What Does it Take to Disconnect a P2P Network?

Dmitri Loguinov

Department of Computer Science
Texas A&M University
College Station, TX 77843

September 29, 2005

Outline

Outline

P2P Basics

Basic Operation

P2P Basics

Basic Operation

- P2P networks can be viewed as random graphs

P2P Basics

Basic Operation

- P2P networks can be viewed as random graphs
- Each joining user v gets k random neighbors from among the existing users (assume a *k-regular* graph for now)

P2P Basics

Basic Operation

- P2P networks can be viewed as random graphs
- Each joining user v gets k random neighbors from among the existing users (assume a *k-regular* graph for now)
- Upon detecting neighbor failure, v searches for random replacement (unstructured P2P) or waits for zone repair (DHTs), in both cases obtaining a new neighbor

P2P Basics

Basic Operation

- P2P networks can be viewed as random graphs
- Each joining user v gets k random neighbors from among the existing users (assume a *k-regular* graph for now)
- Upon detecting neighbor failure, v searches for random replacement (unstructured P2P) or waits for zone repair (DHTs), in both cases obtaining a new neighbor

Resilience Metrics

P2P Basics

Basic Operation

- P2P networks can be viewed as random graphs
- Each joining user v gets k random neighbors from among the existing users (assume a *k-regular* graph for now)
- Upon detecting neighbor failure, v searches for random replacement (unstructured P2P) or waits for zone repair (DHTs), in both cases obtaining a new neighbor

Resilience Metrics

- *Local*: node isolation

P2P Basics

Basic Operation

- P2P networks can be viewed as random graphs
- Each joining user v gets k random neighbors from among the existing users (assume a *k-regular* graph for now)
- Upon detecting neighbor failure, v searches for random replacement (unstructured P2P) or waits for zone repair (DHTs), in both cases obtaining a new neighbor

Resilience Metrics

- *Local*: node isolation
- *Global*: disconnection of the graph (its undirected version)

Outline

Motivation I

Common Node Failure

Motivation I

Common Node Failure

- Traditional P2P resilience centers around uniform, independent, simultaneous node failure (*static* resilience)

Motivation I

Common Node Failure

- Traditional P2P resilience centers around uniform, independent, simultaneous node failure (*static* resilience)
- Each node is considered dead with probability p

Motivation I

Common Node Failure

- Traditional P2P resilience centers around uniform, independent, simultaneous node failure (*static* resilience)
- Each node is considered dead with probability p
- Value $p = 0.5$ is frequently used (Chord, Koorde, etc.)

Motivation I

Common Node Failure

- Traditional P2P resilience centers around uniform, independent, simultaneous node failure (*static* resilience)
- Each node is considered dead with probability p
- Value $p = 0.5$ is frequently used (Chord, Koorde, etc.)

P2P Churn

Motivation I

Common Node Failure

- Traditional P2P resilience centers around uniform, independent, simultaneous node failure (*static* resilience)
- Each node is considered dead with probability p
- Value $p = 0.5$ is frequently used (Chord, Koorde, etc.)

P2P Churn

- Static failure model is rarely applicable to real P2P networks (Gnutella, KaZaA, etc.)

Motivation I

Common Node Failure

- Traditional P2P resilience centers around uniform, independent, simultaneous node failure (*static* resilience)
- Each node is considered dead with probability p
- Value $p = 0.5$ is frequently used (Chord, Koorde, etc.)

P2P Churn

- Static failure model is rarely applicable to real P2P networks (Gnutella, KaZaA, etc.)
- Nodes depart asynchronously based on user browsing habits or interest (*dynamic* resilience)

Motivation II

Common Static Resilience Model

Most resilience results are local: $P(\text{node } v \text{ is isolated}) = p^k$, where k is the degree of v

Motivation II

Common Static Resilience Model

Most resilience results are local: $P(\text{node } v \text{ is isolated}) = p^k$, where k is the degree of v

Static Example

- Assume $n = 10^{11}$ users and desired probability that each node remains connected equal to $1 - 1/n$

Motivation II

Common Static Resilience Model

Most resilience results are local: $P(\text{node } v \text{ is isolated}) = p^k$, where k is the degree of v

Static Example

- Assume $n = 10^{11}$ users and desired probability that each node remains connected equal to $1 - 1/n$
- Then k must be at least 37

Motivation II

Common Static Resilience Model

Most resilience results are local: $P(\text{node } v \text{ is isolated}) = p^k$, where k is the degree of v

Static Example

- Assume $n = 10^{11}$ users and desired probability that each node remains connected equal to $1 - 1/n$
- Then k must be at least 37
- Can we do better?

