CHAPTER 3

MODELING AND SIMULATION OF PUBLIC P2P CONTENT SHARING NETWORK

3.1 BASICS OF PUBLIC P2P NETWORK

The public P2P network for large scale distributed applications has recently gained widespread interest due to the success of P2P content sharing, media streaming, and telephony applications. There are a large range of other applications under development or being proposed like IPTV, Video-on-Demand in the areas of P2P architecture, search and queries, incentive mechanisms, multimedia streaming, service oriented architectures, collaboration to share non-storage resources, mobile P2P, theory and analysis, and P2P databases. Among all existing P2P applications, content sharing attracts quite a large number of users due to low cost of implementation setup and freedom in bandwidth capacity of peers.

A public P2P network is a logical overlay network on top of a physical network that supports anonymity and autonomy of each peer. The underlying P2P architecture shares features such as decentralization, sharing of end system resources, virtualization, and self-organization. There is no priority or preference of any peer to join or leave the network. As the name suggests, no one can claim for any credit or damages caused by content downloading through public P2P networks. The network can be monitored by any peer to collect (a) the statistics of the network, or (b) the information of a particular peer for personal use, or (c) in the direction of any agency. These networks support unstructured architecture with any kind of topology that is shaped during overlay formation. Moreover, there is no restriction
on the types of content that can be shared. The majority of deployed P2P applications use unstructured overlays. There are many public P2P clients available for content sharing in which BitTorrent is one of the most used P2P clients with millions of users. Commonly used terminologies in P2P content sharing networks are listed below (pages 361 to 365 of [1]).

- **Peer**: Any node or host connected to the network with sufficient processing power, and storage capacity for content sharing application.
- **Seed or Provider**: The peer which has the complete set of files or content for sharing.
- **Bootstrap peer**: The peer that initiates the creation of a new overlay or helps in the joining process of a new peer in the overlay network.
- **Churn rate**: The rate at which peers join or leave the P2P network in an unpredictable pattern that changes the peer population of overlay network.
- **Malicious Peer**: Any peer that challenges integrity of the network.
- **Overlay Network**: A logical network built on top of the existing physical network.
- **Homogeneous Network**: A network connected with peers having similar amount of capacity in usage bandwidth, computing power, and storage space.
- **Heterogeneous Network**: A network formed by peers having diverse usage bandwidth, computing power, and storage space.
- **Resource sharing**: Each peer in the P2P network contributes to system resources like processing power, storage capacity, and network link capacity in the operation of the P2P system. Ideally this resource sharing is proportional to the peer’s usage of the P2P system.
- **Structured overlay**: An overlay in which peers cooperatively maintain routing information on how to reach all peers in the overlay.
- **Unstructured overlay**: An overlay in which a peer relies only on its adjacent peers for delivery of messages to other peers in the overlay.
- **Partitioned Overlay Network**: An overlay network partition is the separation of two or more sets of peers into separate overlays. Partitions can occur due to sudden failure of connections between the two sets of peers due to a network
failure, sudden massive churn, or a denial-of-service attack on the overlay itself.

- **Decentralization**: The behavior of the P2P system is determined by the collective actions of peers, and there is no central control point. Some systems however secure the P2P system using a central login server. The ability to manage the overlay and control its operation may require centralized elements.

- **Symmetry**: Peers assume equal roles in the operation of the P2P system. In many designs, this property is relaxed by the use of special peer roles such as super peers or relay peers.

- **Autonomy**: Participation of the peer in the P2P system is determined locally, and there is no single administrative context for the P2P system.

- **Self-organization**: The organization of the P2P system increases over time using local knowledge and local operations at each peer, and no peer dominates the system.

- **Scalable**: This is a pre-requisite of operating P2P systems with millions of simultaneous peers, and this means that the resources used at each peer exhibit a growth rate as a function of overlay size. Ideally, there is no limit for scalability in P2P systems.

- **Stability**: Within a maximum churn rate, the P2P system should be stable, i.e., it should maintain its connected graph and be able to route deterministically within a practical hop-count bounds.

### 3.2 PROBLEMS IN PUBLIC P2P NETWORK

Content sharing has emerged as a popular application in P2P networks. The increasing number of free-riders and untrustworthy peers in content sharing network reduces the credibility of the shared contents. Also, malicious peers pose the danger of security violation and bandwidth wastage by sharing polluted or infected contents in the network for their personal benefits. In P2P content sharing networks,
frequently encountered problems are free-riders and content pollution which consume huge bandwidth. Recently, numerous studies discussed that the following problems are responsible in bringing P2P content sharing overlay network to a full halt [99]-[104], [108], [152], [153]:

(i) **Free-riders**: Peers involved only in downloading the content and not interested in uploading them. They consume resources of other peers in excess of the amount of resources they contribute to the P2P system.

(ii) **Content Pollution**: Accidental corruption of contents due to malicious codes (auto-executable computer program intended to cause undesired effects, security breaches or damage to a system) or transmission errors.

(iii) **Content Poisoning**: Deliberate injection of decoys or bad files in the network for stopping piracy.

(iv) **Denial of Service attack**: Any attempt to make content provider unavailable to the churns.

(v) **Distributed Denial of Service attack**: Any attempt to organize a group of peers without their knowledge in the overlay network in launching Denial of Service attack on an external peer that is not a part of the overlay network.

(vi) **Content Availability**: It indicates how easily content is found in the network and downloaded through queries.

(vii) **Flash Crowd**: A sudden network traffic flow caused by a significant arrival of churns attempting to download the same content.

(viii) **Virus Spreading / Infected Content**: Any content intentionally hiding malicious codes to affect the integrity of the peer’s system that downloads it.

(ix) **Peer Availability**: When the peers leave the network after acquiring all blocks of a content they require, the new comers are left with unfinished downloads until a peer with complete set of content rejoins the overlay network.

(x) **Sybil Attack**: Any peer forging its identity by creating multiple pseudonymous entities to gain a disproportionately large influence from reputation based systems.
(xi) **Collusion attack**: A certain number of peers collaborate with the goal to defraud the trust or reputation management system.

(xii) **Identity Attack**: A malicious peer exploits the reputation system by hijacking the information of any targeted peer.

(xiii) **Whitewasher**: A user who leaves a P2P system and rejoins it after acquiring a new identity to avoid reputational penalties.

(xiv) **Eclipse attack**: An attack on an overlay network in which the attacker controls a large fraction of neighboring peers.

### 3.3 PROPOSED UNIFIED TRUST MANAGEMENT MODEL

In public P2P systems, there are many independent research works under study or in progress in the areas of search and query processing, incentive mechanisms, trust, anonymity, privacy, mobility, fault tolerance and measurement of P2P traffic. In this thesis, a trust management model is proposed which combines the above said areas as a unified approach. The unified trust management model works with different phases that cover each of the following issues efficiently.

(a) **Search and Query Processing**: Search is perhaps the most fundamental service in an overlay. A wide range of techniques are used, like basic keyword search, semantic search, database query processing, and indexing mechanisms in P2P architecture. In the proposed model, database query processing and indexing mechanism are used.

(b) **Incentive Mechanisms**: In practice, peers are not altruistic. So, techniques to ensure fair and mutual resource sharing have been proposed. An important category is incentive mechanisms which allow peers to participate proportionally to their resource contribution compared to other peers. A reliable incentive and penalty mechanism is considered in the proposed model.
(c) **Trust, Anonymity, and Privacy:** In most P2P overlays, peers are in autonomous security domains. Anonymity and privacy expectation of the peers creates problem with the trust management process. Peer reputation management is an important category of enabling trust. The central idea of this thesis is to manage and monitor the trust among the peers in the network. Anonymity and privacy is considered only to some extent as trust management is not possible in a purely anonymous state.

(d) **Mobile P2P:** When peers roam in ad hoc networks, the efficiency and stability of the overlay is affected. The growing importance of mobile and ad hoc networking for many applications has attracted research on the use of P2P overlays in Mobile Ad hoc Networks (MANETs). The proposed model has considered mobile peers with appropriate weightage in P2P networks.

(e) **Fault Tolerance in P2P Networks:** Uneven workloads and instability due to churn are some of the practical issues in operating large-scale P2P systems. Various load balancing techniques have been studied to adapt in uneven workloads. Stabilization of the overlay and automatically correcting network partitions are the highlights of the proposed unified model.

(f) **Measurement and P2P Traffic Characteristics:** Another topic of great practical interest is the network traffic of P2P applications, which has become a dominant flow on the Internet. Efficient bandwidth usage is a main concern for Internet Service Providers (ISPs) and peers. Bandwidth wastage is taken as an important problem in the model.

