A Network Analysis of the Gnutella P2P Network

Noah Zoarski

Introduction

The study of peer-to-peer (P2P) networks, particularly those as significant as the Gnutella network, holds profound implications for our understanding of digital communication and file-sharing ecosystems. P2P networks are fascinating because they embody the principles of decentralization, resilience, and community-driven growth much like human networks. P2P networks are not just technical constructs, but reflections of collective human behavior in digital space. Through this research, we explore the topology and dynamics of the Gnutella network to glean insights into how information and resources flow within such a decentralized system. Our findings have the potential to inform contributors and users of the significance of participation in robust, efficient, and equitable network structures, while simultaneously identifying key-players in peering.

Network Theory

Network theory offers a lens through which the complexity of P2P interactions can be understood. Social concepts like "six degrees of separation," "scale-free networks," and "small-world phenomena" are considered highly relevent to this field of study[1][3]. P2P networks are often posited as scale-free[2], characterized by a power-law degree distribution where most nodes have a few connections, and a few nodes (hubs) have a very high number of connections. This structure greatly impacts the network's robustness and the efficiency of file transfers, and is the primary focus of this article. Additionally, the "strength of weak ties" theory may explain the pivotal role of less-connected nodes in bridging distinct communities within the network, though will not be explored in the paper. Still, the hypotheses are structured around these theoretical underpinnings:

> **H_1:** The Gnutella network exhibits a scale-free topology, indicative of the presence of hub nodes that dominate the connectivity within the network.

H_2: Nodes with higher betweenness centrality play a crucial role in the network's data flow, acting as bridges in the file-sharing process.
H_3: The network demonstrates resilience against random node failures, a characteristic trait of scale-free networks.

These hypotheses are grounded in seminal works such as Albert-László Barabási discovery of scale-free networks and Granovetter's exploration of the strength of weak ties, among others.

Method and Data

To empirically investigate these hypotheses, we will deploy network analysis methods on a dataset derived from the Gnutella network. This dataset comprises an edgelist detailing the connections between individual nodes on a given day in 2002. We will employ centrality measures to identify important nodes within the network and explore the degree distribution to confirm or refute the scale-free nature of the network. Additionally, we will conduct robustness tests through simulated node failures to observe the network's response to disruptions. The choice of a Davidson-Harel layout algorithm in our visualization aims to prevent node overlap and ensures that edges do not obscure nodes, which significantly enhances the clarity and intelligibility of our network diagrams for accurate identification and interpretation of structural patterns within the network. Through this methodological approach, we aim to answer our research questions and validate our hypotheses, thereby contributing valuable insights to the field of network theory.

Analysis

Sampling the Data

The smallest dataset from Ripeanu, Foster, and Iamnitchi's Gnutella analysis contained thousands of nodes. With the intent to display the data on a single page in a PDF, including all nodes was considered infeasible. As such, a representative sample of one quarter of all nodes was taken to allow for an intelligble visualization in PDF format. The sample used in this instance is easily reproducable by specifying `set.seed(1)`.

```
# Read the edgelist into R
edgelist <- read.table("/Users/pangea/Documents/Education/Purdue/class/COM 411/FINAL PROJECT/

# Calculate the number of unique nodes
num_nodes <- length(unique(c(edgelist$FromNodeId, edgelist$ToNodeId)))

# Determine the number of nodes to sample (1/4 of the nodes, in this instance)
```