Outline

Classical Results I

Disconnection of Random Graphs

Classical Results I

Disconnection of Random Graphs

- *Almost every* (i.e., with probability $1 - o(1)$ as $n \to \infty$) random graph $G(n, p)$, $G(n, M)$, $G(n, k_{out})$ remains connected after static failure if and only if it has no isolated vertices

Classical Results I

Disconnection of Random Graphs

- *Almost every* (i.e., with probability $1 - o(1)$ as $n \to \infty$) random graph $G(n, p)$, $G(n, M)$, $G(n, k_{out})$ remains connected after static failure if and only if it has no isolated vertices
- *If each user manages to avoid isolation, the graph almost surely remains connected after the failure!*

Classical Results I

Disconnection of Random Graphs

- *Almost every* (i.e., with probability $1 - o(1)$ as $n \to \infty$) random graph $G(n,p)$, $G(n,M)$, $G(n,k_{out})$ remains connected after static failure if and only if it has no isolated vertices
- *If each user manages to avoid isolation, the graph almost surely remains connected after the failure!*

Intuition

Classical Results I

Disconnection of Random Graphs

- *Almost every* (i.e., with probability $1 - o(1)$ as $n \to \infty$) random graph $G(n, p)$, $G(n, M)$, $G(n, k_{out})$ remains connected after static failure if and only if it has no isolated vertices
- *If each user manages to avoid isolation, the graph almost surely remains connected after the failure!*

Intuition

Conditional probability of partitioning along a set boundary *while* not developing isolated nodes tends to zero

Classical Results II

Deterministic Networks

Classical Results II

Deterministic Networks

- Burtin (1977) and Bollobás (1983) showed the same result for certain deterministic graphs such as hypercubes

Classical Results II

Deterministic Networks

- Burtin (1977) and Bollobás (1983) showed the same result for certain deterministic graphs such as hypercubes
- This can be extended to any graph with similar or better node expansion properties (Chord, CAN, Pastry, etc.)

Classical Results II

Deterministic Networks

- Burtin (1977) and Bollobás (1983) showed the same result for certain deterministic graphs such as hypercubes
- This can be extended to any graph with similar or better node expansion properties (Chord, CAN, Pastry, etc.)

Table: Chord with $n = 16384$ under p-percent failure

p	$P(G$ is connected)	P(no isolated nodes)
0.5	0.99996	0.99996
0.6	0.99354	0.99354
0.7	0.72619	0.72650
0.8	0.00040	0.00043

Classical Result III

Application to P2P graphs

All tested P2P graphs (Chord, Symphony, CAN, Pastry, Randomized Chord, de Bruijn, and several unstructured random graphs) remained connected almost surely as long as they did not have an isolated node

Classical Result III

Application to P2P graphs

All tested P2P graphs (Chord, Symphony, CAN, Pastry, Randomized Chord, de Bruijn, and several unstructured random graphs) remained connected almost surely as long as they did not have an isolated node

Milestone

Local resilience of P2P networks *under static failure* implies their global resilience

Outline

Lifetime-Based Extension I

Definition

Let ϕ be the probability that of a joining user becomes isolated *during its lifetime* due to neighbor failure

Lifetime-Based Extension I

Definition

Let ϕ be the probability that of a joining user becomes isolated *during its lifetime* due to neighbor failure

Problem

Compute $P(G \text{ survives } N \text{ user joins without disconnecting})$

Lifetime-Based Extension I

Definition

Let ϕ be the probability that of a joining user becomes isolated *during its lifetime* due to neighbor failure

Problem

Compute $P(G$ survives N user joins without disconnecting$)$

Definition

Let Y be the number of user joins before the first disconnection of the network

Lifetime-Based Extension II

Asymptotic Independence

Lifetime-Based Extension II

Asymptotic Independence

Dependency between user isolation events diminishes to zero as $n \to \infty$, in which case the following result holds

Lifetime-Based Extension II

Asymptotic Independence

Dependency between user isolation events diminishes to zero as $n \to \infty$, in which case the following result holds