### 3.3.1 Model Description

In the proposed UNified Trust management strategY (UNITY) for public P2P content sharing networks, the assumption is made that the centralized server or
tracker\textsuperscript{1}, authentication server, and database server are highly secure and trustable. The centralized server or tracker is employed with additional duties of authentication and database maintenance. The existing trust management models are unified to a single model to perform trust management along with ensured security for the peers in the P2P network as shown in Figure 3.1. The model has initializing and stabilizing phase for the peers in the P2P network that utilizes controlled scalability. The decaying phase in which the peers leave the network indefinitely is regulated by the model without disrupting content availability. The terms, notations, and definitions used in the model are shown in Table 3.1.

![Figure 3.1](image)

**Figure 3.1** Peers attempt to connect with public P2P networks through a centralized server.

### 3.3.2 Initialization Phase

When a peer (provider or seed) is willing to share a particular content, it generates a meta-data (similar to .torrent file in BitTorrent-like systems) for the content to be shared. The generated meta-data contains information about provider

\textsuperscript{1}Centralized Server and Tracker are used interchangeably throughout the rest of the thesis.
ID, provider IP address, file name, size, hash values, and number of tokens. The provider needs to authenticate with its login ID and password to the centralized server in order to generate and publish its meta-data in the web server.

Table 3.1 Notations and Definitions used in public P2P UNITY model

<table>
<thead>
<tr>
<th>Term or Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>P_ID</td>
<td>Provider’s identity</td>
</tr>
<tr>
<td>P_IP</td>
<td>Provider’s IP address</td>
</tr>
<tr>
<td>R_ID</td>
<td>Requestor’s identity</td>
</tr>
<tr>
<td>R_IP</td>
<td>Requestor’s IP address</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the Content</td>
</tr>
<tr>
<td>Size</td>
<td>Size of the Content</td>
</tr>
<tr>
<td>H_C</td>
<td>Hash Values of the Content</td>
</tr>
<tr>
<td>G_T</td>
<td>Number of Tokens to generate</td>
</tr>
<tr>
<td>T_n</td>
<td>Token Number</td>
</tr>
<tr>
<td>P_TV</td>
<td>Provider’s Trust Value</td>
</tr>
<tr>
<td>R_TV</td>
<td>Requestor’s Trust Value</td>
</tr>
<tr>
<td>V_TV</td>
<td>Voting Trust Value</td>
</tr>
<tr>
<td>QC</td>
<td>Quality Check of the Content</td>
</tr>
<tr>
<td>t_s</td>
<td>Time Stamp of token creation</td>
</tr>
<tr>
<td>val</td>
<td>Validity of the token</td>
</tr>
<tr>
<td>ON_i</td>
<td>Unique Overlay Network number</td>
</tr>
<tr>
<td>OE_m</td>
<td>Overlay Expansion Method</td>
</tr>
</tbody>
</table>

In the existing BitTorrent-like systems, offline generation of .torrent file is allowed and the generated .torrent file is stored in the peer’s node. But, in our model, the generated meta-data is directly uploaded in the web server maintained by the tracker. As any provider is prohibited from storing meta-data generated by it, the meta-data is inaccessible to unauthenticated peers. This process helps in avoiding a collusive attack (a certain number of peers collaborate with the goal to defraud the management system) performed by any provider. The generated meta-data with
detailed description of the content is published through an authenticated login given by meta-data = \{PID, PIP, Name, Size, HC, GT\}.

The database server maintains the trust value of every registered peer. When a peer needs to download any content, it has to authenticate itself with the authentication server to get access to the peer’s profile page containing the meta-data according to its trust value. If any peer fails to get successful authentication or wants to be in an anonymous state, it is directed to the restricted or newcomer profile page containing only the verified and genuine meta-data as shown in Figure 3.2. The difference between the newcomer and anonymous mode is that the newcomer is a registered user and it can claim trust points for its successful service, whereas, anonymous peers cannot be involved in the voting process or claim trust values. By default, the number of tokens to be generated for a particular content is decided upon the provider’s bandwidth and size of the content. The option of manually deciding the number of tokens to be generated by any peer is allowed by the calculation of the trust value on par with its contribution.

### 3.3.2.1 Overlay Formation

In the unstructured public P2P networks, peers may join or leave the overlay network at any time. Overlay networks use join and leave protocols for the neighbors to update their routing state, and the protocols help newly joined peers to quickly make connections with active neighbors. A candidate peer needs to discover an existing peer to join the overlay network. The process of discovering and contacting an existing peer is called peer bootstrap, and involves mechanisms outside the overlay, such as contacting a well-known bootstrap server or tracker. When a peer joins the overlay network, it typically receives its initial routing and object state from one or more peers designated by the bootstrap peer. After that, the peer modifies its state based on the operation of the overlay protocols. When a peer leaves the overlay, it may signal its neighbors using a leave protocol. The neighbors then make
changes to their routing state, and the object state may be migrated or replicated as well. If a peer is disconnected without notification, neighbors use a heartbeat mechanism [154] to detect the departure, and trigger the corresponding routing and object state updates.

In the proposed UNITY model, the centralized server publishes the name, size, and description of the content in its content site along with the provider’s trust value. This information (Info) is visible only to the authenticated peers whose trust values are greater than the provider’s trust value. The system works similar to the social networking concept in which each peer has a profile page to show its accessible contents list according to its trust values as in Figure 3.2. The update of new content availability is sent through the News Feed. In case the provider has the highest trust value, the tracker decides the threshold for trust values to show the meta-data depending upon the allowable initial network size.

![Figure 3.2](image1.jpg)

**Figure 3.2** A Typical profile page of a peer for different trust values (a) High trust value (b) Medium trust value, and (c) Newcomer or anonymous mode.

Once a peer decides to download the content, it requests for a token to access the content. The tracker generates the token depending upon the provider’s allowed number of overlay network size in a first come first served basis. The tracker also confirms the IP address of the provider before generating the token, as the IP addresses are likely to change in a mobile environment. The information visible to the peers and the token generated are given by
\[Info = \{\text{Name, Size, } P_{TV}, \text{ Description}\},\]

\[Token = \{T_n, P_{IP}, R_{IP}, t_n, \text{Name, Size, } H_C, \text{ val, } ON_i\},\] respectively.

The peer after receiving the token uses the information from the token to contact the provider for bootstrapping it into the overlay network. The provider confirms the overlay number with the tracker for the first peer and initiates the bootstrapping operation. The subsequent peers are bootstrapped immediately if they have a valid token.

Churn is the arrival and departure of peers to and from the overlay, which changes the peer population in the overlay network. Overlay network maintenance is the operation of the overlay to repair and stabilize the overlay routing state in response to the churn. The overhead for overlay network maintenance increases as the churn rate increases. It also increases proportionally to the routing state maintained by each peer, which is in turn proportional to the size of the overlay, and the degree of each peer. There are techniques to reduce the churn, such as incentives for peers to stay connected to the overlay. In addition, newly joined nodes can be quarantined, treated as client-only nodes, either due to limited capacity or until the peer reaches a lifetime threshold. This relies on peer lifetime distribution being heavy tailed, which has been found to occur in practice.

3.3.2.2 Voting Process

Initially, a limited number of trusted peers with valid tokens are allowed to download the content by forming an overlay among them. When a specified number of tokens are generated by the tracker, the peers requesting new tokens have to wait for the decision from the first voting process. Once the content has been fully downloaded by the peers, they check the content for its genuineness or quality by testing it through their trusted anti-virus software or sand-box methods. If the trusted
peers are convinced with the content for good quality, they vote positively in favor of the provider and share the content as per the decision made by the tracker through decision support (Algorithm 3.2). The initial voting process (as in Algorithm 3.1) for the provider uses direct trust values as shown in Table 3.2.

<table>
<thead>
<tr>
<th>Voting Trust Value ($V_{TV}$)</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3</td>
<td>Polluted / Poisoned content with severe threat</td>
</tr>
<tr>
<td>-2</td>
<td>Poisoned content with less threat</td>
</tr>
<tr>
<td>-1</td>
<td>Polluted content</td>
</tr>
<tr>
<td>0.5</td>
<td>Already available content shared for wide distribution with minor up-gradation</td>
</tr>
<tr>
<td>1</td>
<td>Normal quality content</td>
</tr>
<tr>
<td>2</td>
<td>Good quality content</td>
</tr>
<tr>
<td>3</td>
<td>Good quality and rare content</td>
</tr>
</tbody>
</table>

**Definition 1:** The trust value of any peer starts with zero. i.e., initially, $\forall P_{TV} \forall R_{TV} = 0$.

When a peer with trust value greater than zero is penalized heavily, it takes the same trust value of a newcomer peer. The minimum trust value is not less than a newcomer peer in order to prevent the peers in taking multiple identities.

