On the Partitioning Behavior of Churn-Based Peer-to-Peer Systems

Derek Leonard, Zhongmei Yao, Xiaoming Wang, and Dmitri Loguinov

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

March 6, 2006
Outline

1. Background
   - P2P Basics
   - Available Resilience Results

2. Global P2P Resilience
   - Classical Results
   - Churn Extension

3. Churn Resilience
   - Assumptions
   - Expected Time to Isolation
   - Probability of Isolation
   - Degree-Irregular Graphs

4. Conclusion
1. **Background**
   - P2P Basics
   - Available Resilience Results

2. **Global P2P Resilience**
   - Classical Results
   - Churn Extension

3. **Churn Resilience**
   - Assumptions
   - Expected Time to Isolation
   - Probability of Isolation
   - Degree-Irregular Graphs

4. **Conclusion**
Basic Operation
Basic Operation

- Decentralized P2P networks organize peers into distributed graphs
Basic Operation

- Decentralized P2P networks organize peers into distributed graphs
- Search performed by routing between users
Basic Operation

- Decentralized P2P networks organize peers into distributed graphs
- Search performed by routing between users

Classification
Basic Operation

- Decentralized P2P networks organize peers into distributed graphs
- Search performed by routing between users

Classification

- *Unstructured* networks (e.g., Gnutella, KaZaA) randomly connect users to each other
## Basic Operation
- Decentralized P2P networks organize peers into distributed graphs
- Search performed by routing between users

## Classification
- **Unstructured** networks (e.g., Gnutella, KaZaA) randomly connect users to each other
- **Structured** networks or **DHTs** (e.g., Chord, CAN, Pastry) connect peers based on distributed hashing of their identities into some virtual space
Design Tradeoff

(a) Unstructured
(b) Structured
Design Tradeoff

Flexibility of construction vs search efficiency
Design Tradeoff

Flexibility of construction vs search efficiency

(a) Unstructured
Design Tradeoff

Flexibility of construction vs search efficiency

(a) Unstructured

(b) Structured
Degree Regularity

A graph is called \( k \)-regular if each user has \( k \) neighbors and irregular otherwise (in directed graphs, regularity of in- and out-degree may be counted separately).
Degree Regularity

A graph is called $k$-regular if each user has $k$ neighbors and irregular otherwise (in directed graphs, regularity of in- and out-degree may be counted separately)
Degree Regularity

A graph is called $k$-regular if each user has $k$ neighbors and irregular otherwise (in directed graphs, regularity of in- and out-degree may be counted separately).

(a) Undirected, irregular
Degree Regularity

A graph is called $k$-regular if each user has $k$ neighbors and irregular otherwise (in directed graphs, regularity of in- and out-degree may be counted separately)

(a) Undirected, irregular

(b) Directed, 2-regular
Observation
Observation

P2P networks in the steady-state can be viewed as random graphs (even deterministic DHTs due to node failure or incompleteness are random)
Observation

P2P networks in the steady-state can be viewed as random graphs (even deterministic DHTs due to node failure or incompleteness are random)

(a) Full hypercube
Observation

P2P networks in the steady-state can be viewed as random graphs (even deterministic DHTs due to node failure or incompleteness are random)

(a) Full hypercube
(b) Random instance
Model

Assume a $k$-regular graph (relaxed later)

Each joining user $v$ gets $k$ random neighbors (set $M$)

Upon detecting failure of nodes in $M$, $v$ searches for a random replacement (unstructured P2P) or waits for zone repair (DHTs), in both cases obtaining a new neighbor.
Model

- Assume a $k$-regular graph (relaxed later)
Model

- Assume a $k$-regular graph (relaxed later)
- Each joining user $v$ gets $k$ random neighbors (set $M$) from among the existing users
Model

- Assume a $k$-regular graph (relaxed later)
- Each joining user $v$ gets $k$ random neighbors (set $M$) from among the existing users
- Upon detecting failure of nodes in $M$, $v$ searches for a random replacement (unstructured P2P) or waits for zone repair (DHTs), in both cases obtaining a new neighbor
Model