Simple Model

Lifetime-Based Extension II

Asymptotic Independence

Dependency between user isolation events diminishes to zero as $n \to \infty$, in which case the following result holds

Simple Model

- For almost every sufficiently large graph:

$$P(Y > N) \approx (1 - \phi)^N$$

Lifetime-Based Extension II

Asymptotic Independence

Dependency between user isolation events diminishes to zero as $n \to \infty$, in which case the following result holds

Simple Model

- For almost every sufficiently large graph:

$$P(Y > N) \approx (1 - \phi)^N$$

- *Knowledge of ϕ is all we need to understand dynamic resilience of P2P systems!*

Lifetime-Based Extension III

Example

- CAN with exponential lifetimes (mean 30 minutes)

Lifetime-Based Extension III

Example

- CAN with exponential lifetimes (mean 30 minutes)
- Degree $k = 12$ ($n = 4096$ nodes)

Lifetime-Based Extension III

Example

- CAN with exponential lifetimes (mean 30 minutes)
- Degree $k = 12$ ($n = 4096$ nodes)
- Metric ϕ measured empirically

Lifetime-Based Extension III

Example

- CAN with exponential lifetimes (mean 30 minutes)
- Degree $k = 12$ ($n = 4096$ nodes)
- Metric ϕ measured empirically

Search time (min)	Actual $P(Y > N)$	Model
6	0.9732	0.9728
7.5	0.8118	0.8124
8.5	0.5669	0.5659
9	0.4065	0.4028
9.5	0.2613	0.2645
10.5	0.0482	0.0471

Table: Comparison of model to simulations for $N = 10^6$ user joins

Lifetime-Based Extension IV

Milestone

Local resilience of P2P networks *under dynamic failure* implies their global resilience

Outline

Overview of Lifetime Model I

Model Assumptions

Overview of Lifetime Model I

Model Assumptions

- *Arrival*: nodes arrive independently according to any process as long as system size remains non-trivial

Overview of Lifetime Model I

Model Assumptions

- *Arrival*: nodes arrive independently according to any process as long as system size remains non-trivial
- *Departure*: nodes deterministically die (fail) after spending L_i time units in the system

Overview of Lifetime Model I

Model Assumptions

- *Arrival*: nodes arrive independently according to any process as long as system size remains non-trivial
- *Departure*: nodes deterministically die (fail) after spending L_i time units in the system
- *Neighbor selection*: neighbors are picked from among the existing nodes using any rules that do not involve node lifetimes or age (e.g., based on random walks, DHT space assignment, topological locality, content interests, etc.)

Overview of Lifetime Model I

Model Assumptions

- *Arrival*: nodes arrive independently according to any process as long as system size remains non-trivial
- *Departure*: nodes deterministically die (fail) after spending L_i time units in the system
- *Neighbor selection*: neighbors are picked from among the existing nodes using any rules that do not involve node lifetimes or age (e.g., based on random walks, DHT space assignment, topological locality, content interests, etc.)
- *Neighbor replacement*: once a failed neighbor is detected, a replacement search is performed

Overview of Lifetime Model II

Definition

A node becomes *isolated* when all of the neighbors in its table are simultaneously in the failed state

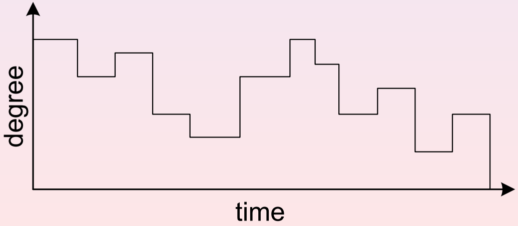
Overview of Lifetime Model II

Definition

A node becomes *isolated* when all of the neighbors in its table are simultaneously in the failed state

Degree Evolution

User degree $W(t)$ is a random process shown below



Overview of Lifetime Model III

Node Arrival

Overview of Lifetime Model III

Node Arrival

- Node v enters at time t_v with random lifetime L_v

Overview of Lifetime Model III

Node Arrival

- Node v enters at time t_v with random lifetime L_v
- It then selects k random neighbors from the system

Overview of Lifetime Model III

Node Arrival

- Node v enters at time t_v with random lifetime L_v
- It then selects k random neighbors from the system

Definition

Let R_i be the remaining (i.e., *residual*) lifetime of neighbor i when v joined the system

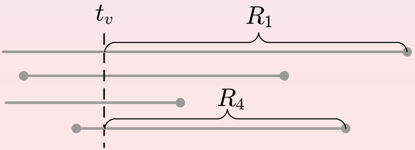
Overview of Lifetime Model III

Node Arrival

- Node v enters at time t_v with random lifetime L_v
- It then selects k random neighbors from the system

Definition

Let R_i be the remaining (i.e., *residual*) lifetime of neighbor i when v joined the system



Overview of Lifetime Model IV

Replacement Algorithms

- Any mechanism for detecting dead neighbors is suitable (e.g., periodic probing, timeouts, retransmission, etc.)