**Definition 2:** The peer trust values are non-negative numbers. i.e., at any time, the peer trust value is given as $\forall P_{TV} \forall R_{TV} \geq 0$.

**Algorithm 3.1:** First voting procedure for content provider

**Input:** $R_{ID}$, $ON_i$, QC.

**Output:** $V_{TV}$, $R_{IDON_i}$, Status;
// Peers’ reply to the tracker //
01: if (QC = 1) then
02: \( V_{TV} \leftarrow [0.5, 3]; \)
03: \( \text{RIDONiStatus} \leftarrow \text{TRUE}; \)
04: \( \text{NewPID} \leftarrow \text{RID}; \)
05: \( \text{meta-data} \leftarrow \{\text{NewPID}, \text{NewPIP}, \text{Name}, \text{Size}, \text{HC}, \text{GT}\}; \)
06: else
07: \( V_{TV} \leftarrow [-3, -1]; \)
08: \( \text{RIDONiStatus} \leftarrow \text{FALSE}; \)
09: end

In the first voting process, only authenticated peers are allowed. If a newcomer happens to be the provider, peers with trust value above 0.1 are permitted to get involved in the first voting process. Once a peer votes in favor of the content provider, it generates an updated meta-data (as in Algorithm 3.1) and publishes the same in the web server which is visible to the peers as in Figure 3.2. In case the permitted number of tokens is not requested at a stipulated time, the tracker reduces the threshold of the trust value for the benefit of other low trust value peers which is informed through News Feed. When a malicious peer is identified through Algorithm 3.2, it is not punished by blocking or black listing its IP address or peer ID as the peers change their identities very often. Moreover, on some occasions, accidental errors in the shared content can identify a genuine peer as a polluter, which in turn demotivates the peer to share any new contents.

**Algorithm 3.2:** Decision after voting process

**Input:** R\(_{ID}\), ON\(_{i}\), V\(_{TV}\).

**Output:** Average\(_{TV}\), NewP\(_{TV}\), ON\(_{i}\)Status, P\(_{ID}\)Update, R\(_{ID}\)Update;

// Tracker’s Response //
01: repeat
02: \( \text{Average}_{TV} \leftarrow \frac{\text{Total}_{TV}}{\text{Total}_{ID}}; \)
03: \( \text{NewP}_{TV} \leftarrow \text{P}_{TV} + \text{Average}_{TV}; \)
04: if (\( \text{NewP}_{TV} < 0 \)) then
05: \( \text{NewP}_{TV} \leftarrow 0; \)
06: \( \text{ON}_{i}\)Status \( \leftarrow \text{FALSE}; \)
07: end
The decision on the quality of any shared content and overlay partition is made in Algorithm 3.2 and Algorithm 3.3, respectively. The new overlay expansion network numbers are assigned as an extension of the initial overlay network number. The tracker recognizes whether it is the first voting or subsequent voting from the overlay network number. The subsequent voting procedure of the new peers uses a single voting value, which is decided by the tracker in the range 0.1 to 0.5 depending upon the size of the content. As the seeds share the content which they obtained from the provider, the trust value in the second and subsequent voting process is either positive or negative. This makes the initial content provider to get more trust values than the seeds which help in sharing the content.

### 3.3.3 Stabilization Phase

In the UNITY model, P2P overlay network expansion works in three methods after the first voting process as in Algorithm 3.3. The first method of expansion is through each peer of an overlay network after becoming a provider, forming independent and individual overlay networks to decrease the involvement of malicious peers from poisoning the entire overlay as in Figure 3.3.
Figure 3.3 Overlay network expansions as in the proposed model of public P2P content sharing network.
The second method is that all providers in every stage combine to contribute in the sharing process. The problem with this method is that if any peer turns malicious, it is hard to identify. The voting trust values are provided when the total upload becomes greater than the total download, or depending on the value assigned by the tracker for the upload to download ratio. This is to check whether the tokens generated by the peer are serviced properly. The third method is the extension of the second method by combining all the existing providers of the previous stages so that the provider is immersed into the pool of seeds that come for public sharing. The provider contributes until a specified number of seeds exist at a stage and then leaves the overlay network indefinitely with the earned voting trust values. But, the provider earns its trust value only through the first voting process (Algorithm 3.1) for the tokens generated by it. The consequent tokens generated by the provider for further sharing will earn trust value as per the second and subsequent voting process (Algorithm 3.4). In the \( n \)th stage, all existing providers are grouped as per their trust values so that any highly reputed peer attempting to download a particular content can easily access tokens from an equally reputed provider.

**Algorithm 3.3:** Overlay Network Expansion  
**Input:** ON\(_i\), Average\( V_{TV} \).  
**Output:** OE\(_m\), ON\(_i\)Status, P\(_{ID}\)Update, R\(_{ID}\)Update;

```plaintext
// Tracker’s Response //
01: repeat
02: if (Average\( V_{TV} < 1 \)) then
03: \( O_{E_m} \leftarrow 1; \) // First method is selected //
04: for each R\(_{ID}\) in ON\(_i\) do
05: \( R_{ID}Update \leftarrow \{O_{E_m}, NewON_i, G_T, R_{TV}\}; \)
06: end
07: ON\(_i\)Status \leftarrow FALSE;
08: else if (Average\( V_{TV} \geq 1 \) and Average\( V_{TV} < 2 \)) then
09: \( O_{E_m} \leftarrow 2; \) // Second method is selected //
10: for each R\(_{ID}\) in ON\(_i\) do
11: \( R_{ID}Update \leftarrow \{O_{E_m}, ON_i, G_T, R_{TV}\}; \)
12: end
13: \( P_{ID}Update \leftarrow \{P_{ID}, ON_i, Average\( V_{TV}, NewP_{TV}\}; \)
14: for P\(_{ID}\) in ON\(_i\) do
```


\[ ON_{i}Status \leftarrow FALSE; \]

\[ \text{end} \]

\[ \text{else} \]

\[ OE_{m} \leftarrow 3; \quad \text{// Third method is selected //} \]

\[ \text{for each } R_{iD} \text{ in } ON_{i} \text{ do} \]

\[ R_{iD}Update \leftarrow \{OE_{m}, ON_{i}, GT, RTV\}; \]

\[ \text{end} \]

\[ P_{iD}Update \leftarrow \{P_{ID}, ON_{i}, AverageVT, NewPTV\}; \]

\[ \text{end} \]

\[ \text{until } ON_{i}Status = TRUE; \]

### 3.3.4 Decaying Phase

The meta-data viewed by any newcomer or anonymous peer is verified through several overlay expansion stages. The newcomers request for the tokens from the latest stage peers, whereas, a part of the number of allowed tokens are given to the anonymous peers. Apart from the tokens given to anonymous peers, only the tokens serviced to the registered peers are eligible for the voting process. The decaying phase starts when a meta-data is outdated, or the number of tokens allowed by the peers for any meta-data is unused for a longer time. The tracker regulates the peers in leaving the network indefinitely by reducing their number of tokens to service in due course of time. The privilege to leave the network indefinitely is given initially to high trust value peers and later to the medium and low trust value peers. The peers willing to share a particular content further can still be in the overlay network with the option to leave the network at any time. The tracker holds a limited number of tokens from low trust value peers with a time bound.

**Algorithm 3.4:** Second and subsequent voting procedure

**Input:** \( R_{ID}, ON_{i}, QC \).

**Output:** \( V_{TV}, R_{iD}ON_{i}Status; \)

\[ \text{// Peers’ reply to the tracker //} \]

01: if \( QC = 1 \) then

02: \( V_{TV} \leftarrow [0.1, 0.5]; \)
03: \(R_{ID}\text{ONI Status} \leftarrow \text{TRUE};\)
04: \(\text{NewPID} \leftarrow R_{ID};\)
05: \(\text{meta-data} \leftarrow \{\text{NewPID}, \text{NewPIP}, \text{Name}, \text{Size}, H_C, G_T\};\)
06: \textbf{else}
07: \(V_{TV} \leftarrow 0;\)
08: \(R_{ID}\text{ONI Status} \leftarrow \text{FALSE};\)
09: \textbf{end}

// Tracker’s update on the trust database //

10: \textbf{repeat}
11: \(\text{AverageV}_{TV} \leftarrow \text{TotalV}_{TV} / \text{TotalR}_{ID};\)
12: \(\text{NewP}_{TV} \leftarrow P_{TV} + \text{AverageV}_{TV};\)
13: \(\text{Trust-Database} \leftarrow \{P_{ID}, P_{TV}, ON, G_T, \text{averageV}_{TV}, \text{NewP}_{TV}\};\)
14: \(P_{ID}\text{Update} \leftarrow \{P_{ID}, ON, \text{averageV}_{TV}, \text{NewP}_{TV}\};\)
15: \textbf{until} \(G_T = \text{TotalR}_{ID};\)

3.4 MATHEMATICAL ANALYSIS

The P2P networks can be considered as homogeneous or heterogeneous depending upon the peers’ resource diversity. In a homogeneous P2P network, the peers are considered to have identical resource capability including their bandwidth, whereas in heterogeneous P2P networks, the peers have varying resource capacities including their contribution level and unpredictable churn rate. The mathematical analysis considers three approaches for P2P overlay network expansion, namely, (a) the \(n^{th}\) stage seeds only, (b) the \(n^{th}\) stage seeds along with its immediate previous stage seeds, and (c) combined number of seeds from all stages. The notations used in the mathematical analysis are shown in Table 3.3.