- Assume a $k$-regular graph (relaxed later)
- Each joining user $v$ gets $k$ random neighbors (set $M$) from among the existing users
- Upon detecting failure of nodes in $M$, $v$ searches for a random replacement (unstructured P2P) or waits for zone repair (DHTs), in both cases obtaining a new neighbor

Resilience Metrics
Model

- Assume a $k$-regular graph (relaxed later)
- Each joining user $v$ gets $k$ random neighbors (set $M$) from among the existing users
- Upon detecting failure of nodes in $M$, $v$ searches for a random replacement (unstructured P2P) or waits for zone repair (DHTs), in both cases obtaining a new neighbor

Resilience Metrics

- **Local**: node isolation (degree of some node is zero)
**Model**

- Assume a $k$-regular graph (relaxed later)
- Each joining user $v$ gets $k$ random neighbors (set $M$) from among the existing users
- Upon detecting failure of nodes in $M$, $v$ searches for a random replacement (unstructured P2P) or waits for zone repair (DHTs), in both cases obtaining a new neighbor

**Resilience Metrics**

- *Local*: node isolation (degree of some node is zero)
- *Global*: disconnection of the graph (not all pairs of users are connected)
Outline

1. Background
   - P2P Basics
   - Available Resilience Results

2. Global P2P Resilience
   - Classical Results
   - Churn Extension

3. Churn Resilience
   - Assumptions
   - Expected Time to Isolation
   - Probability of Isolation
   - Degree-Irregular Graphs

4. Conclusion
Node Failure (Static Model)

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

Node Failure (Dynamic Model)

Nodes depart asynchronously based on user browsing habits or interest. This type of failure combined with recovery strategies is generally called churn. Churn is a normal state of P2P networks.
Node Failure (Static Model)

- Traditional P2P resilience centers around uniform, independent, simultaneous node failure
Node Failure (Static Model)

- Traditional P2P resilience centers around uniform, independent, simultaneous node failure
- Each node is considered dead with probability $p$
Node Failure (Static Model)

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

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

Node Failure (Dynamic Model)
Node Failure (Static Model)

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

Node Failure (Dynamic Model)

- Nodes depart asynchronously based on user browsing habits or interest
Node Failure (Static Model)

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

Node Failure (Dynamic Model)

- Nodes depart asynchronously based on user browsing habits or interest
- This type of failure combined with recovery strategies is generally called *churn*
Node Failure (Static Model)

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

Node Failure (Dynamic Model)

- Nodes depart asynchronously based on user browsing habits or interest
- This type of failure combined with recovery strategies is generally called *churn*
- Churn is a normal state of P2P networks
Static Resilience
Static Resilience

Local results are trivial: \( \phi = P(\text{node } v \text{ is isolated}) = p^k \), where \( k \) is the degree of \( v \).
**Static Resilience**

- Local results are trivial: \( \phi = P(\text{node } v \text{ is isolated}) = p^k \), where \( k \) is the degree of \( v \)
- Global results available only for special types of networks (e.g., Massoulie 2003, Pandurangan 2001)
Static Resilience

- Local results are trivial: $\phi = P(\text{node } v \text{ is isolated}) = p^k$, where $k$ is the degree of $v$
- Global results available only for special types of networks (e.g., Massoulie 2003, Pandurangan 2001)

Dynamic Resilience
Static Resilience

- Local results are trivial: \( \phi = P(\text{node } v \text{ is isolated}) = p^k \), where \( k \) is the degree of \( v \).
- Global results available only for special types of networks (e.g., Massoulie 2003, Pandurangan 2001).