Overview of Lifetime Model IV

Replacement Algorithms

- Any mechanism for detecting dead neighbors is suitable (e.g., periodic probing, timeouts, retransmission, etc.)
- Replacement decisions cannot be made based on lifetimes of potential neighbors

Overview of Lifetime Model IV

Replacement Algorithms

- Any mechanism for detecting dead neighbors is suitable (e.g., periodic probing, timeouts, retransmission, etc.)
- Replacement decisions cannot be made based on lifetimes of potential neighbors
- Otherwise, replacement process may be arbitrary

Overview of Lifetime Model IV

Replacement Algorithms

- Any mechanism for detecting dead neighbors is suitable (e.g., periodic probing, timeouts, retransmission, etc.)
- Replacement decisions cannot be made based on lifetimes of potential neighbors
- Otherwise, replacement process may be arbitrary

Definition

Let S_i be a random variable describing the total search time for the i-th replacement in the system

Overview of Lifetime Model VI

Pertinent Questions

Overview of Lifetime Model VI

Pertinent Questions

- What is the average amount of time a node will spend in the system before becoming isolated?

Overview of Lifetime Model VI

Pertinent Questions

- What is the average amount of time a node will spend in the system before becoming isolated?
- What is the probability that a node will become isolated from the network within its lifetime? (metric ϕ)

Overview of Lifetime Model VI

Pertinent Questions

- What is the average amount of time a node will spend in the system before becoming isolated?
- What is the probability that a node will become isolated from the network within its lifetime? (metric ϕ)
- How does varying node degree between users improve/degrade resilience?

Overview of Lifetime Model VI

Pertinent Questions

- What is the average amount of time a node will spend in the system before becoming isolated?
- What is the probability that a node will become isolated from the network within its lifetime? (metric ϕ)
- How does varying node degree between users improve/degrade resilience?
- How to increase resilience of existing systems?

Outline

Expected Time to Isolation I

Definition

Let T be the random time instance when v becomes isolated (i.e., $T = \inf\{t > 0 : W(t) = 0\}$ is the first hitting time of $W(t)$ on level 0)

Expected Time to Isolation I

Definition

Let T be the random time instance when v becomes isolated (i.e., $T = \inf\{t > 0 : W(t) = 0\}$ is the first hitting time of $W(t)$ on level 0)

Theorem

Assuming relatively small search delays, the following approximation holds for all lifetime and search distributions:

$$E[T] \approx \frac{E[S_i]}{k}\left[\left(1 + \frac{E[R_i]}{E[S_i]}\right)^k - 1\right]$$

Expected Time to Isolation II

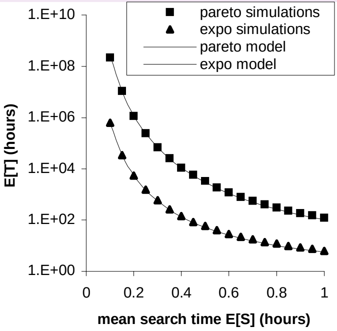
Simulations

Simulations with average lifetime 30 minutes and $k = 10$ for a 1000 node system (four distributions of S_i)

Expected Time to Isolation II

Simulations

Simulations with average lifetime 30 minutes and $k = 10$ for a 1000 node system (four distributions of S_i)