In the analysis, the above mentioned three approaches are considered in three main cases and two sub-cases as per the practical situation. The first case is with every seed in the network cooperating for network expansion. The second case considers a specified number of non-cooperating peers in each overlay partition of the considered stage, while the last main case considers a variable number of non-cooperating peers in each overlay partition of the considered stage. The second and
third main cases mimic the real-world scenario. The two sub-cases are with a specified and variable number of non-cooperating peers in each stage as shown in Figure 3.3.

Table 3.3 Notations and their interpretation in mathematical analysis of public P2P network

<table>
<thead>
<tr>
<th>Notations</th>
<th>Homogeneous Network</th>
<th>Heterogeneous Network</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>n</td>
<td>Overlay stage number</td>
<td></td>
</tr>
<tr>
<td>Na1</td>
<td>N_b1</td>
<td>Total number of seeds in the $n^{th}$ stage only</td>
<td></td>
</tr>
<tr>
<td>Na2</td>
<td>N_b2</td>
<td>Total number of seeds in the $n^{th}$ stage along with its immediate previous stage seeds</td>
<td></td>
</tr>
<tr>
<td>Na3</td>
<td>N_b3</td>
<td>Total combined number of seeds in $n$ stages</td>
<td></td>
</tr>
<tr>
<td>l</td>
<td>A_0</td>
<td>Initial number of seeds</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>$\alpha$, $\beta$</td>
<td>Number of peers added in each stage</td>
<td></td>
</tr>
<tr>
<td>u</td>
<td>u</td>
<td>Number of non-cooperating peers</td>
<td></td>
</tr>
<tr>
<td>u_i</td>
<td>u_i</td>
<td>Number of non-cooperating peers varying in each stage of the overlay</td>
<td></td>
</tr>
</tbody>
</table>

3.4.1 Homogeneous P2P Network

In the homogeneous network, three cases are considered for analysis. First case is when each seed in the network cooperates for network expansion. The second case considers a specified number of non-cooperating peers in each overlay stage, whereas the third case is an extension of the second case with variable number of non-cooperating peers in each stage.

3.4.1.1 Cooperating peers in overlay partition

Mathematically, the homogenous P2P network of ‘S’ peers constantly bootstrapped in the network at each stage by every seed, with the assumption that all
seeds cooperate for overlay expansion, is shown in Table 3.4. In the initial stage, a
single provider starts the sharing process with ‘S’ peers, which leads to ‘S’ number
of seeds available at the end of the first stage. When the network expands through
each stage, multiples of ‘S’ number of seeds are attained in every stage leading to
Equation (3.1) in n\textsuperscript{th} stage, given by

\[ N_{a1} = S^n. \]  

(3.1)

In order to improve seed availability, the immediate previous stage seeds are
retained in the network with the current stage of seeds, until a new stage is formed.
As per the second approach, the number of seeds in the n\textsuperscript{th} stage is combined with
the seeds of its immediate previous stage, (n-1)\textsuperscript{th} stage, given by

\[ N_{a2} = S^{n-1} + S^n. \]  

(3.2)

Table 3.4 Homogeneous Network expansion with cooperating peers in overlay
partition

<table>
<thead>
<tr>
<th>Overlay Stage (n)</th>
<th>Mathematical Representation</th>
<th>Mathematical Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1 * S</td>
<td>S</td>
</tr>
<tr>
<td>2</td>
<td>1 * S * S</td>
<td>S^2</td>
</tr>
<tr>
<td>3</td>
<td>1 * S * S * S</td>
<td>S^3</td>
</tr>
<tr>
<td>4</td>
<td>1 * S * S * S * S</td>
<td>S^4</td>
</tr>
<tr>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
</tr>
<tr>
<td>n</td>
<td>1 * S * S * S * S * \ldots * S</td>
<td>S^n</td>
</tr>
</tbody>
</table>
The third approach considers the contribution from all seeds since the initial stage as the theoretical ideal case. It is obtained by the summation of the number of seeds from the initial stage till the $n^{th}$ stage, given by

$$N_{a3} = \sum_{i=0}^{n} S^i. \quad (3.3)$$

### 3.4.1.2 Constant non-cooperating peers in each stage

In a homogeneous network, non-cooperating peers are formed due to various reasons, like their malicious behavior, free-riding nature or unavoidable circumstances. Those non-cooperating peers are excluded from the overlay network at every stage for monitoring the exact number of seeds and their contributions as specified in the second case, as shown in Table 3.5. From Table 3.5, the number of seeds in the $n^{th}$ stage after removing ‘u’ non-cooperating peers from each stage is given by

$$N_{a1} = S^n - u \left[ S^{n-1} - S^{n-2} - S^{n-3} - \cdots - S^2 - S^1 - S^0 \right]. \quad (3.4)$$

Equation (3.4) can be rewritten in the general form as

$$N_{a1} = S^n - u \sum_{i=0}^{n-1} S^i. \quad (3.5)$$
Table 3.5 Homogeneous Network expansion with constant non-cooperating peers

<table>
<thead>
<tr>
<th>Overlay Stage ( n )</th>
<th>Mathematical Representation</th>
<th>Mathematical Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( 1 )</td>
<td>( 1 )</td>
</tr>
<tr>
<td>1</td>
<td>( (1 \times S) - u )</td>
<td>( S - u )</td>
</tr>
<tr>
<td>2</td>
<td>( S[(1 \times S) - u] - u )</td>
<td>( S^2 - Su - u )</td>
</tr>
<tr>
<td>3</td>
<td>( S[S[(1 \times S) - u] - u] - u )</td>
<td>( S^3 - S^2u - Su - u )</td>
</tr>
<tr>
<td>4</td>
<td>( S[S[S[(1 \times S) - u] - u] - u] - u )</td>
<td>( S^4 - S^3u - S^2u - Su - u )</td>
</tr>
<tr>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
</tr>
<tr>
<td>( n )</td>
<td>( S \cdots [S[S[(1 \times S) - u] - u] - u] - u \cdots - u )</td>
<td>( S^n - S^{n-1}u - S^{n-2}u - \cdots - S^{n-n}u )</td>
</tr>
</tbody>
</table>
The number of seeds through second and third approach is obtained through the process similar to Equations (3.2) and (3.3), which are given in Equations (3.6) and (3.7), respectively, as

\[ N_{a2} = \sum_{j=n-1}^{n} \left( S^j - u \sum_{i=0}^{j-1} S^i \right), \]  

(3.6)

\[ N_{a3} = 1 + \sum_{j=1}^{n} \left( S^j - u \sum_{i=0}^{j-1} S^i \right). \]  

(3.7)

### 3.4.1.3 Variable non-cooperating peers in each stage

In the third case of the homogeneous network, when the number of non-cooperating peers, ‘\( u \)’, in each stage is a variable, the value varies from 0 as the lower limit to ‘\( S \)’ as the upper limit. The overlay network expansion through each stage is shown in Table 3.6. For a constant ‘\( S \)’ peers joining the network and variable (\( u \)) non-cooperating peers removed from the network, the total number of seeds in Equations (3.5), (3.6) and (3.7) are rewritten as Equations (3.8), (3.9) and (3.10), respectively, while considering ‘\( u \)’ non-cooperating peers at each stage of the P2P network expansion, as given by

\[ N_{a1} = S^n - \sum_{i=0}^{n-1} S^i u_{n-i}, \]  

(3.8)

\[ N_{a2} = \sum_{j=n-1}^{n} \left[ S^j - \sum_{i=0}^{j-1} S^i u_{j-i} \right], \]  

(3.9)

and

\[ N_{a3} = 1 + \sum_{j=1}^{n} \left[ S^j - \sum_{i=0}^{j-1} S^i u_{j-i} \right]. \]  

(3.10)
Table 3.6 Homogeneous network expansion with variable non-cooperating peers