Dynamic Resilience

Lower bounds on the rate of user notification (Liben-Nowell 2002) and non-closed-form results for Chord’s connectivity (Krishnamurthy 2005).
Static Resilience

- Local results are trivial: \( \phi = P(\text{node } v \text{ is isolated}) = p^k \), where \( k \) is the degree of \( v \)
- Global results available only for special types of networks (e.g., Massoulie 2003, Pandurangan 2001)

Dynamic Resilience

Lower bounds on the rate of user notification (Liben-Nowell 2002) and non-closed-form results for Chord’s connectivity (Krishnamurthy 2005)

Open Question

*How does churn (distribution of user lifetimes and replacement delays) affect the connectivity of the system?*
Outline

1 Background
   • P2P Basics
   • Available Resilience Results

2 **Global P2P Resilience**
   • Classical Results
   • Churn Extension

3 Churn Resilience
   • Assumptions
   • Expected Time to Isolation
   • Probability of Isolation
   • Degree-Irregular Graphs

4 Conclusion
Definition

Let $\Phi(G)$ be the probability that an $n$-node graph $G$ is *connected* after static node/edge failure
Definition

Let $\Phi(G)$ be the probability that an $n$-node graph $G$ is connected after static node/edge failure.

Disconnection of Random Graphs
Definition

Let $\Phi(G)$ be the probability that an $n$-node graph $G$ is connected after static node/edge failure.

Disconnection of Random Graphs

Erdös and Rényi in the 1960s demonstrated that almost every (i.e., with probability $1 - o(1)$ as $n \to \infty$) random graph $G(n, p)$ is connected if and only if it has no isolated vertices:

$$\Phi(G) = P(G \text{ has no isolated nodes}) = \phi$$
Definition

Let \( \Phi(G) \) be the probability that an \( n \)-node graph \( G \) is **connected** after static node/edge failure.

Disconnection of Random Graphs

- Erdös and Rényi in the 1960s demonstrated that *almost every* (i.e., with probability \( 1 - o(1) \) as \( n \to \infty \)) random graph \( G(n, p) \) is connected if and only if it has no isolated vertices:

\[
\Phi(G) = P(G \text{ has no isolated nodes}) = \phi
\]

- *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.

(a) Single-node isolation
(b) Two-node disconnection

Figure: Disconnection of larger sets is significantly less likely.
Intuition

Conditional probability of partitioning along a set boundary \textit{while} not developing isolated nodes tends to zero.
Intuition

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

(a) Single-node isolation
**Intuition**

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

---

**Figure:** Disconnection of larger sets is significantly less likely.

(a) Single-node isolation  
(b) Two-node disconnection
Intuition (cont’d)
Intuition (cont’d)

*Well connected* graphs: each set $M$ has a set boundary $\partial M$ whose size is a certain increasing function of $|M|$.
Intuition (cont’d)

*Well connected* graphs: each set $M$ has a set boundary $\partial M$ whose size is a certain increasing function of $|M|$.

*Figure:* Boundary size $|\partial M|$ 4, 6, 8 nodes.
Intuition (cont’d)

Result does not apply to poorly connected graphs (e.g., cycles, trees)
Intuition (cont’d)

Result does not apply to *poorly connected* graphs (e.g., cycles, trees)
Intuition (cont’d)

Result does not apply to *poorly connected* graphs (e.g., cycles, trees)

**Figure:** Both small and large sets partition easily.
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:

<table>
<thead>
<tr>
<th>p</th>
<th>P(G is connected)</th>
<th>P(no isolated nodes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.5</td>
<td>0.99996</td>
</tr>
<tr>
<td>0.6</td>
<td>0.99354</td>
<td>0.99354</td>
</tr>
<tr>
<td>0.7</td>
<td>0.72650</td>
<td>0.72619</td>
</tr>
<tr>
<td>0.8</td>
<td>0.00043</td>
<td>0.00040</td>
</tr>
</tbody>
</table>
Deterministic Networks

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

<table>
<thead>
<tr>
<th>$p$</th>
<th>$P(G$ is connected)</th>
<th>$P$(no isolated nodes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.99996</td>
<td>0.99996</td>
</tr>
<tr>
<td>0.6</td>
<td>0.99354</td>
<td>0.99354</td>
</tr>
<tr>
<td>0.7</td>
<td>0.72619</td>
<td>0.72650</td>
</tr>
<tr>
<td>0.8</td>
<td>0.00040</td>
<td>0.00043</td>
</tr>
</tbody>
</table>
Application to P2P Graphs

All tested P2P systems (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.
Application to P2P Graphs

All tested P2P systems (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.
Definition

Let $\phi$ be the probability that of a joining user becomes isolated *during its lifetime* due to neighbor failure.
Definition

Let $\phi$ be the probability that of a joining user becomes isolated \emph{during its lifetime} due to neighbor failure.

