1

Peer-to-Peer Multimedia Distribution on Radio Channel and Asymmetric Channel

Danilo Merlanti[1] and Gianluca Mazzini[2]
[1]*Department of Engineering, University of Ferrara, Ferrara,*
[2]*Lepida S.p.a., Bologna,*
Italy

1. Introduction

In the Internet and Communication Technology (ICT) field, sharing and distribution of information is very important. Various mechanisms and techniques are used to manage information; one of these is based on peer-to-peer networks. In today's world and in the near future, the exchange and distribution of information will be a very important aspect in the workplace and in daily life. Consequently, mobile devices, devices for home entertainment, personal computers and office terminals must have the mechanisms to achieve the above functionality. Thus the peer-to-peer networks can be used to achieve (Tomoya & Shigeki, 2003) the following: Video conferences or phone calls (Bakos et al., 2006), in which more users can communicate together simultaneously. The distribution of multimedia contents provided by a single source node, for example: streaming distribution of TV contents or radio broadcasting contents (Ciullo et al., 2008; Leonardi et al., 2008; Mazzini & Rovatti, 2008). An example of a real-time algorithm used to create a simple distribution peer-to-peer network on asymmetric channels is given in article (Mazzini & Rovatti, 2008) and the issues of performance of peer-to-peer file sharing over asymmetric and wireless networks is addressed in article (Lien, 2005). Information sharing, for example in a company, the peer-to-peer network system can be used by employees to allow them to work in a shared manner. In daily life, the peer-to-peer network system can be used for sharing personal information such as audio and video contents, documents and others. The more significant peer-to-peer applications used for this purpose are: Gnutella ("The Gnutella Protocol Specification v0.4"; Matei et al, 2002; Wang et al., 2007), Kademlia (Maymounkov & Mazieres, 2002), KaZaA ("http://www.kazaa.com."), Bit-Torrent ("http://www.bittorrent.com."), massively multiplayer online game (MMOG) (Carter et al., 2010; Tay, 2005).

The scenario discussed in this chapter is the distribution of multimedia contents provided by a single source node with an appropriate peer-to-peer network on asymmetric channels and on wireless channels.

This chapter is organized as follows, the scenario and the main hypotheses of the chapter are explained in section 2. Section 3 describes the peer-to-peer algorithms used to build the peer-to-peer distribution networks. In section 4 we present how is estimated the maximum delay of a peer-to-peer distribution network. In this section we present the theoretical optimum in which it is maximized the average maximum number of peers and it is minimized the average maximum delay of the peer-to-peer distribution network. Moreover the simulation results for the asymmetric channel are reported in the last part of this section. In section 5 we analyse the behaviour of the peer-to-peer algorithms in a simple radio channel. In this section we present:

- the radio channel characterization.
- The model used to establish the bit error probability of each peer of a peer-to-peer distribution network.
- The peer-to-peer network simulator used to simulate the behaviour of the radio channel in the peer-to-peer distribution network.
- The validation of the model of the peer-to-peer network in an unreliable environment (radio channel) through the simulation results.
- The results used to establish which peer-to-peer algorithm builds the best peer-to-peer distribution network in an unreliable environment.

The conclusion are presented in the last section of the chapter.

2. Scenario and hypotheses

The scenarios discussed in this chapter refer to the distribution of multimedia contents transmitted by a single source node with an appropriate peer-to-peer network in an asymmetric channel and in a wireless environment.

In this chapter we present two different classes of algorithms. The first class is based on the Tier based algorithm presented in the article (Mazzini & Rovatti, 2008). In this class we have a central entity (server) that manages the insertion of the new peers and the construction of the network.

The second class of algorithms, is based on a peer list. In this class we have a distributed system in which a new node gets from a server, the list of the nodes of the peer-to-peer network and then the new node periodically performs a query flooding to keep the list updated (such as Kademlia (Maymounkov & Mazieres, 2002) is a distributed hash table for decentralized peer-to-peer computer networks).