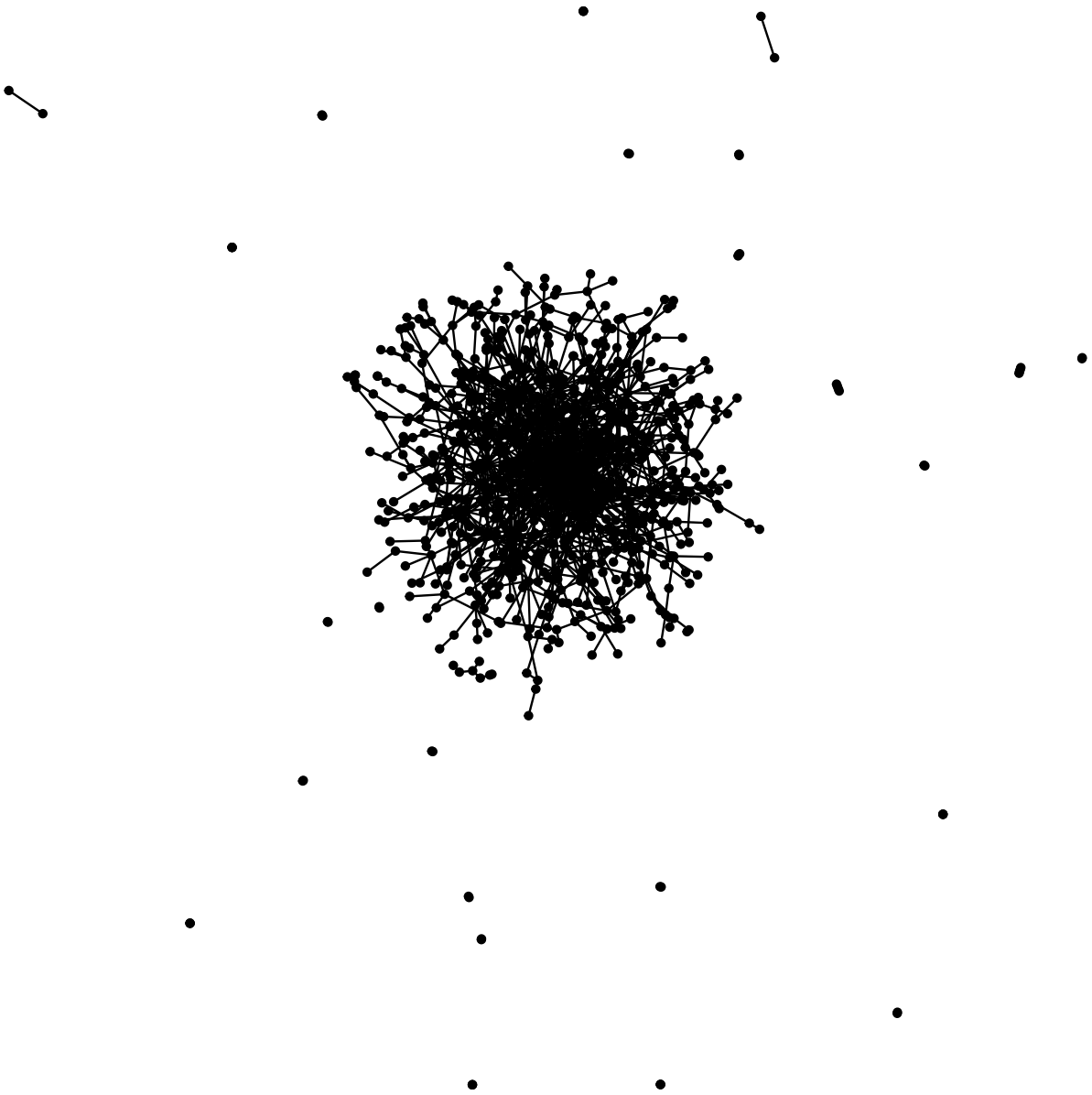
```
num_sample_nodes <- num_nodes / 4

# Get a list of unique nodes
nodes <- unique(c(edgelist$FromNodeId, edgelist$ToNodeId))

# Randomly sample the nodes
set.seed(1) # for reproducibility
sample_nodes <- sample(nodes, size = num_sample_nodes)

# Create a mask to filter edges that have both nodes in the sample_nodes list
edge_mask <- edgelist$FromNodeId %in% sample_nodes & edgelist$ToNodeId %in% sample_nodes

# Subset the edgelist to only include edges between the sampled nodes
sampled_edgelist <- edgelist[edge_mask, ]

# Write the sampled_edgelist to a file: p2p-GnutellaSAMPLED.txt (HARDCODED, pls fix!)
write.table(sampled_edgelist, file = "/Users/pangea/Documents/Education/Purdue/class/COM 411/
print("DONE")
```

```
[1] "DONE"
```

A general outline of the network structure is best visualized below, without custom layout.

```
# Example Visualization
ggraph(g, layout = 'igraph', algorithm = 'fr') +
  geom_edge_link() +
  geom_node_point() +
  theme_void()
```

```
Warning: Using the `size` aesthetic in this geom was deprecated in ggplot2 3.4.0.
i Please use `linewidth` in the `default_aes` field and elsewhere instead.
```



Are P2P Netoworks Scale-Free?