Problem

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

Let $\phi$ be the probability that of a joining user becomes isolated *during its lifetime* due to neighbor failure.

Problem

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

Definition

Let $Z$ be the number of user joins before the first disconnection of the network.
Asymptotic Independence

Asymptotic Independence

Asymptotic Independence

Asymptotic Independence

Asymptotic Independence

Asymptotic Independence

Asymptotic Independence
Asymptotic Independence

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

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

Simple Model
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:

$$P(Z > N) \approx (1 - \phi)^N$$
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:
  \[
P(Z > N) \approx (1 - \phi)^N
  \]

  Knowledge of $\phi$ is all we need to understand dynamic resilience of P2P systems!
Example

- CAN with exponential lifetimes (mean 30 minutes)
Example

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

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

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

<table>
<thead>
<tr>
<th>Search time (min)</th>
<th>Actual $P(Z &gt; N)$</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>0.9732</td>
<td>0.9728</td>
</tr>
<tr>
<td>7.5</td>
<td>0.8118</td>
<td>0.8124</td>
</tr>
<tr>
<td>9</td>
<td>0.4065</td>
<td>0.4028</td>
</tr>
<tr>
<td>9.5</td>
<td>0.2613</td>
<td>0.2645</td>
</tr>
<tr>
<td>10.5</td>
<td>0.0482</td>
<td>0.0471</td>
</tr>
</tbody>
</table>

Table: Model vs simulations for $N = 10^6$ user joins
Example

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

<table>
<thead>
<tr>
<th>Search time (min)</th>
<th>Actual $P(Z &gt; N)$</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>0.9732</td>
<td>0.9728</td>
</tr>
<tr>
<td>7.5</td>
<td>0.8118</td>
<td>0.8124</td>
</tr>
<tr>
<td>9</td>
<td>0.4065</td>
<td>0.4028</td>
</tr>
<tr>
<td>9.5</td>
<td>0.2613</td>
<td>0.2645</td>
</tr>
<tr>
<td>10.5</td>
<td>0.0482</td>
<td>0.0471</td>
</tr>
</tbody>
</table>

Table: Model vs simulations for $N = 10^6$ user joins

Milestone

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

1 Background
   - P2P Basics
   - Available Resilience Results

2 Global P2P Resilience
   - Classical Results
   - Churn Extension

3 Churn Resilience
   - Assumptions
     - Expected Time to Isolation
     - Probability of Isolation
     - Degree-Irregular Graphs

4 Conclusion
Model Assumptions
Model Assumptions

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

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

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

\[ \text{degree} \]

\[ \text{time} \]
Node Arrival

Definition

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

Node $v$ enters at time $t_v$, then selects $k$ random neighbors from the system.
Node Arrival

- Node $v$ enters at time $t_v$ with random lifetime $L_v$
Node Arrival

- Node $v$ enters at time $t_v$ with random lifetime $L_v$
- It then selects $k$ random neighbors from the system
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.
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
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 $\phi$)
- How does varying node degree between users improve/degrade resilience?
- How to increase resilience of existing systems?
Pertinent Questions

- What is the average amount of time a node will spend in the system before becoming isolated?
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 $\phi$)
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 $\phi$)
- How does varying node degree between users improve/degrade resilience?
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 $\phi$)
- How does varying node degree between users improve/degrade resilience?
- How to increase resilience of existing systems?
Outline

1. Background
   - P2P Basics
   - Available Resilience Results

2. Global P2P Resilience
   - Classical Results
   - Churn Extension

3. Churn Resilience
   - Assumptions
   - Expected Time to Isolation
   - Probability of Isolation
   - Degree-Irregular Graphs

4. Conclusion
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 asymptotically small search delays, the following approximation holds for all lifetime and search distributions:

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

Notice that the main term that determines \( E[T] \) is the ratio \( \rho = E[R_i] / E[S_i] \).