<table>
<thead>
<tr>
<th>Overlay Stage (n)</th>
<th>Mathematical Representation</th>
<th>Mathematical Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>$(1 \times S) - u_1$</td>
<td>$S - u_1$</td>
</tr>
<tr>
<td>2</td>
<td>$S[(1 \times S) - u_1] - u_2$</td>
<td>$S^2 - Su_1 - u_2$</td>
</tr>
<tr>
<td>3</td>
<td>$S[S[(1 \times S) - u_1] - u_2] - u_3$</td>
<td>$S^3 - S^2u_1 - Su_2 - u_3$</td>
</tr>
<tr>
<td>4</td>
<td>$S[S[S[(1 \times S) - u_1] - u_2] - u_3] - u_4$</td>
<td>$S^4 - S^3u_1 - S^2u_2 - Su_3 - u_4$</td>
</tr>
<tr>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
</tr>
<tr>
<td>$n$</td>
<td>$S \cdots [S[S[(1 \times S) - u_1] - u_2] - u_3] \cdots - u_n$</td>
<td>$S^n - S^{n-1}u_i - S^{n-2}u_{i+1} - \cdots - S^nu_n$</td>
</tr>
</tbody>
</table>
3.4.2 Heterogeneous P2P Network

Heterogeneous P2P network considers the unpredictable variation of the churn rate in each stage. In this network, three main cases and two sub-cases are considered for analysis. Similar to homogeneous network analysis, first case is with every seed in the network cooperating for network expansion. Second case considers a specified number of non-cooperating peers in each overlay partition of the considered stage, while the last main case considers a variable number of non-cooperating peers in each overlay partition of the considered stage. The second and third main cases mimic the real-world scenario. The two sub-cases are similar to the second and third cases of the homogeneous network with a specified and variable number of non-cooperating peers in each stage. When the initial number of providers is $A_0$, the total number of seeds in each stage with peers cooperating and non-cooperating with each other for network expansion is shown as the following cases.

3.4.2.1 Cooperating peers in overlay partition

When the initial number of providers is $A_0$, the total number of seeds in each stage through peers cooperating with each other for network expansion is shown in Table 3.7. Consider a provider ($\alpha$) bootstrapping a certain number of peers ($\beta$) into the overlay network. Then, the number of seeds added to the network at the end of the stage is the product of $\alpha$ and $\beta$.

When the new seeds bootstrap with different number of peers depending upon their capacities, the number of seeds in the network at the next stage is the summation of the product of $\alpha$ and $\beta$ as shown in Equation (3.11). Equation (3.11) is obtained for the $n^{th}$ stage alone, excluding the seeds from the previous stages. In order to improve seed availability, the immediate previous stage seeds are retained in the network with the current stage of seeds, until a new stage is formed. As per the
second approach, the number of seeds in the \(n^{th}\) stage is combined with the seeds of its immediate previous stage, \((n-1)^{th}\) stage, and is given by

\[
N_{b1} = \sum_{i=1}^{A_{n-1}} \alpha_{ni} \beta_{ni},
\]

and

\[
N_{b2} = \sum_{j=n-1}^{n} \left\{ \sum_{i=1}^{A_{j-1}} \alpha_{ji} \beta_{ji} \right\}.
\]

Table 3.7 Heterogeneous Network expansion with cooperating peers

<table>
<thead>
<tr>
<th>Overlay Stage (n)</th>
<th>Mathematical Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(A_0)</td>
</tr>
<tr>
<td>1</td>
<td>(A_1 = \sum_{i=1}^{A_0} \alpha_{1i} \beta_{1i})</td>
</tr>
<tr>
<td>2</td>
<td>(A_2 = \sum_{i=1}^{A_1} \alpha_{2i} \beta_{2i})</td>
</tr>
<tr>
<td>3</td>
<td>(A_3 = \sum_{i=1}^{A_2} \alpha_{3i} \beta_{3i})</td>
</tr>
<tr>
<td>\vdots</td>
<td>\vdots</td>
</tr>
<tr>
<td>(n)</td>
<td>(A_n = \sum_{i=1}^{A_{n-1}} \alpha_{ni} \beta_{ni})</td>
</tr>
</tbody>
</table>

The third approach considers the contribution from all seeds from the initial stage as the theoretical ideal case. The total number of seeds from all stages is
calculated by the summation of seeds from each stage as in Equation (3.13), and its mathematical equivalent form from Equation (3.14) is given by

\[ A_0 + A_1 + A_2 + A_3 + A_4 + \cdots + A_n = \sum_{i=0}^{n} A_i, \quad (3.13) \]

\[ N_{b3} = A_0 + \sum_{j=1}^{n} \left[ \sum_{i=1}^{A_{j-1}} \alpha_{ji} \beta_{ji} \right] A_j. \quad (3.14) \]

### 3.4.2.2 Constant non-cooperating peers in overlay partition

In a heterogeneous network, non-cooperating peers exist due to various reasons, like their malicious behavior, free-riding nature or unavoidable circumstances. Those non-cooperating peers are excluded from the overlay network at each overlay partition for monitoring the exact number of seeds as shown in Table 3.8. The constant number of non-cooperating peers is removed from each overlay partition of any considered stage as shown in Figure 3.4. The number of seeds in Equations (3.11), (3.12) and (3.14) are rewritten as \( N_1, N_2, N_3 \), while considering ‘u’ non-cooperating peers in each overlay partition as Equations (3.15), (3.16) and (3.17), respectively, given by

\[ N_{b1} = \sum_{i=1}^{A_{n-1}} \alpha_{ni} \beta_{ni} - u, \quad (3.15) \]

\[ N_{b2} = \sum_{j=n-1}^{n} \left[ \sum_{i=1}^{A_{j-1}} \left( \alpha_{ji} \beta_{ji} \right) - u \right] A_j, \quad (3.16) \]
and

$$N_{b3} = A_0 + \sum_{j=1}^{n} \left[ \sum_{i=1}^{A_j} (\alpha_{ji} \beta_{ji}) - u \right].$$

(3.17)

Figure 3.4 Overlay network expansions with constant non-cooperating peers in an overlay partition.
Table 3.8 Heterogeneous network expansion with constant non-cooperating peers

<table>
<thead>
<tr>
<th>Overlay Stage (n)</th>
<th>Mathematical Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$A_0$</td>
</tr>
<tr>
<td>1</td>
<td>$A_1 = \sum_{i=1}^{A_0} \alpha_{1i} \beta_{1i} - u$</td>
</tr>
<tr>
<td>2</td>
<td>$A_2 = \sum_{i=1}^{A_2} \alpha_{2i} \beta_{2i} - u$</td>
</tr>
<tr>
<td>3</td>
<td>$A_3 = \sum_{i=1}^{A_3} \alpha_{3i} \beta_{3i} - u$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$n$</td>
<td>$A_n = \sum_{i=1}^{d_{n-1}} \alpha_{ni} \beta_{ni} - u$</td>
</tr>
</tbody>
</table>

3.4.2.3 Variable non-cooperating peers in overlay partition

When the number of non-cooperating peers ($u$) as considered in the previous case in each overlay partition is a variable, the value of $u$ varies from 0 to $\beta$ as shown in Figure 3.5. This case mimics the practical situation of considering mobile devices along with high performance computers in the content sharing network and its network expansion as shown in Table 3.9.

Equations (3.15), (3.16) and (3.17) are rewritten for $N_1$, $N_2$, $N_3$ as Equations (3.18), (3.19) and (3.20), respectively. Considering $u_i$ non-cooperating peers at each stage ‘$i$’ from the previous model, the number of seeds is given by
\[ N_{b1} = \sum_{i=1}^{A_{n-1}} \alpha_{ni} \beta_{ni} - u_{ni}, \] (3.18)

\[ N_{b2} = \sum_{j=n-1}^{n} \sum_{i=1}^{A_{j-1}} \left( \alpha_{ji} \beta_{ji} \right) - u_{ji}, \] (3.19)

and
\[ N_{b3} = A_0 + \sum_{j=1}^{n} \sum_{i=1}^{A_{j-1}} \left( \alpha_{ji} \beta_{ji} \right) - u_{ji}. \] (3.20)

**Table 3.9** Heterogeneous Network expansion with variable non-cooperating peers

<table>
<thead>
<tr>
<th>Overlay Stage (n)</th>
<th>Mathematical Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( A_0 )</td>
</tr>
<tr>
<td>1</td>
<td>( A_1 = \sum_{i=1}^{A_{0}} \alpha_{1i} \beta_{1i} - u_{1i} )</td>
</tr>
<tr>
<td>2</td>
<td>( A_2 = \sum_{i=1}^{A_{1}} \alpha_{2i} \beta_{2i} - u_{2i} )</td>
</tr>
<tr>
<td>3</td>
<td>( A_3 = \sum_{i=1}^{A_{2}} \alpha_{3i} \beta_{3i} - u_{3i} )</td>
</tr>
<tr>
<td>( \vdots )</td>
<td>( \vdots )</td>
</tr>
<tr>
<td>( n )</td>
<td>( A_n = \sum_{i=1}^{A_{n-1}} \alpha_{ni} \beta_{ni} - u_{ni} )</td>
</tr>
</tbody>
</table>
Figure 3.5 Overlay network expansions with variable non-cooperating peers in an overlay partition.