In order to assess the viability of hypothesis H1, we examine the network's degree distribution. The code below begins by preparing a dataset of nodes that excludes nodes without any connections, or peers that are not actively involved in any file transfers. The degree distribution is then transformed on a logarithmic scale in order to discern a power-law relationship. Linear regression is performed to calculate an R-squared value and line of best fit which serve to

demonstrate whether or not the network abides by a power-law: where a straight line validates the scale-free nature of the network.

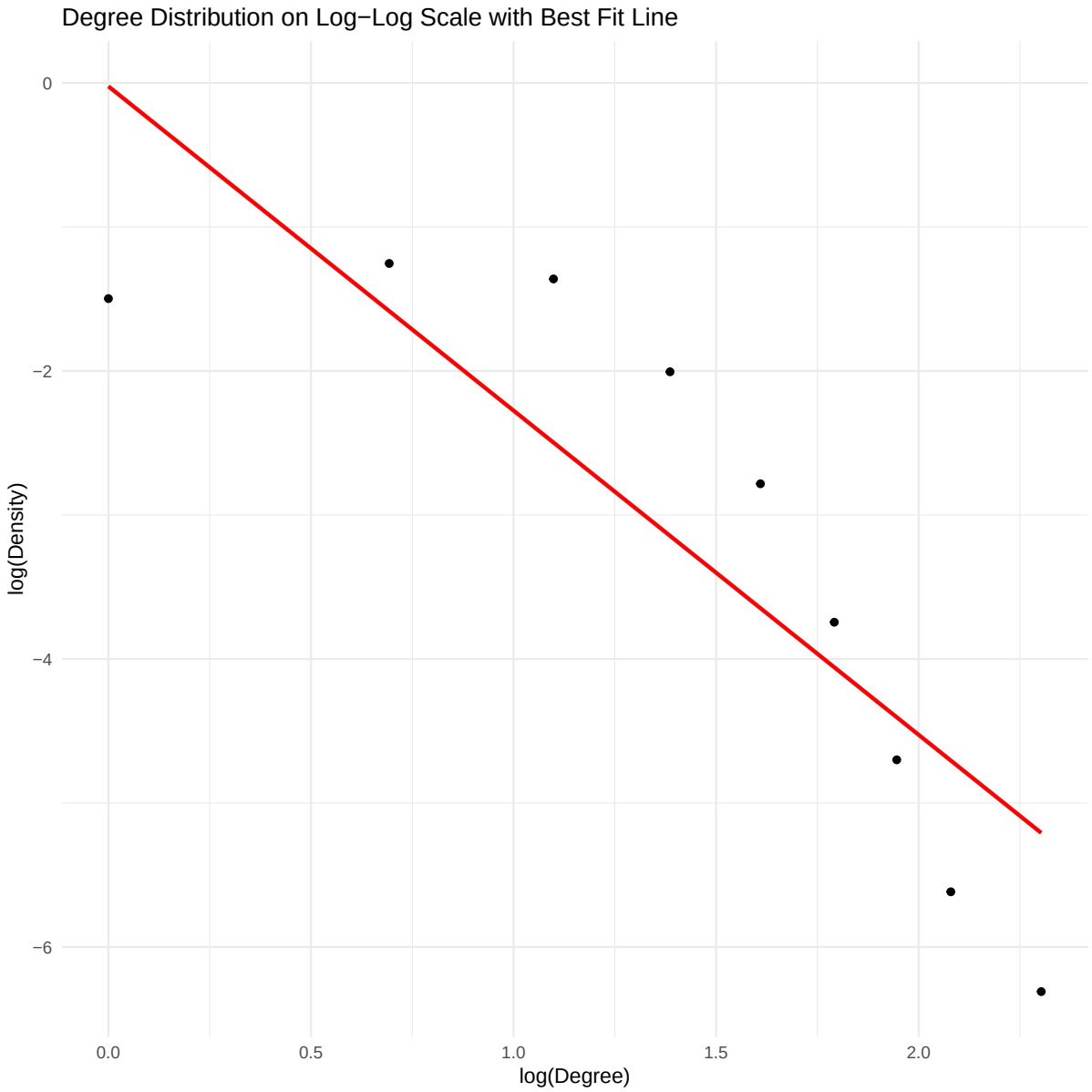
```
# Create object network_metrics as a dataframe containing the degree for all nodes
network_metrics <- g %>%
  activate(nodes) %>%
  mutate(degree = centrality_degree(),         # Node degree
         closeness = centrality_closeness(),   # Closeness centrality
         betweenness = centrality_betweenness()) %>%
  as.data.frame()

# Ensure that degree values are greater than 0 (Surprisingly, nodes with degree=0 exist in tl
network_metrics <- network_metrics %>% filter(degree > 0)

# Pre-calculate the log values
degree_density <- network_metrics %>%
  count(degree) %>%
  mutate(density = n / sum(n),
         log_degree = log(degree),
         log_density = log(density))

# Fit a linear model on the log-log transformed data
lm_model <- lm(log_density ~ log_degree, data = degree_density)

# Obtain the summary of the model
summary_lm <- summary(lm_model)