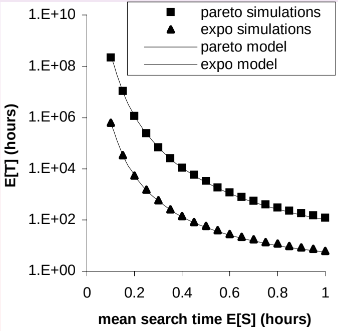


(a) uniform S_i

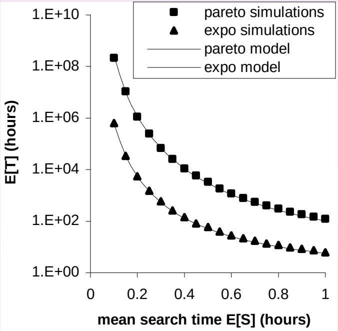
Expected Time to Isolation II

Simulations

Simulations with average lifetime 30 minutes and $k = 10$ for a 1000 node system (four distributions of S_i)

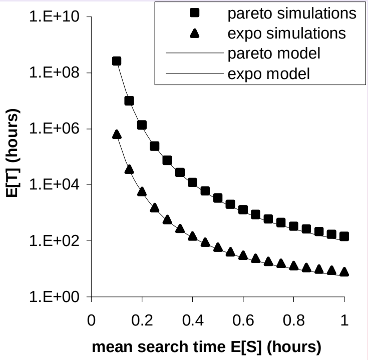


(e) uniform S



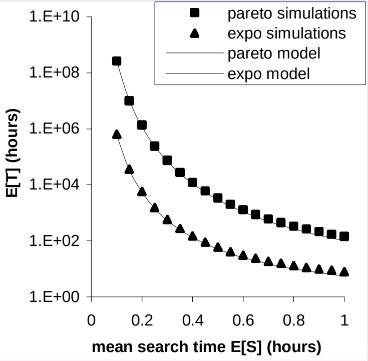
(f) binomial S

Expected Time to Isolation III

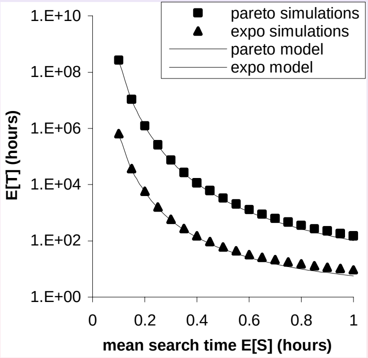


(g) exponential S_i

Expected Time to Isolation III



(i) exponential S_i



(j) Pareto S_i with $\alpha = 3$

Expected Time to Isolation IV

Definition

Let δ be the keep-alive timeout and d be the average inter-peer delay in the overlay

Expected Time to Isolation IV

Definition

Let δ be the keep-alive timeout and d be the average inter-peer delay in the overlay

Result for Chord

We immediately obtain from the main model:

$$E[T] \approx \frac{\delta + d\log_2 n}{2k}\left(1 + \frac{2E[R_i]}{\delta + d\log_2 n}\right)^k$$

Expected Time to Isolation V

Example

Expected Time to Isolation V

Example

Chord with $n = 1$ million, $d = 200$ ms, $E[R_i] = 1$ hour (Pareto lifetimes with $E[L_i] = 30$ minutes)

Expected Time to Isolation V

Example

Chord with $n = 1$ million, $d = 200$ ms, $E[R_i] = 1$ hour (Pareto lifetimes with $E[L_i] = 30$ minutes)

Timeout δ	$k = 20$	$k = 10$	$k = 5$
20 sec	10^{41} years	10^{17} years	$188,034$ years
2 min	10^{28} years	10^{11} years	282 years
45 min	$404,779$ years	680 days	49 hours

Table: Expected time $E[T]$ to isolation

Outline

Probability of Isolation I

General Idea

- Recall that ϕ is the probability that a node v becomes isolated during its lifetime

Probability of Isolation I

General Idea

- Recall that ϕ is the probability that a node v becomes isolated during its lifetime
- Notice that $\phi = P(T < L_v)$, where both T and L_v are random variables

Probability of Isolation I

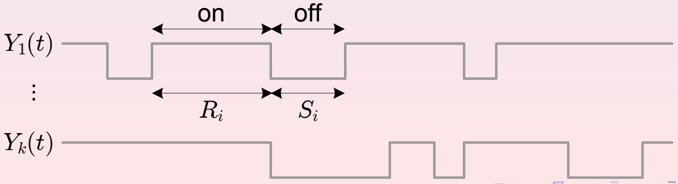
General Idea

- Recall that ϕ is the probability that a node v becomes isolated during its lifetime
- Notice that $\phi = P(T < L_v)$, where both T and L_v are random variables
- We model the neighbor failure/replacement procedure as an on/off process $Y_i(t)$