3.4.2.4 Constant non-cooperating peers in each stage

In heterogeneous networks, the multiplicative factor varies usually due to different computational capacities. When high bandwidth seeds are less and requestors in the queue for tokens are huge, a constant multiplicative factor is
considered for all seeds in a particular stage along with a constant number of non-cooperating peers as shown in Figure 3.6. This is a subcase of case 2. The number of seeds in the $n^{th}$ stage with ‘u’ non-cooperating peers in each stage is given in Table 3.10 as

$$N_{b1} = \left[ \prod_{i=1}^{n} \beta_i \right] - \left[ \sum_{j=1}^{n-1} u \left( \prod_{i=j+1}^{n} \beta_i \right) \right] - u.$$  (3.21)

**Figure 3.6** Overlay network expansions with constant non-cooperating peers in each stage.
Table 3.10 Heterogeneous network expansion with constant non-cooperating peers in each stage

<table>
<thead>
<tr>
<th>Overlay Stage (n)</th>
<th>Mathematical Representation</th>
<th>Mathematical Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>$\beta_1 - u$</td>
<td>$\beta_1 - u$</td>
</tr>
<tr>
<td>2</td>
<td>$\beta_2 (\beta_1 - u) - u$</td>
<td>$\beta_2 \beta_1 - \beta_2 u - u$</td>
</tr>
<tr>
<td>3</td>
<td>$\beta_3 [\beta_2 (\beta_1 - u) - u] - u$</td>
<td>$\beta_3 \beta_2 \beta_1 - \beta_3 \beta_2 u - \beta_3 u - u$</td>
</tr>
<tr>
<td>4</td>
<td>$\beta_4 [\beta_3 [\beta_2 (\beta_1 - u) - u] - u] - u$</td>
<td>$\beta_4 \beta_3 \beta_2 \beta_1 - \beta_4 \beta_3 \beta_2 u - \beta_4 \beta_2 u - \beta_4 u - u$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$n$</td>
<td>$\beta_n \cdots [\beta_3 [\beta_2 (\beta_1 - u) - u] - u] - u$</td>
<td>$\beta_n \beta_{n-1} \beta_{n-2} \cdots \beta_1 - \beta_n \beta_{n-1} \cdots \beta_2 u - \beta_n \beta_{n-1} \cdots \beta_2 u - \cdots - \beta_n u - u$</td>
</tr>
</tbody>
</table>
Equation (3.22) is obtained by considering the $n^{th}$ and $(n-1)^{th}$ stages, whereas Equation (3.23) is from all ‘n’ stages combined, and is given by

$$N_{b2} = \sum_{k=n-1}^{n} \left( \prod_{i=1}^{k} \beta_i \right) - \left( \sum_{j=1}^{k-1} u \left( \prod_{l=j+1}^{k} \beta_l \right) \right) - u$$

(3.22)

and

$$N_{b3} = A_0 + \sum_{k=1}^{n} \left[ \left( \prod_{i=1}^{k} \beta_i \right) - \left( \sum_{j=1}^{k-1} u \left( \prod_{l=j+1}^{k} \beta_l \right) \right) - u \right]$$

(3.23)

### 3.4.2.5 Variable non-cooperating peers in each stage

For a constant multiplicative factor for all seeds in a particular stage and a variable ($u$) non-cooperating peers at each stage ‘$i$’, the network expansion works as shown in Table 3.11. This case is similar to the variable non-cooperating peers in an overlay partition due to the variable nature of the non-cooperating peers.

Equations (3.21), (3.22) and (3.23) for $N_1$, $N_2$, $N_3$ are rewritten with variable number of non-cooperating peers as Equations (3.24), (3.25) and (3.26), respectively, from Table 3.11, as given by

$$N_{b1} = \left[ \prod_{i=1}^{n} \beta_i \right] - \left[ \sum_{j=1}^{n-1} u_j \left( \prod_{l=j+1}^{n} \beta_l \right) \right] - u_n$$

(3.24)
\[ N_{b2} = \sum_{k=n-1}^{n} \left[ \left( \prod_{i=1}^{k} \beta_i \right) - \left( \sum_{j=1}^{k-1} u_j \left( \prod_{l=j+1}^{k} \beta_l \right) \right) - u_k \right] \] \hspace{1cm} (3.25)

\[ N_{b3} = A_0 + \sum_{k=1}^{n} \left[ \left( \prod_{i=1}^{k} \beta_i \right) - \left( \sum_{j=1}^{k-1} u_j \left( \prod_{l=j+1}^{k} \beta_l \right) \right) - u_k \right] \] \hspace{1cm} (3.26)

Figure 3.7 Overlay network expansions with variable non-cooperating peers in each stage.
Table 3.11 Heterogeneous Network expansion with variable non-cooperating peers in each stage

<table>
<thead>
<tr>
<th>Overlay Stage (n)</th>
<th>Mathematical Representation</th>
<th>Mathematical Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>$(1 \times \beta_1) - u_1$</td>
<td>$\beta_1 - u_1$</td>
</tr>
<tr>
<td>2</td>
<td>$\beta_2(\beta_1 - u_1) - u_2$</td>
<td>$\beta_2\beta_1 - \beta_2u_1 - u_2$</td>
</tr>
<tr>
<td>3</td>
<td>$\beta_3[\beta_2(\beta_1 - u_1) - u_2] - u_3$</td>
<td>$\beta_3\beta_2\beta_1 - \beta_3\beta_2u_1 - \beta_3u_2 - u_3$</td>
</tr>
<tr>
<td>4</td>
<td>$\beta_4[\beta_3[\beta_2(\beta_1 - u_1) - u_2] - u_3] - u_4$</td>
<td>$\beta_4\beta_3\beta_2\beta_1 - \beta_4\beta_3\beta_2u_1 - \beta_4\beta_3u_2 - \beta_4u_3 - u_4$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$n$</td>
<td>$\beta_n[\beta_{n-2}(\beta_1 - u_1) - u_2] - \cdots - u_n$</td>
<td>$\beta_n\beta_{n-2}\beta_1 - \beta_n\beta_{n-1} - \beta_2u_1 - \beta_n\beta_{n-1} - \cdots - \beta_nu_{n-1} - u_n$</td>
</tr>
</tbody>
</table>
3.5 NUMERICAL RESULTS AND DISCUSSIONS

The numerical calculations are done using the parametric values given in Table 3.12. The comparison between three homogeneous cases for the number of seeds, $N_{a1}$ and $N_{a3}$ is given in Figure 3.8. The differences in the number of seeds between $N_{a1}$ and $N_{a3}$ are not with much variation. This less variation allows the seeds from the previous stages to leave the network to save their bandwidth without affecting overlay expansion.

Table 3.12 Parametric values for homogeneous network expansion through a mathematical model

<table>
<thead>
<tr>
<th>Homogeneous Network</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>All peers cooperating</td>
<td>Degree or Number of stages</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Multiplicative Factor</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Number of peers removed</td>
<td>0</td>
</tr>
<tr>
<td>Constant peers non-cooperating</td>
<td>Degree or Number of stages</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Multiplicative Factor</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Number of peers removed</td>
<td>1</td>
</tr>
<tr>
<td>Variable peers non-cooperating</td>
<td>Degree or Number of stages</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Multiplicative Factor</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Number of peers removed</td>
<td>1 - 5</td>
</tr>
</tbody>
</table>
Figure 3.8 Comparison of homogeneous network cases with total number of seeds given by (a) $N_{a1}$ and (b) $N_{a3}$.

Similarly, the comparison between three heterogeneous main cases for the number of seeds, $N_{b1}$ and $N_{b3}$, shows only minor variations as observed in Figure 3.9(a), which exhibits that $N_{b1}$ is sufficient for consistent network expansion. The sub-cases in a heterogeneous network for the number of seeds, $N_{b1}$ and $N_{b3}$, show variations as observed in Figure 3.9(b) due to the unpredictable nature of the churn rate that is considered. These sub-cases imitate the practical situations.
Figure 3.9 Comparison of heterogeneous network cases with total number of seeds given by $N_{b1}$ and $N_{b3}$ for (a) each partition overlay and (b) each stage.
3.6 SIMULATION SETUP

PeerSim [124] is a simulation environment for P2P protocols which offers extreme scalability and support for dynamism. It supports two different simulation models: the cycle-based model and the event-based one. PeerSim is highly customizable by adding different extendable components which can be set up through a simple configuration mechanism. Public P2P content sharing network is simulated in PeerSim up to the scalability of 10,000 nodes in event driven mode.