# Print the summary to see coefficients, p-values, etc.
print(summary_lm)
```

```
Call:
lm(formula = log_density ~ log_degree, data = degree_density)

Residuals:
    Min      1Q  Median      3Q     Max
-1.4739 -0.9112  0.3129  0.8638  1.1391

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.02385    0.81192  -0.029   0.9774
log_degree  -2.25142    0.50986  -4.416   0.0031 **
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.059 on 7 degrees of freedom
Multiple R-squared:  0.7358,    Adjusted R-squared:  0.6981
F-statistic:  19.5 on 1 and 7 DF,  p-value: 0.003097
```

```r
# Extract R-squared value and p-value
r_squared <- summary_lm$r.squared
p_value <- summary_lm$coefficients[2, 4]

# Plotting the degree distribution on a log-log scale
ggplot(degree_density, aes(x = log_degree, y = log_density)) +
  geom_point() +
  geom_smooth(method = 'lm', color = 'red', se = FALSE) +  # Fit a linear model without confi
  labs(title = "Degree Distribution on Log-Log Scale with Best Fit Line",
       x = "log(Degree)",
       y = "log(Density)") +
  theme_minimal() +
  annotate("text", x = Inf, y = Inf, label = paste("R^2 =", round(r_squared, digits = 3)),
           hjust = 11, vjust = 40)  # Adjust 'x' and 'y' to position the label as needed
```

```
`geom_smooth()` using formula = 'y ~ x'
```




```
# Output R-squared value to check it before adding it to the plot
cat("R^2:", r_squared, "\n")
```

```
R^2: 0.7358385
```

```
cat("P-value:", p_value, "\n")
```

```
P-value: 0.003097136
```

