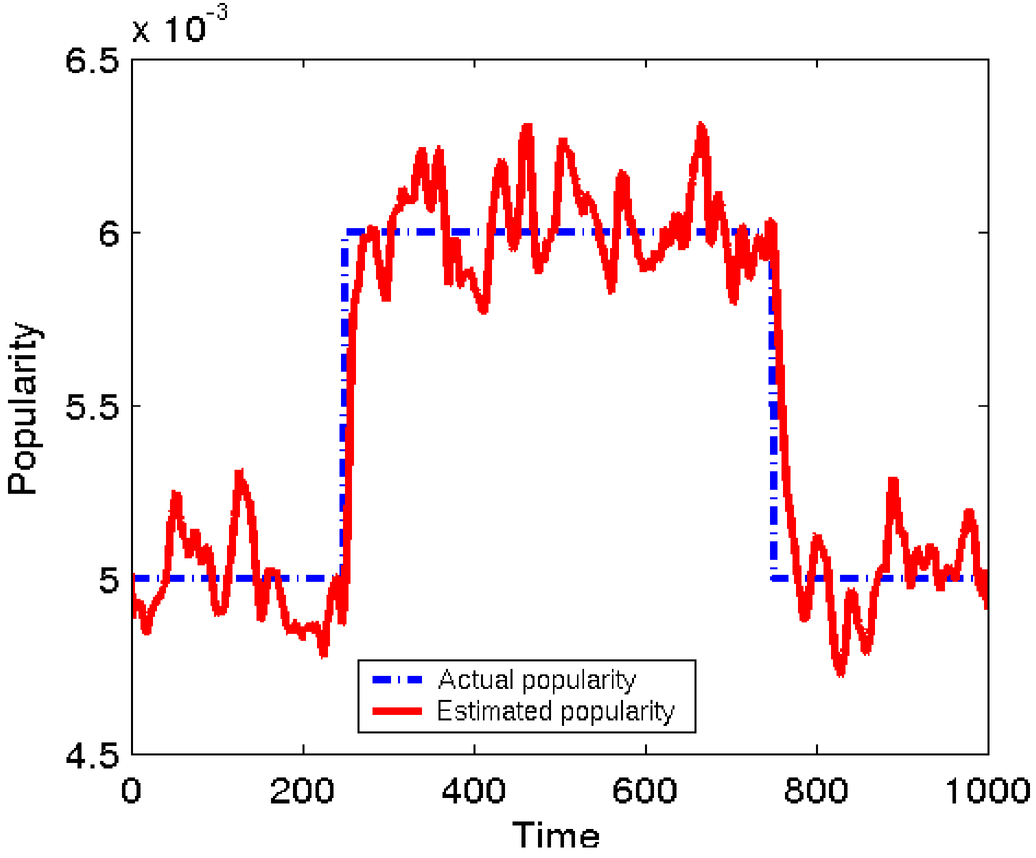


- Search mechanisms and trade-offs involved
- Analytical results
- EBAS
- Simulation Results
- Conclusion and future Work

Simulation Scenario

- Extensive simulations done for
  - Evaluating performance of popularity estimator
  - Comparing performance EBAS with non-adaptive random walk
- Network for simulation
  - 10^4 nodes
  - $px10^4$ nodes randomly chosen and marked to have resource
  - Network grown according to [2]