The simulation uses three different scenarios, namely, existing BitTorrent model, homogenous UNITY model and heterogeneous UNITY model. In BitTorrent model, each node is configured as in a real-world environment with the existing BitTorrent protocol setup by allowing uncontrolled churn rate. Choking algorithm using tit-for-tat mechanism, rarest first, endgame mode, tracker down time, flash crowd and peer down time are considered for all the three scenarios.

In simulation, the content size is 100 MB, which is split into pieces of 256 KB each, and further each piece is split into 16 blocks, each with block size 16 KB. If the download of a piece crashes, the piece has to be entirely re-downloaded. Splitting the file into pieces enhances the precision in the estimate of download time and allows a fast download of the block from peers with low bandwidth. Different blocks belonging to the same piece can be downloaded from different peers. After the download of a piece has been completed, the peer must send a message to its neighbors declaring that it has that particular piece of file.

3.6.1 Simulation Parameters

The BitTorrent protocol works through the following allowed protocol parameters. The proposed UNITY model simulation uses the same parameters with values assigned as per the algorithms are shown in Table 3.13.
Table 3.13 Simulation parametric values for UNITY P2P network

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>File Size</td>
<td>100 MB</td>
</tr>
<tr>
<td>2.</td>
<td>Network Size</td>
<td>10</td>
</tr>
<tr>
<td>3.</td>
<td>Maximum Swarm Size</td>
<td>80</td>
</tr>
<tr>
<td>4.</td>
<td>Peer Set Size</td>
<td>50</td>
</tr>
<tr>
<td>5.</td>
<td>Duplicated Requests</td>
<td>1</td>
</tr>
<tr>
<td>6.</td>
<td>Maximum Growth</td>
<td>30</td>
</tr>
<tr>
<td>7.</td>
<td>Newer Distribution</td>
<td>9</td>
</tr>
<tr>
<td>8.</td>
<td>Seeder Distribution</td>
<td>1</td>
</tr>
<tr>
<td>9.</td>
<td>Tracker Down</td>
<td>1</td>
</tr>
<tr>
<td>10.</td>
<td>Minimum Network Size</td>
<td>5</td>
</tr>
<tr>
<td>11.</td>
<td>Step</td>
<td>100,000 Seconds</td>
</tr>
<tr>
<td>12.</td>
<td>Add</td>
<td>2 to 50</td>
</tr>
<tr>
<td>13.</td>
<td>Remove</td>
<td>0 to 30</td>
</tr>
<tr>
<td>14.</td>
<td>Delay</td>
<td>10 to 400 ms</td>
</tr>
</tbody>
</table>

(a) **File Size:** It is the size (in Megabytes) of the content or files shared by the nodes of the P2P content sharing network.

(b) **Network Size:** This defines the number of nodes in the network at the starting time of simulation. This also specifies the number of nodes in the network, including the tracker.

(c) **Maximum Swarm Size:** It is the maximum size of the swarm for each node; this value specifies the dimension of the cache (for each node) in which the neighbor node details are stored.

(d) **Peer Set Size:** It is the number of nodes contained in the PEERSET message; this kind of message is sent from the tracker to the new nodes in the network. The message contains a set of nodes to be communicated to download a particular content.

(e) **Duplicated Requests:** It is the number of duplicated REQUEST messages that can be sent after an UNCHOKE for the same block to different peers.
(f) **Transport**: This specifies the transport layer used for simulation. All tests have been made using the Uniform Random Transport layer which is provided with the official distribution of PeerSim.

(g) **Maximum Growth**: This value defines how much the network can grow when the control mechanisms are used. This value also defines the incremental factor of the size of the network.

(h) **Network Initializer**: It performs the initialization of the tracker (updating its neighbor list) and initializes all nodes participating in the network.

(i) **Node Initializer**: The initialization of a single node (with the exception of the tracker) is performed through the setting of the bandwidth for the peer and choosing the shared file and its percentage that has been downloaded, according to the configuration parameters.

(j) **Newer Distribution**: Specifies the distribution of the shared file for nodes joining the network. This value refers to the percentage of nodes with zero downloaded pieces.

(k) **Seeder Distribution**: Specifies the percentage of seeders in the network. For example, the value 10 means that 10% of the nodes in the network are seeders.

(l) **Tracker**: It periodically keeps track of the status of the shared file (number of pieces downloaded per nodes), the number of pieces currently uploaded/downloaded by each node, the status of the node (leech or seed) and some other values like the average number of neighbors per peer.

(m) **Minimum Size**: Specifies the minimum size of the network (in number of nodes) when the control mechanism is used; the network size can decrease until the minimum size.

(n) **Tracker Down**: Specifies if the tracker can disappear from the network. Note that if the tracker is alive, new nodes can join the network and participate in file sharing. Otherwise, only the existing nodes can continue sharing phase and no new nodes can be added to the network.

(o) **Step**: Defines the frequency of execution of control mechanisms.

(p) **Add**: Defines the number of new nodes to be added to the network.

(q) **Remove**: Defines the number of nodes to be removed from the network.
(r) **Choke:** Choking is a temporary refusal of uploading pieces to a peer. The main idea is to provide a correspondence between upload and download traffic, achieving connections in both the directions.

### 3.6.2 Scenarios and Simulation Results

UNITY-Model-1 is the scenario with all peers cooperating for network expansion, UNITY-Model-2 is considered with 'u' non-cooperating peers in each stage, and UNITY-Model-3 is for ‘ui’ non-cooperating peers in each stage of the overlay network. In a homogenous UNITY model, each peer is configured with same bandwidth and delay. In a heterogeneous UNITY model, each peer is configured with different bandwidth and delay as in real-world networks comprising high performance peers and mobile peers. The bandwidth varies from 256 kbps to 16 Mbps and delay varying from 10 ms to 450 ms. BitTorrent is one of the most successful P2P file sharing network used by majority of internet users. Hence, BitTorrent simulation is used as a benchmark for comparison of performance measures with the proposed model. In BitTorrent protocol, tit-for-tat strategy is adopted based on bandwidth sharing. This is an incentive mechanism to cooperate with the peers that share more bandwidth and choking other peers for less uploads.

#### 3.6.2.1 Homogeneous Network

Comparison of BitTorrent simulation with three homogeneous and heterogeneous network models for network size is shown in Figure 3.10. At the initial time, BitTorrent network size exceeds the network size of homogeneous and heterogeneous network models due to controlled scalability of the network. In due course of time, the network size of the proposed UNITY models exceed BitTorrent simulation providing large number of peers available for content sharing as shown in Figure 3.11. In homogeneous and heterogeneous network models, the effects of flash crowd are simulated and compared with the BitTorrent protocol for which results are shown in Figure 3.12.
Figure 3.10 Comparison of BitTorrent and UNITY homogeneous network models
(a) number of seeds and (b) number of peers.
Figure 3.11 Comparison of BitTorrent and UNITY homogeneous network models
(a) network size at initial time and (b) network size in prolonged time.
The theoretical results are compared with simulation results of the homogeneous network models and heterogeneous network models for the number of seeds in the combined stages given by Figure 3.13.

**Figure 3.12** Comparison of the effect of flash crowd in BitTorrent and UNITY models for homogeneous network.

**Figure 3.13** Comparison of simulation results with theoretical results for homogenous of $N_{a3}$. 
3.6.2.2 Heterogeneous Network

In order to compare with the existing BitTorrent-like networks, one main and two sub-cases of a heterogeneous network model are considered, which suit well with real-time scenarios. UNITY-Model-1 is the scenario with all peers cooperating for network expansion, UNITY-Model-2 is considered with ‘u’ non-cooperating peers in each stage and UNITY-Model-3 is for ‘ui’ non-cooperating peers in each stage of the overlay network.

In a heterogeneous UNITY model, each peer is configured with different bandwidth and delay as in real-world networks comprising high performance peers and mobile peers. The bandwidth varies from 256 kbps to 16 Mbps and delay varies from 10 ms to 450 ms. Simulated results of heterogeneous network models for the number of seeds as in Equations (3.14), (3.23), (3.26) are shown in Figure 3.14 (a). Figure 3.14 (b) shows a comparison of BitTorrent simulation along with heterogeneous network models for the total number of peers in the network. Comparison of BitTorrent simulation with three heterogeneous network models for network size is shown in Figure 3.15. Initially, BitTorrent network size exceeds the network size of UNITY models due to controlled scalability of the network. In due course of time, the network size of the proposed UNITY model exceeds BitTorrent simulation provided that a large number of peers are available for content sharing.