The visualization above presents compelling evidence in support of the scale-free hypothesis. The degree distribution, when plotted on a log-log scale, reveals a linear trend with an R-squared value of approximately 0.738, suggesting that the network's degree distribution does indeed follow a power-law. Importantly, the p-value associated with this analysis is 0.0031, which strongly supports the statistical significance of this finding.While the fit is not perfect—-as one might expect in the real-world—-it certainly aligns with the theoretical model of scale-free networks described by Barabási and Albert.

Basically, hubs (or, super-seeders) are tremendously beneficial for network efficiency and resiliency in file transfers. At the same time, their centrality also makes them points of vulnerability. In the context of internet piracy, for instance, targeting these hubs would be in law enforcement's best interest to disrupt the distribution of illegal content. Conversely, from the file-sharer's perspective, oveerreliance on such hubs could enable for a widespread and unexpected dissemination of malware. Or, in the case that an actor wanted to disable the network, a denial-of-service attack on any one hub could have profound impact on file availability and download speeds.

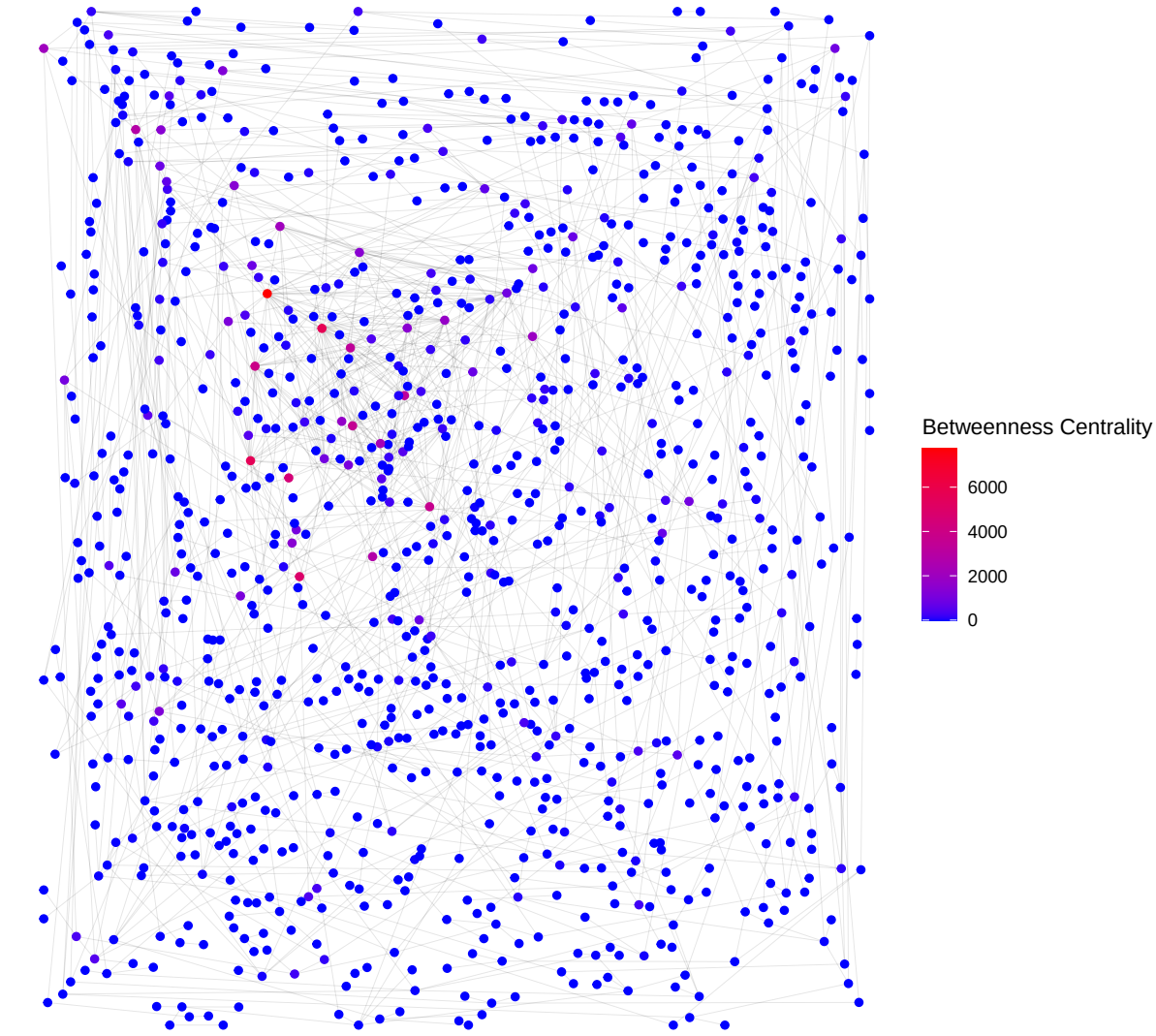
The scale-free topology informs us mostly about the network's resilience. While P2P networks are generally considered robust to random failures, such networks can be disproportionately crippled by targeted attacks on their most connected nodes.

Node Importance

Shifting focus to H2, we seek to identify which super-seeders increase the speed of the overall network. The nodes' betweenness centrality, a measure of their role in facilitating connections between other nodes, is calculated and visualized to demonstrate their influence over the network beyond mere connectivity. In other words, we anticipate to find numerous key intermediaries–seeders who might not necessarily posses the most connections but are vital in maintaing network cohesion and efficiency (download speeds).

```
# Calculate basic network metrics
g <- g %>%
  activate(nodes) %>%
  mutate(degree = centrality_degree(),            # Node degree
         closeness = centrality_closeness(),      # Closeness centrality
         betweenness = centrality_betweenness())  # Betweenness centrality

# Convert the graph to a dataframe containing the degree for all nodes
network_metrics <- g %>%
  activate(nodes) %>%
  as.data.frame()

# Use a continuous color scale to represent betweenness centrality
```

```
ggraph(g, layout = 'igraph', algorithm = 'dh') +
  geom_edge_link(edge_width = 0.05, alpha = 0.1) +
  geom_node_point(aes(color = betweenness)) +
  scale_color_gradient(low = "blue", high = "red") +
  theme_void() +
  labs(title = "Network Nodes Colored by Betweenness Centrality",
       color = "Betweenness Centrality")
```



Our analysis indeed found several key seeders within the Gnutella network. While these nodes may not boast the highest number of direct connections, their placement within the network

amplifies their importance, validating H2. The nodes identified through this analysis likely represent the network's super-seeders—entities that not only share a vast amount of content but also enable significant portions of the network's traffic to pass through them, reinforcing the network's reliance on their uninterrupted operation.