Probability of Isolation I

General Idea

- Recall that ϕ is the probability that a node v becomes isolated during its lifetime
- Notice that $\phi = P(T < L_v)$, where both T and L_v are random variables
- We model the neighbor failure/replacement procedure as an on/off process $Y_i(t)$



Probability of Isolation II

Degree Evolution

Probability of Isolation II

Degree Evolution

- Then the degree of node v at time t is $W(t) = \sum_{i=1}^{k} Y_i(t)$, which is a classical birth-death process

Probability of Isolation II

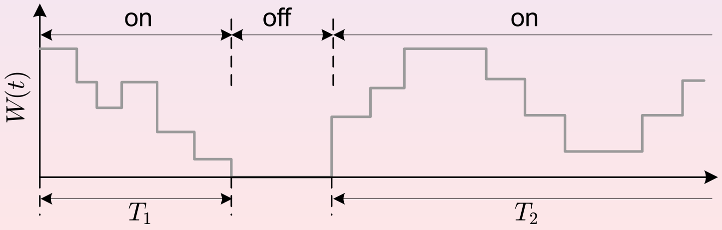
Degree Evolution

- Then the degree of node v at time t is $W(t) = \sum_{i=1}^{k} Y_i(t)$, which is a classical birth-death process
- Convenient to view as an alternating on/off process

Probability of Isolation II

Degree Evolution

- Then the degree of node v at time t is $W(t) = \sum_{i=1}^{k} Y_i(t)$, which is a classical birth-death process
- Convenient to view as an alternating on/off process



Probability of Isolation III

Theorem

For exponential lifetimes and $E[S_i] \ll E[L_i]$*, the probability of isolation* ϕ *can be approximated by:*

$$\phi \approx \frac{E[L_i]}{E[T]} = \frac{\rho k}{(1+\rho)^k + \rho k - 1}$$

where $\rho = E[L_i]/E[S_i]$

Probability of Isolation III

Theorem

For exponential lifetimes and $E[S_i] \ll E[L_i]$*, the probability of isolation* ϕ *can be approximated by:*

$$\phi \approx \frac{E[L_i]}{E[T]} = \frac{\rho k}{(1+\rho)^k + \rho k - 1}$$

where $\rho = E[L_i]/E[S_i]$

Verification

Probability of Isolation III

Theorem

For exponential lifetimes and $E[S_i] \ll E[L_i]$*, the probability of isolation* ϕ *can be approximated by:*

$$\phi \approx \frac{E[L_i]}{E[T]} = \frac{\rho k}{(1+\rho)^k + \rho k - 1}$$

where $\rho = E[L_i]/E[S_i]$

Verification

Simulations match the model very well and, for small S_i, the results are not sensitive to the distribution of search delay