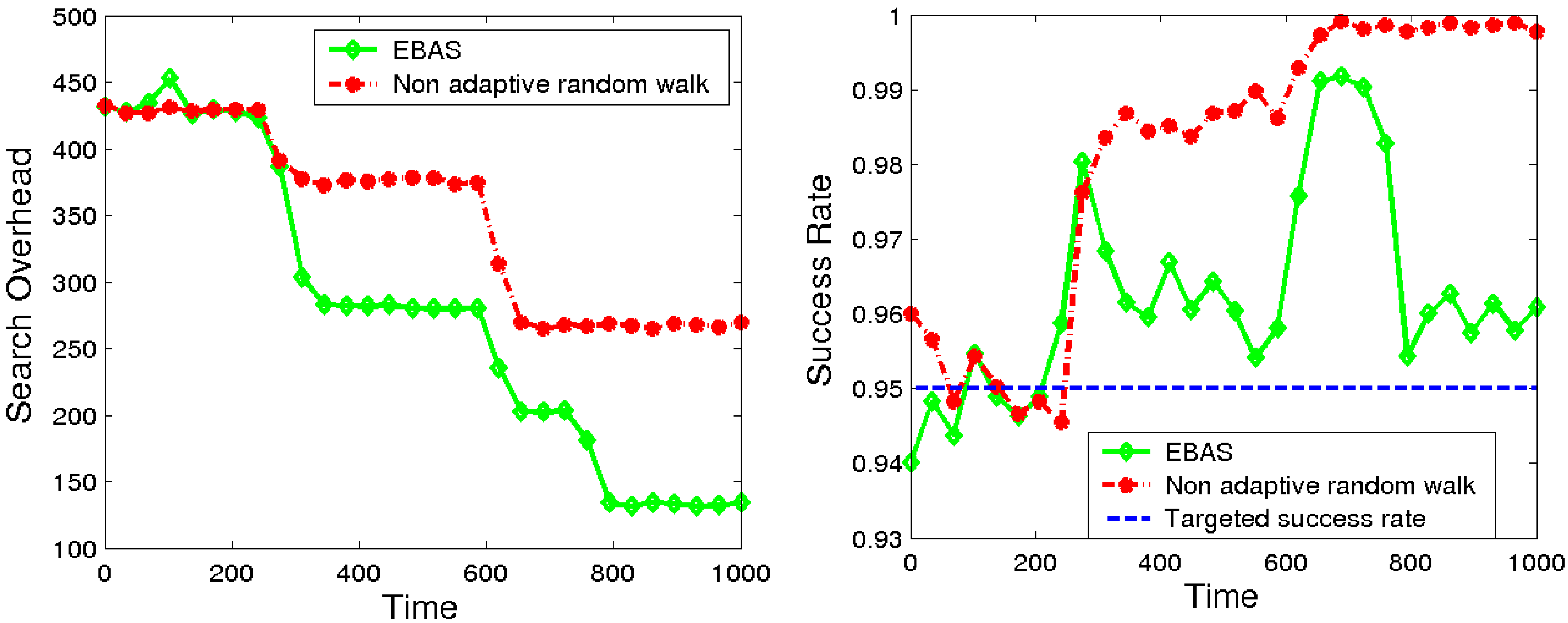
Simulation Results – Popularity Estimator Module



The popularity estimate maintained by Popularity Estimator Module closely follows actual popularity of the resource

Simulation Results - EBAS

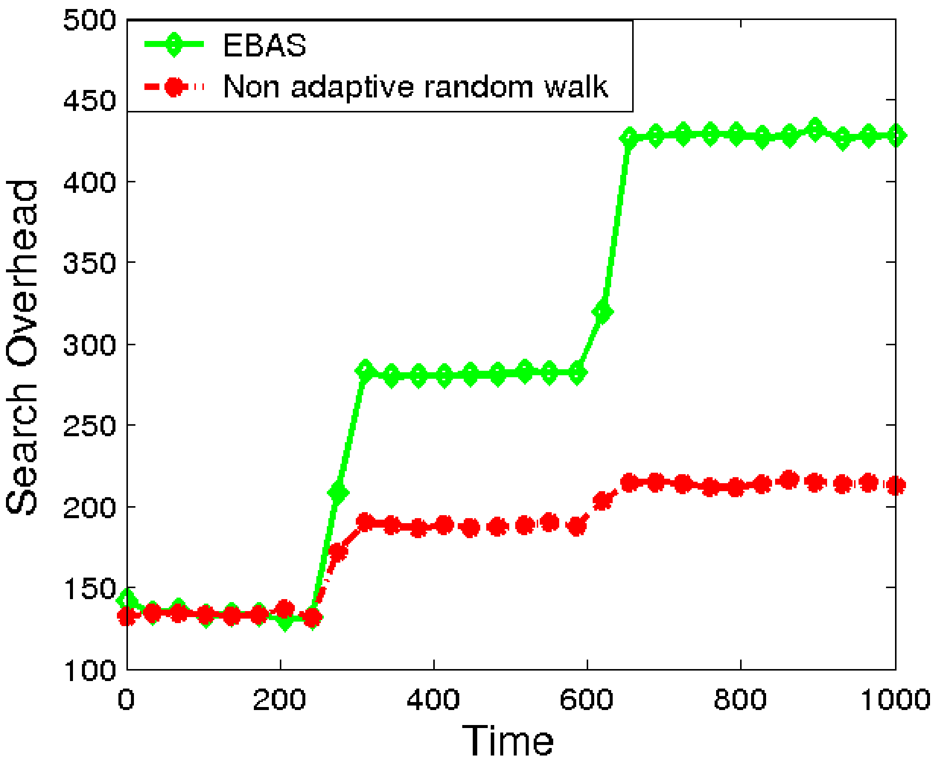
Scenario-1 : Popularity increases with time