Each node's central positioning underscores a duality: these nodes enhance file-sharing efficiency while contributing to an overreliance on a few nodes. Again, these nodes could be targeted by law enforcement or other actors to prevent the distribution of illegal content or cut off a segment (cluster) of the network from access to resources. Conversely, should any hub be compromised or decide to distribute malware within their seeded torrents, users who neglect best practices in file verification and digital signatures are at significant risk. This hypothetical underscores the inherent dangers in a network where substantial trust and traffic are concentrated in just a few nodes. The findings not only advocate for greater participation from users with powerful hardware and speedy internet connections, but also reinforce the necessity for users to adhere to best practices in digital signatures. These insights suggest that P2P networks might not be as decentralized as one might expect.

Network Resiliency

Understanding how peer-to-peer (P2P) networks respond to random failure is crucial for assessing their robustness and resilience. This section of analysis focuses on the network's resilience by simulating random node failures—a scenario that mirrors potential real-world disruptions, such as server outages or users disconnecting from the network. Specifically, we investigate the effects of these failures on two key network properties: the size of the largest connected component, which reflects the network's ability to remain interconnected, and the average path length, which influences the efficiency of data transfer across the network. The code provided below systematically removes 10% of the nodes from the network to simulate these random failures and measures the subsequent changes in these metrics.

```
library(tidygraph)
library(igraph)

# Function to simulate random failures
simulate_random_failures <- function(graph, fraction_to_remove) {
  num_nodes_to_remove <- as.integer(vcount(graph) * fraction_to_remove) # "all nodes in the g
  nodes_to_remove <- sample(V(graph), num_nodes_to_remove)
  graph_after_failures <- delete_vertices(graph, nodes_to_remove)
  return(graph_after_failures)
}

# Function to calculate metrics
calculate_network_metrics <- function(graph) {
```

```
  largest_comp_size <- max(components(graph)$csize)
  avg_path_length <- mean_distance(graph, directed = FALSE)
  return(c(largest_comp_size, avg_path_length))
}

# Calculate baseline metrics
baseline_metrics <- calculate_network_metrics(g)
print(paste("Baseline Largest Component Size:", baseline_metrics[1]))
```

```
[1] "Baseline Largest Component Size: 940"
```

```
print(paste("Baseline Average Path Length:", baseline_metrics[2]))
```

```
[1] "Baseline Average Path Length: 6.6430259243904"
```

```
# Set number of trials
num_trials <- 30
results <- matrix(nrow = num_trials, ncol = 2)  # Prepare a matrix to store results

# Simulate multiple failures and calculate metrics
set.seed(123) # Ensure reproducibility
fraction_to_remove <- 0.1 # Remove 10% of nodes

for (i in 1:num_trials) {
  simulated_graph <- simulate_random_failures(g, fraction_to_remove)
  results[i, ] <- calculate_network_metrics(simulated_graph)
}

# Calculate average results
average_results <- colMeans(results)
names(average_results) <- c("Average Largest Component Size", "Average Path Length")

# Print the average results and compare to baseline
print("Average metrics after failures:")
```

```
[1] "Average metrics after failures:"
```

```
print(average_results)
```

```
Average Largest Component Size            Average Path Length
                     793.600000                       6.987633
```

```
print("Comparison to baseline:")
```

```
[1] "Comparison to baseline:"
```

```
print(paste("Change in Largest Component Size:", baseline_metrics[1] - average_results[1]))
```

```
[1] "Change in Largest Component Size: 146.4"
```

```
print(paste("Change in Average Path Length:", baseline_metrics[2] - average_results[2]))
```

```
[1] "Change in Average Path Length: -0.344607302358128"
```

The statistical analysis demonstrates that the Gnutella network's structure responds well to random disruptions. The random removal of 10% of nodes reduced the average largest component size by only 15.7%, and increased the average path length by a mere 5.2%. Initially, our network displayed a largest component size of 940 nodes and an average path length of approximately 6.643. And, after simulating random failures across 30 trials, the average largest component size observed was approximately 793 nodes, while the average path length was only slightly increased to about 6.987. These results affirm our hypothesis, H3, regarding the network's robustness, as the network largely preserves its connectivity and functional integrity despite significant node losses.