In this study we are not interested in how the network is managed (centralized or distributed). Instead, by using new algorithms we aim to: maximize the average maximum number of peers that can access the multimedia content and minimize the average maximum delay of the network, in the case of the asymmetric channel, and minimize the bit error probability of each node of the network, in the case of the wireless channel .

In our aim, the source node can be a home-user that streams multimedia content (i.e. audio/video) with a limited output bandwidth ($B < 2$) or a server with a higher output bandwidth ($B \geq 2$) which can supply the content to more than two users, where B is the output bandwidth of the source node.

In the case of the asymmetric channel the building of the network is done in real-time thus the algorithm we use creates a peer-to-peer network for streaming applications in which a source continuously provides content that must be played by a large and unknown number of home-users (Mazzini & Rovatti, 2008). For hypothesis each home-user (peer) is characterized by an asymmetric channel such as ADSL and each peer has a uniform distributed output bandwidth.

An ADSL system with a cooperative bit-loading approach for each peer of the peer-to-peer network (Papandreou and Antonakopoulos, 2003) is used to ensure this hypothesis.

In case of the wireless system, we assume that each peer is an access point and that the network infrastructure is produced by the algorithm in non real-time and the algorithms we use in this chapter suppose that the peer-to-peer network is created before the initializing of the stream; moreover it is supposed that the placement of the various access points (peers) is done so that all wireless links have the same signal to noise ratio.

In both cases, the source node transmits the content while the receiving nodes are able to accept partial streams, from more than one node, through their inbound link and to redistribute it to one or more further peers through their outbound links. In this way the source node supplies the multimedia content to a limited number of requesting peers. The peers, that directly receive the streaming from the source node, provide the multimedia content to the other requesting nodes through a peer-to-peer network. The structure of this network depends on the algorithm used for incremental construction of the peer-to-peer network itself.

The base algorithm considers the source bandwidth as a constraint and minimizes the maximum delay in terms of intermediate links (Mazzini & Rovatti, 2008) without considering the number of nodes that the network is able to accept in accordance with bandwidth constraints.

Below is a list of hypothesis used in the next algorithms:

- the nodes of a peer-to-peer network are characterized by asymmetric channels.
- All peers are always available during the streaming.
- The source node of the network has a finite output bandwidth B.
- The inbound bandwidth of each node is adequate to accept the content.
- All bandwidths are normalized with respect to the bandwidth required to acquire the multimedia content. In this way the bandwidth required to acquire the multimedia content is normalized to 1.
- With respect to the bandwidth referred to above, the output bandwidth of each i-th peer is $0 < \Omega_i < 2$ and Ω_i can be different from Ω_j for each i-th and j-th peer of the network with $i \neq j$.
- Instead in the Mazzini-Rovatti Algorithm (Mazzini & Rovatti, 2008) all the peers have the same output bandwidth value $\Omega \in (0,1)$.
- The delay of each link is normalized to 1.
- In the case of the wireless channel all the links between couple of peers feature an identical signal to noise ratio.

3. Algorithms

In this section we give a brief description of all the algorithms used in this chapter. There are two classes of algorithms that we are going to consider.

3.1 Tiers based algorithms

The first group of algorithms we will consider are classified under the Tier based algorithm (based on Mazzini-Rovatti Algorithm (Mazzini & Rovatti, 2008)). The first new algorithm we introduce is the Tier based algorithm (T). This algorithm is formulated by making a generalization of Mazzini-Rovatti Algorithm (Mazzini & Rovatti, 2008) with an output bandwidth of each peer distributed between 0 and 2. The second new algorithm we introduce is the Tier based algorithm with network Reconstruction (TR). The TR algorithm is formulated from the T algorithm we introduce above and its aim is to maximize the number of the peers accepted in the network. In this algorithm the output bandwidths of the peers of each tier are greater than the output bandwidths of the peers found in the next tier. Moreover when the network produced by the T and TR Algorithms don't accept a new peer for the first time, they don't accept more peers. The third algorithm we introduce is the TR Algorithm without Input Blockage. In this algorithm, if a new peer is not accepted in the network, this peer is inserted into a waiting queue. When a new node able to increase the residual output bandwidth of the network is inserted, the algorithm takes the peers from the waiting list and tries to re-insert them.