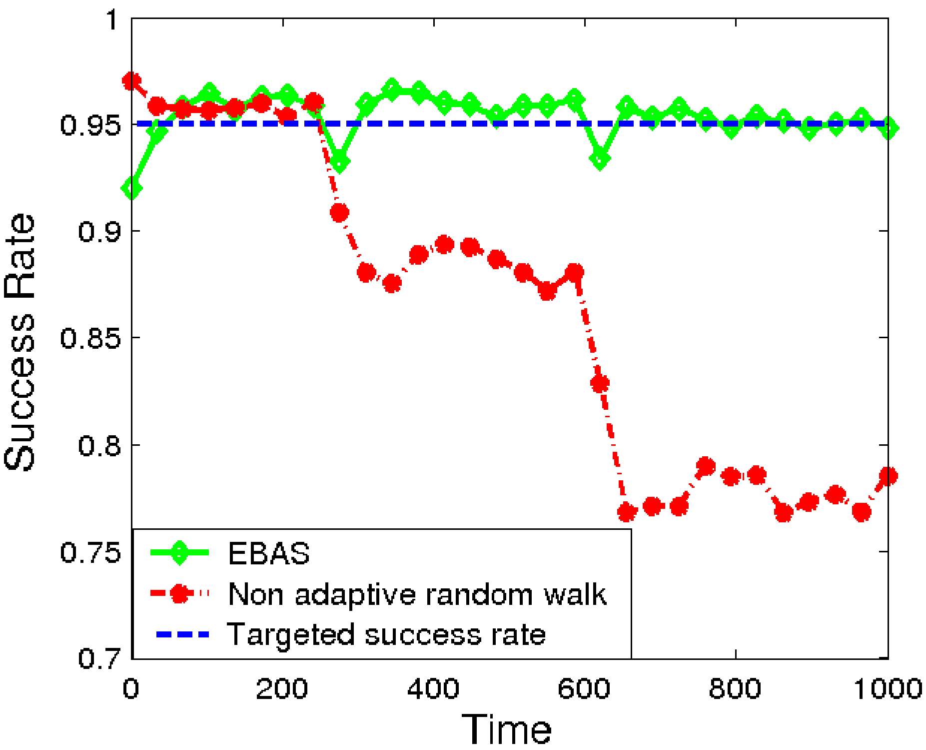


EBAS reduces overhead by adaptively decreasing the number of random walkers

Simulation Results - EBAS

Scenario-2 : Popularity decreases with time





EBAS maintains required success rate by adaptively increasing number of random walkers

- Search mechanisms and trade-offs involved
- Analytical results
- EBAS
- Simulation Results
- Conclusion and future Work

Conclusion & Future Work

- EBAS effectively maintains popularity estimates and performs better than random walk
- EBAS performs best in scenarios where same or “similar’’ item is searched several times
- Modeling problem of choosing optimal parameters as control theoretic problem
- Model performance of other state-full search mechanisms such as APS

References

[1] C. Gkantsidis, M. Mihail, and A. Saberi. Random walks in peer-to-peer networks. In *Proc. of IEEE INFOCOM*, Mar. 2004.

[2] P. Holme and B. J. Kim. Growing Scale-free Networks with Tunable Clustering. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 65, 2002

[3] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. Search and Replication in Unstructured Peer-to-Peer Networks. In *Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 258–259. ACM Press, 2002.

Thank You