Resilience is necessary in decentralized systems, such as Gnutella, where maintaining service continuity through random failure is crucial. The slight increase in average path length suggests that, while the network remains largely intact, the efficiency of data transmission experiences marginal losses. The effect on file availability is unknown, but should be expected (although minimal). This is an expected outcome as the network compensates for the loss of nodes by rerouting connections, which typically leads to longer path lengths.

This analysis not only supports the theoretical model of scale-free networks as described by Barabási and Albert but also provides a quantitative basis for evaluating the resilience of real-world network infrastructures against random disturbances. Further investigations could explore different rates of node removal or the impact of targeted attacks to fully understand the spectrum of network resilience in P2P systems.

Conclusion

Our analysis explored the intricate dynamics of the Gnutella P2P network, shedding light on its complexity and resilience. Through a blend of network theory and empirical investigation, we've confirmed hypotheses regarding the network's structural makeup and operational robustness. The scale-free nature of the network was demonstrated, underscoring both the network's efficiency and vulnerability. Moreover, our exploration of betweenness centrality identified the specific nodes critical in maintaining network cohesion, while also recognizing them as potential points of exploitation.

Examining the network's resilience to random node failures further validated the network's robustness in sustaining service continuity in the decentralized system. The network displayed remarkable resilience to significant disruptions, underscoring its capacity for fault-tolerance and functionality under ordinary adverse conditions.

Looking ahead, further research could benefit from a more detailed dataset that includes specific information about file transfers, such as file types, sizes, and other temporal dynamics of file sharing. This would provide deeper insights into the nature of the content being shared and the social dynamics within the network. Questions such as the influence of popular culture on illegal file-sharing activities, the role of internet speeds in node centrality, and the implications of file-sharing on copyright infringement could be explored to provide a richer understanding of P2P networks.

References

1. Montresor, A., & Jelasity, M. (2009). PeerSim: A scalable P2P simulator. *2009 IEEE Ninth International Conference on Peer-to-Peer Computing* (P2P '09). https://doi.org/10.1109/P2P.2009.5284506
2. "Scale-Free Networks." (2010). *P2P Foundation.* Retrieved from https://wiki.p2pfoundation.net/Scale-Free_Networks
3. Schollmeier, R. (2001). A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. *Proceedings First International Conference on Peer-to-Peer Computing* (P2P '01). https://doi.org/10.1109/P2P.2001.990434
4. Ripeanu, M., Foster, I., Iamnitchi, A. (2002). Mapping the Gnutella Network: Properties of Large-Scale Peer-to-Peer Systems and Implications for System Design. *IEEE Internet Computing (Volume: 6, Issue: 1)* Retrieved from https://doi.org/10.48550/arXiv.cs/0209028
5. tom91. (2019). Using log-log graph to find equation of power law relationship?. *CrossValidated.* https://stats.stackexchange.com/questions/392464/using-log-log-graph-to-find-equation-of-power-law-relationship

Glossary

1. **Torrent**: A torrent file is a metadata file used by BitTorrent clients to initiate the download of the actual content files, which are not stored on a central server but are distributed among users. A torrent contains information about pieces of the file(s) such as their size, network locations of trackers (which are servers that coordinate the distribution of files among users), and other data necessary to download and assemble the file.
2. **Seeder**: In the context of BitTorrent, a seeder is a user who has a complete copy of the file being shared across the network and continues to upload it to other peers. Seeders are crucial to the health of a BitTorrent network because they increase the availability of file segments to new downloaders.
3. **Super-seeder**: A super-seeder is a seeder mode designed to maximize the distribution efficiency of a torrent. When enabled, this mode allows the seeder to send out pieces of a file to different peers, ensuring that all pieces are available in the network at least once before sending out duplicates. This strategy helps in spreading out the file quickly especially when there are many downloaders and not enough seeders.
4. **Digital Signature**: A digital signature is a mathematical scheme for verifying the authenticity of digital messages or documents. In P2P networks, digital signatures are used to verify that a file has not been altered in transit, ensuring that what was downloaded is what the seeder intended to distribute. This is crucial for maintaining trust in the integrity of files exchanged over decentralized networks.