Simulations show that the result is accurate even for large \( E[S_i] \).
Theorem

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

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

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

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

Notice

Main term that determines \( E[T] \) is the ratio \( \rho = E[R_i]/E[S_i] \)
Theorem

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

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

Notice

Main term that determines \( E[T] \) is the ratio \( \rho = \frac{E[R_i]}{E[S_i]} \)

Simulations Show

Result accurate even for large \( E[S_i] \approx E[L_i] \)
Expected Time to Isolation

Churn Resilience

Assumptions

Probability of Isolation

Degree-Irregular Graphs
Simulations

Average lifetime 30 min and $k = 10$ (1000-node system)

(a) uniform $S_i$
Simulations

Average lifetime 30 min and $k = 10$ (1000-node system)

(a) uniform $S_i$

(b) binomial $S_i$

Figure: Model vs simulation.
(a) exponential $S_i$
(a) exponential $S_i$ 

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

**Figure**: Model vs simulation.
**Definition**

Let $\delta$ be the keep-alive timeout and $d$ be the average inter-peer delay in the overlay.
Definition

Let $\delta$ be the keep-alive timeout and $d$ be the average inter-peer delay in the overlay.

Result for Chord-Like Systems

We immediately obtain from the main model:

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

Chord with $n = 1\text{ million}$, $d = 200\text{ ms}$, $E[R] = 1\text{ hour}$ (Pareto lifetimes with $E[L] = 30\text{ minutes}$)

<table>
<thead>
<tr>
<th>Expected Time to Isolation</th>
<th>$\delta$</th>
<th>$k$</th>
<th>$T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 sec</td>
<td>10 years</td>
<td>17 years</td>
<td>188 years</td>
</tr>
<tr>
<td>2 min</td>
<td>28 years</td>
<td>11 years</td>
<td>282 years</td>
</tr>
<tr>
<td>45 min</td>
<td>404,779 years</td>
<td>680 days</td>
<td>49 hours</td>
</tr>
</tbody>
</table>

Table:
Example

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

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

<table>
<thead>
<tr>
<th>Timeout $\delta$</th>
<th>$k = 20$</th>
<th>$k = 10$</th>
<th>$k = 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 sec</td>
<td>$10^{41}$ years</td>
<td>$10^{17}$ years</td>
<td>$188,034$ years</td>
</tr>
<tr>
<td>2 min</td>
<td>$10^{28}$ years</td>
<td>$10^{11}$ years</td>
<td>282 years</td>
</tr>
<tr>
<td>45 min</td>
<td>404,779 years</td>
<td>680 days</td>
<td>49 hours</td>
</tr>
</tbody>
</table>

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

1. Background
   - P2P Basics
   - Available Resilience Results

2. Global P2P Resilience
   - Classical Results
   - Churn Extension

3. Churn Resilience
   - Assumptions
   - Expected Time to Isolation
   - Probability of Isolation
   - Degree-Irregular Graphs

4. Conclusion
General Idea

- Recall that $\phi$ is the probability that a node $v$ becomes isolated during its lifetime.
General Idea

- Recall that $\phi$ 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.
General Idea

- Recall that \( \phi \) 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 \( Y_i(t) \).
Recall that $\phi$ 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 $Y_i(t)$.
Degree Evolution

Then the degree of node $v$ at time $t$ is $W(t) = \sum_{i=1}^{k} Y_i(t)$

Convenient to view as an alternating on/off process $W(t)$

$T_1$ $T_2$

on off on

Dmitri Loguinov
On the Partitioning Behavior of Churn-Based Peer-to-Peer Systems
Degree Evolution

Then the degree of node $v$ at time $t$ is $W(t) = \sum_{i=1}^{k} Y_i(t)$
Degree Evolution

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

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

For exponential lifetimes and $E[S_i] \to 0$, the probability of isolation $\phi$ converges to:

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

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

For exponential lifetimes and $E[S_i] \to 0$, the probability of isolation $\phi$ converges to:

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

where $\rho = \frac{E[L_i]}{E[S_i]}$

Verification
Theorem

For exponential lifetimes and $E[S_i] \to 0$, the probability of isolation $\phi$ converges to:

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

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.
Simulations

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

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

(a) exponential $S_i$
Simulations

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

(a) exponential $S_i$

(b) constant $S_i$
Observation: Heavy-tailed search delays are better than light-tailed.