Probability of Isolation IV

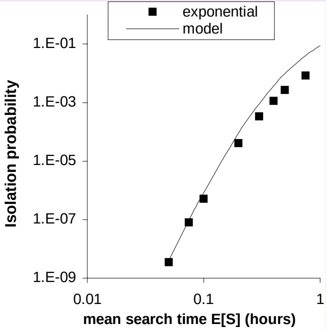
Simulations

System with $E[L_i] = 0.5$ hours and $k = 10$

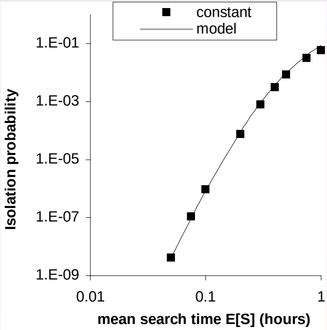
Probability of Isolation IV

Simulations

System with $E[L_i] = 0.5$ hours and $k = 10$

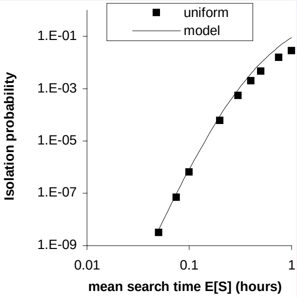


(m) exponential S_i

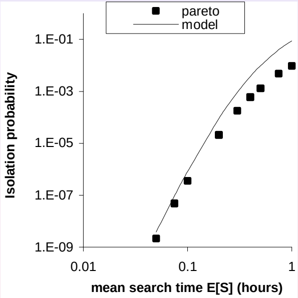


(n) constant S_i

Probability of Isolation V

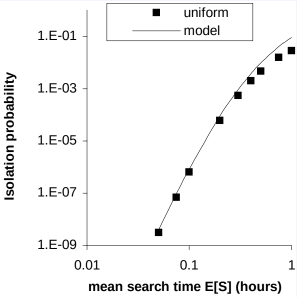


(o) uniform S_i

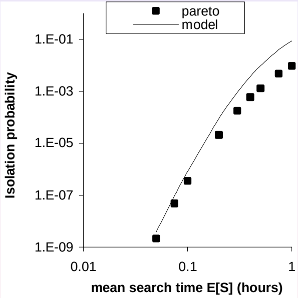


(p) Pareto S_i with $\alpha = 3$

Probability of Isolation V



(q) uniform S_i



(r) Pareto S_i with $\alpha = 3$

Observation

Heavy-tailed search delays are better than light-tailed

Probability of Isolation VI

Application to Pareto Lifetimes

- Notice that heavy-tailed (e.g., Pareto) lifetimes L_i imply stochastically larger residual lifetimes R_i

Probability of Isolation VI

Application to Pareto Lifetimes

- Notice that heavy-tailed (e.g., Pareto) lifetimes L_i imply stochastically larger residual lifetimes R_i
- For example, shape parameter $\alpha = 3$ leads to $E[R_i] = 2E[L_i]$

Probability of Isolation VI

Application to Pareto Lifetimes

- Notice that heavy-tailed (e.g., Pareto) lifetimes L_i imply stochastically larger residual lifetimes R_i
- For example, shape parameter $\alpha = 3$ leads to $E[R_i] = 2E[L_i]$
- The exponential result can be used as an upper bound on ϕ for heavy-tailed distributions of lifetime:

$$\phi \leq \frac{\rho k}{(1+\rho)^k + \rho k - 1}$$

where $\rho = E[L_i]/E[S_i]$

Probability of Isolation VII

Simulations

Table shows the minimum degree needed to guarantee a certain ϕ under Pareto lifetimes with $\alpha = 2.06$ and $k = 10$

Probability of Isolation VII

Simulations

Table shows the minimum degree needed to guarantee a certain ϕ under Pareto lifetimes with $\alpha = 2.06$ and $k = 10$

ϕ	Static	Lifetime	Mean search time $E[S_i]$		
	$p = 1/2$	node failure	6 min	2 min	20 sec
10^{-6}	20	Upper-bound model	10	7	5
		Simulations	9	6	4
10^{-9}	30	Upper-bound model	14	9	6
		Simulations	13	8	6

Examples I

Local Resilience Example

Consider a Chord system with $n = 10^{11}$ nodes, $E[L_i] = 30$ minutes, and $E[S_i] = 1$ minute

Examples I

Local Resilience Example

Consider a Chord system with $n = 10^{11}$ nodes, $E[L_i] = 30$ minutes, and $E[S_i] = 1$ minute

Static Result

Classical analysis with $p = 0.5$ requires $k = 37$ to ensure that a given node remains connected w.p. $1 - 1/n$

Examples I

Local Resilience Example

Consider a Chord system with $n = 10^{11}$ nodes, $E[L_i] = 30$ minutes, and $E[S_i] = 1$ minute

Static Result

Classical analysis with $p = 0.5$ requires $k = 37$ to ensure that a given node remains connected w.p. $1 - 1/n$

Lifetime Result

The same bound can be achieved with $k = 9$ as long as the tail of the lifetime distribution is exponential or heavier

Examples II

Global Resilience Example

Consider CAN with exponential lifetimes (mean 30 minutes), degree $k = 12$, and $n = 4096$ nodes

Examples II

Global Resilience Example

Consider CAN with exponential lifetimes (mean 30 minutes), degree $k = 12$, and $n = 4096$ nodes

Search time (min)	Actual $P(Y > N)$	Empirical ϕ	Model ϕ
6	.9732	.9728	.9728
7.5	.8218	.8224	.8215
8.5	.5669	.5659	.5666
9	.4065	.4028	.4016
9.5	.2613	.2645	.2419
10.5	.0482	.0471	.0424

Table: Comparison of model to simulations for $N = 10^6$ user joins

Examples III

Global Resilience Example (continued)

Assume now that the mean search delay is 1-minute and that 10^6 users join/leave per day

Examples III

Global Resilience Example (continued)

Assume now that the mean search delay is 1-minute and that 10^6 users join/leave per day

Model Result

Examples III

Global Resilience Example (continued)

Assume now that the mean search delay is 1-minute and that 10^6 users join/leave per day

Model Result

- Graph stays connected for $2,700$ years w.p. 0.9956

Examples III

Global Resilience Example (continued)

Assume now that the mean search delay is 1-minute and that 10^6 users join/leave per day

Model Result

- Graph stays connected for $2,700$ years w.p. 0.9956
- Mean delay between disconnections is 5.9 million years!