In heterogeneous network models, the effect of flash crowd is simulated and compared with BitTorrent for which the results are shown in Figure 3.16. The theoretical results are compared with the simulation results of UNITY models for the number of seeds in the combined stages shown in Figure 3.17. When an infected or polluted content is shared in the P2P network, the network size is uncontrollable in BitTorrent-like systems, whereas the UNITY model utilizing controlled scalability efficiently prevents bandwidth wastage and spreading infected content, through the voting process shown in Figure 3.18.
Figure 3.14 Comparison of BitTorrent and UNITY heterogeneous network models
(a) number of seeds (b) number of peers.
Figure 3.15 Comparison of BitTorrent and UNITY heterogeneous network models
(a) network size at initial time (b) network size in prolonged time.
Figure 3.16 Comparison of the effect of initial flash crowd in BitTorrent and UNITY heterogeneous network models.

Figure 3.17 Comparison of simulation results with theoretical results for heterogeneous networks of $N_{b3}$.
Figure 3.18 Comparison of the effect of network size in BitTorrent and UNITY heterogeneous network model in the presence of polluted or infected content.

3.7 DISCUSSIONS

Simulated results of BitTorrent network, homogeneous and heterogeneous network models are compared and discussed as shown below:

(a) Peer Availability: In the simulated homogeneous and heterogeneous network models as shown in Figure 3.10(a) and Figure 3.14(a), it is observed that the total number of seeds in UNITY-models show significant increase in peer (seed) availability compared to the seeds in the simulated BitTorrent model. However, during overlay network expansion, the number of seeds from initial stages is comparatively fewer in the pool of seeds. When the total number of seeds in any stage is much higher (> 10 times) than the previous stage, the initial stage seeds are allowed to exit the P2P network indefinitely with their earned trust values if they wish as per the proposed Algorithm 3.2 and Algorithm 3.4. The initial provider has the option to leave the network after the first voting with its earned trust value. It is mandatory for any seed to
contribute to the upload equivalent of the tokens generated by them. It is also observed that allowing the seeds to exit the P2P network still maintains higher peer availability compared to the simulated BitTorrent model as shown in UNITY-Model-2 and UNITY-Model-3 of Figure 3.10(a) and Figure 3.14(a). As the seeds are penalized for their non-cooperating behavior, the ultimate number of available seeds for contribution at later stages is always higher than their previous stages of overlay, whereas in BitTorrent, unpredictable churn rate and peer availability issues exist at all time.

(b) **Controlled Scalability:** Though BitTorrent-like systems are known for their supported high scalability, problems like free-riders, flash crowd are much prone to heavy network traffic as shown in Figure 3.12 and Figure 3.16. Simulated UNITY model for homogenous and heterogeneous network with the implementation of controlled scalability proves robust against flash crowd. Due to limited number of tokens generated, considering the provider’s resource capacity, any sudden increase in the churn rate is prohibited. Simulation results show that flash crowd leading to DoS attack against the initial provider is prevented fully compared to the BitTorrent-like networks as shown in Figure 3.12 and Figure 3.16. It is also observed that controlled scalability utilizes computational capability and bandwidth of the provider efficiently by allowing limited and reliable peers to join the content sharing network. Controlled scalability outperforms BitTorrent system’s scalability of P2P overlay network in later stages of overlay network as shown in Figure 3.11 and Figure 3.15. The total number of peers, seeds, and network size after certain stages in both homogeneous and heterogeneous environment are more than BitTorrent network as shown in Figures 3.10, 3.11, 3.14 and 3.15.

(c) **Content Availability:** In BitTorrent-like networks, any seed having a particular content may not intend to share it for saving bandwidth. In our proposed UNITY model, it is necessary for each peer to cooperate in uploading the content as the trust value of each peer is calculated through their amount of contribution in content sharing. It is inferred that homogeneous and
heterogeneous network models retain a huge number of seeds after certain stages of overlay expansion as shown in Figure 3.10(a) and Figure 3.14(a). Once a seed is available, it is the seed’s responsibility that the content for which the tokens are generated is accessible for sharing. The free-riding behaviour of any peer results in the reduction of the trust value, which in turn prohibits the peer in getting the contents with high priority along with the privilege of downloading multiple contents simultaneously. These restrictions and imposition of penalty for non-cooperation make every seed ensure their content available for downloading. Each peer has the contents visible in their web page according to their trust value that can be searched and downloaded as shown in Figure 3.2. This makes the search for any available contents easier. As the infected or polluted contents are removed after the voting process, only good quality meta-data is available forever in the web server. This ensures longer decaying phase of the overlay network than BitTorrent based network.

(d) **Heterogeneous Environment**: As each peer in the practical P2P content sharing network has different bandwidth, delay, and computational capacity, the contribution from the low capability peers, like, mobile peers is considered equally with other high performance peers to calculate the trust values. For every peer after attaining seed status, the tracker allows different number of tokens depending upon its bandwidth and resource capacity. The number of tokens generated normalizes high and low capability peers in attaining the same trust values. The trust value is calculated for individual peers through the ratio of tokens serviced to tokens generated by them as in Algorithm 3.4.

(e) **Infected or Polluted Content Prevention**: In BitTorrent-like systems, there is no mechanism to prevent the infected or polluted content, unless each peer tests the content and deletes them individually. Due to intentional sharing of the infected or polluted content by any malicious provider, higher number of peers waste their bandwidth, time and resources. In the UNITY model, the downloaded content is verified for its genuineness and quality, and consequently the first voting process takes place. During the first voting
process, peers with trust values higher than the provider are allowed to vote. This restriction makes the decision on the content to be highly trustable. Moreover, the average value is considered from the voting to avoid the voting process being compromised. Once the content is found to be infected or polluted, all first stage peers discard the content from further sharing as shown in Figure 3.18. This saves bandwidth and time of the large number of peers simultaneously, prevents spreading of polluted and poisoned content in the network of the UNITY model compared to the BitTorrent network.

(f) **Sybil and Collusion Attacks**: As per the UNITY model design, newcomers will gain nothing from the P2P network. If any peer tries for Sybil attack by forging multiple identities, only one of the identities will gain the trust value to share the contents. When the peers use different identity for uploading polluted or infected contents, overlay will be removed during the first voting by the reputed peers, thus making the peer’s identity to gain no trust values which is equal to the newcomers. This situation makes Sybil attack to be unsuccessful for any number of attempts. During the process of collusion attack, only the initial provider gains trust value from first voting by compromising the other reputed peers. When the formed overlay fails to expand due to polluted or infected contents shared by the reputed peers, the initial voting process is challenged with the partitioned overlay. The tracker can trace the peer’s identity through the overlay number from the trust database as shown in Algorithm 3.1 and penalize it heavily.

### 3.8 SUMMARY

Public P2P networks are distributed systems consisting of interconnected peers that are able to self-organize into network topologies with the purpose of sharing resources such as content, CPU cycles, storage capacity, and bandwidth. The network is capable of adapting to failures and accommodating transient populations.
of peers while maintaining acceptable connectivity and performance. Controlled scalability supporting organized churn rate has solved problems of flash crowd, infected or polluted content, DoS attacks, and bandwidth wastage efficiently than the existing models of BitTorrent-like systems. Comparison of BitTorrent simulation with homogeneous and heterogeneous network models for network size is shown in Figure 3.10 and Figure 3.14. At the initial time, BitTorrent network size exceeds the network size of UNITY models due to controlled scalability of the network. In due course of time, the network size of the proposed UNITY model exceeds BitTorrent simulation provided that a large number of peers are available for content sharing. In heterogeneous network models, the effect of flash crowd are simulated and compared with BitTorrent for which the results are shown in Figure 3.16. The theoretical results are compared with simulation results of UNITY models for the number of seeds in the combined stages shown in Figure 3.13 and Figure 3.17. When an infected or polluted content is shared in the P2P network, the network size is uncontrollable in BitTorrent-like systems, whereas, the UNITY model utilizing controlled scalability efficiently prevents bandwidth wastage and spreading infected content through the voting process.

Unified approach of trust management and protection for peers enhances the reputation system of the P2P network. Though the tracker acts as a central authority, its role is limited to the trust management of the peers. As the tracker decides upon the approval from the reputed peers, the cooperation of the peers in content sharing network is vital. The restriction in privileges and penalty imposed for non-cooperation makes the peers to share the content for a longer time than in existing BitTorrent networks. The proposed UNITY model still supports anonymous mode and free-riding as in BitTorrent-like systems, but are limited to old and verified content with restrictions on multiple simultaneous downloads. The model can be easily deployed in existing BitTorrent-like networks with assured trust both on the network and shared contents. In Chapter 4, modeling and analysis of private P2P content sharing network is discussed with emphasis on the usage of cryptographic suite for secured content sharing.