(a) uniform $S_i$

(b) Pareto $S_i$ with $\alpha = 3$
(a) uniform $S_i$

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

Observation

Heavy-tailed search delays are better than light-tailed
Application to Pareto Lifetimes

Notice that heavy-tailed (e.g., Pareto) lifetimes $L_i$ imply stochastically larger residual lifetimes $R_i$.
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 $E[R_i] = 2E[L_i]$

\[
E[R_i] = 2E[L_i]
\]
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 $\phi$ for heavy-tailed distributions of lifetime:

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

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

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

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

<table>
<thead>
<tr>
<th>$\phi$</th>
<th>Static $p = 1/2$</th>
<th>Lifetime node failure</th>
<th>Mean search time $E[S_i]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-6}$</td>
<td>20</td>
<td>Upper-bound model Simulations</td>
<td>10 7 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Simulations</td>
<td>9 6 4</td>
</tr>
<tr>
<td>$10^{-9}$</td>
<td>30</td>
<td>Upper-bound model Simulations</td>
<td>14 9 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Simulations</td>
<td>13 8 6</td>
</tr>
</tbody>
</table>
Static Example

Classical analysis for $n = 10^{11}$ nodes and $p = 0.5$ requires $k = 37$ to ensure $\phi \leq 1/n$
Static Example

Classical analysis for $n = 10^{11}$ nodes and $p = 0.5$ requires $k = 37$ to ensure $\phi \leq 1/n$

Churn Example

- Consider a lifetime P2P system with $E[L_i] = 30$ minutes and $E[S_i] = 1$ minute
**Static Example**

Classical analysis for $n = 10^{11}$ nodes and $p = 0.5$ requires $k = 37$ to ensure $\phi \leq 1/n$.

**Churn Example**

- Consider a lifetime P2P system with $E[L_i] = 30$ minutes and $E[S_i] = 1$ minute.
- The same bound can be achieved with $k = 9$ as long as the tail of the lifetime distribution is exponential or heavier.
Global Resilience Example

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

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

<table>
<thead>
<tr>
<th>Search time (min)</th>
<th>Actual $P(Z &gt; N)$</th>
<th>$P(Z &gt; N)$ using empirical $\phi$</th>
<th>$P(Z &gt; N)$ using model $\phi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>.9732</td>
<td>.9728</td>
<td>.9728</td>
</tr>
<tr>
<td>7.5</td>
<td>.8218</td>
<td>.8224</td>
<td>.8215</td>
</tr>
<tr>
<td>8.5</td>
<td>.5669</td>
<td>.5659</td>
<td>.5666</td>
</tr>
<tr>
<td>9</td>
<td>.4065</td>
<td>.4028</td>
<td>.4016</td>
</tr>
<tr>
<td>9.5</td>
<td>.2613</td>
<td>.2645</td>
<td>.2419</td>
</tr>
<tr>
<td>10.5</td>
<td>.0482</td>
<td>.0471</td>
<td>.0424</td>
</tr>
</tbody>
</table>

Table: Comparison of model to simulations for $N = 10^6$ user joins
Global Resilience Example (continued)

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

1. Background
   - P2P Basics
   - Available Resilience Results

2. Global P2P Resilience
   - Classical Results
   - Churn Extension

3. Churn Resilience
   - Assumptions
   - Expected Time to Isolation
   - Probability of Isolation
   - Degree-Irregular Graphs

4. Conclusion
Degree Regularity and Resilience

- How does varying node degree among users improve/degrade resilience?
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?
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 $E[k_i]$.***
Simulations

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

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

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

Varying node degree from peer to peer can have a positive impact on resilience \textit{only} when these decisions are correlated with lifetimes.

Improvement
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)
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.
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.
Outline

1 Background
   • P2P Basics
   • Available Resilience Results

2 Global P2P Resilience
   • Classical Results
   • Churn Extension

3 Churn Resilience
   • Assumptions
   • Expected Time to Isolation
   • Probability of Isolation
   • Degree-Irregular Graphs

4 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 $E[R]$ 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.
Findings

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