Outline

Effect of Varying Node Degree I

Degree Regularity and Resilience

- How does varying node degree among users improve/degrade resilience?

Effect of Varying Node Degree I

Degree Regularity and Resilience

- How does varying node degree among users improve/degrade resilience?
- In particular, is Gnutella with heavy-tailed degree more resilient than DHTs?

Effect of Varying Node Degree I

Degree Regularity and Resilience

- How does varying node degree among users improve/degrade resilience?
- In particular, is Gnutella with heavy-tailed degree more resilient than DHTs?

Theorem

Under the assumptions made earlier, degree-regular graphs are the most resilient for a given average degree $E[k_i]$

Varying Node Degree II

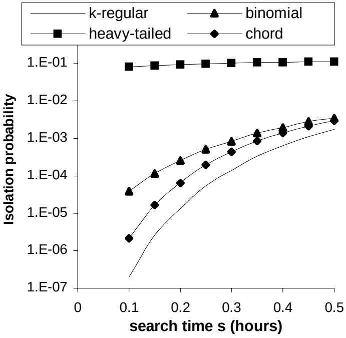
Simulations

Examine three degree-irregular systems with average degree $E[k_i] = 10$ and Pareto lifetimes with $E[L_i] = 0.5$ hours

Varying Node Degree II

Simulations

Examine three degree-irregular systems with average degree $E[k_i] = 10$ and Pareto lifetimes with $E[L_i] = 0.5$ hours



Varying Node Degree III

Implication

Varying node degree from peer to peer can have a positive impact on resilience *only* when these decisions are correlated with lifetimes

Varying Node Degree III

Implication

Varying node degree from peer to peer can have a positive impact on resilience *only* when these decisions are correlated with lifetimes

Improvement

Varying Node Degree III

Implication

Varying node degree from peer to peer can have a positive impact on resilience *only* when these decisions are correlated with lifetimes

Improvement

- Attach to neighbors with larger residual lifetime (age determines the expected residual lifetime of each user)

Varying Node Degree III

Implication

Varying node degree from peer to peer can have a positive impact on resilience *only* when these decisions are correlated with lifetimes

Improvement

- Attach to neighbors with larger residual lifetime (age determines the expected residual lifetime of each user)
- Unstructured systems: sample $2k$ users, sort by age, and choose top k to be your neighbors

Varying Node Degree III

Implication

Varying node degree from peer to peer can have a positive impact on resilience *only* when these decisions are correlated with lifetimes

Improvement

- Attach to neighbors with larger residual lifetime (age determines the expected residual lifetime of each user)
- Unstructured systems: sample $2k$ users, sort by age, and choose top k to be your neighbors
- Structured: do not let users of age smaller than a certain threshold to be responsible for DHT space

Conclusion

Findings

Conclusion

Findings

- P2P systems under churn almost surely remain connected as long as no user suffers isolation from the system

Conclusion

Findings

- P2P systems under churn almost surely remain connected as long as no user suffers isolation from the system
- Under all practical search times, k-regular graphs are much more resilient than traditionally thought

Conclusion

Findings

- P2P systems under churn almost surely remain connected as long as no user suffers isolation from the system
- Under all practical search times, k-regular graphs are much more resilient than traditionally thought
- Increasing the expected residual lifetime $E[R_i]$ of the neighbors is one simple way to improve resilience

Conclusion

Findings

- P2P systems under churn almost surely remain connected as long as no user suffers isolation from the system
- Under all practical search times, k-regular graphs are much more resilient than traditionally thought
- Increasing the expected residual lifetime $E[R_i]$ of the neighbors is one simple way to improve resilience
- Future work: model in-degree, examine lifetime-dependent neighbor selection, take node capacity into consideration