A simple analytical formula for the maximum number of nodes accepted in a T network is:

$$n_T = \lfloor B \rfloor + \sum_{i=1}^{T} \lfloor B_i^* \rfloor,\ B_i^* = \sum_{k=1}^{\lfloor B_{i-1}^* \rfloor} \Omega_{k,i} \tag{1}$$

where B_i^* is the output bandwidth of the i-th tier of the network, i = 1…T and T is the maximum number of tiers. B_{i-1}^* is the output bandwidth of the previous tier (available output bandwidth of the previous tier), $B_0^* = B$ and $\lfloor B_i^* \rfloor$ is the maximum number of peers of (i+1)-th tier. $\Omega_{k,i}$ is the output bandwidth of the k-th peer contained in the i-th tier.

3.1.1 State of the art

The state of the art is based on Mazzini-Rovatti's algorithm (Mazzini & Rovatti, 2008). In this algorithm and in the first three new algorithms, the distribution network is organized in "tiers" numbered from 1 onwards. Peers in the first tier are fed by the source. Peers in the j-th tier receive the content only from peers in the (j−1)-th tier. The number of tiers in the distribution network is indicated by T, that also indicates the maximum delay in terms of intermediate links. In Mazzini-Rovatti's algorithm (Mazzini & Rovatti, 2008) a new peer is inserted into the tier closest to the source node. We indicate with p_j the number of peers in the j-th tier. The overall bandwidth required to distribute the content to the j-th tier is p_j , while the overall bandwidth made available by the j-th tier is $\Omega \cdot p_j$. We assume a finite total output bandwidth B offered to the first tier by the source node. The elementary step of Mazzini-Rovatti's algorithm is "add a peer to the j-th tier if possible". We indicate this step with A(j). A pseudo-code for A(j) is the following:

```
case j = 1
        if B > p_1 then p_1 ++ else failed
case j = 2, . . . , T
        if p_{j-1} ≥ p_j + 1 then p_j ++ else failed
case j = T + 1
        if p_T ≥ 1 then p_{T+1} = 1  T ++ else failed
```

where a peer is added to the j-th tier if and only if the bandwidth emitted by the previous tier (i.e. the source if j = 1) is enough to accommodate it, namely if this bandwidth is ≥ 1. The new peer insertion algorithm has the following pseudo-code:

```
for j = 1 to T + 1
    if A(j) not failed then stop
next
failed
```

This algorithm tries to add the new peer to the smallest-delay tier within bandwidth constraints and fails if no more peers can be fed by peers in the same tier.

The new algorithms aim to maximize the number of accepted nodes (to increase the number of users that can have access to the content), minimize the reconstruction delay and minimize the maximum delay of the network (to provide a better service).

The next subsections describe and introduce new algorithms, adopted for the distribution of multimedia contents.

3.1.2 Tiers based algorithm (T)

The generalization of Mazzini-Rovatti's algorithm (Mazzini & Rovatti, 2008), with the new hypothesis that is " $0 < \Omega_i < 2$ for each i-th peer", is provided by the Tiers based algorithm (T). A pseudo-code for this algorithm is the following:

case j = 1
    if $B \geq p_1 + 1$ then $p_1{+}{+}$ else failed
case j = 2, . . . , T
    if $\sum_{i=1}^{p_{j-1}} \Omega_{i,j-1} \geq p_j + 1$ then $p_j{+}{+}$ else failed
case j = T + 1
    if $\sum_{i=1}^{p_T} \Omega_{i,T} \geq 1$ then $p_{T+1} = 1$, $T{+}{+}$ else failed

where p_j is the number of peers contained in the j-th tier (with j = 1…T) and $\Omega_{i,j-1}$ is the output bandwidth of the i-th peer contained in the (j−1)-th tier of the network and a new peer is added to the j-th tier if and only if the bandwidth emitted by the previous tier (i.e. the source if j = 1) is able to accommodate it. The new peer insertion algorithm has the following pseudo-code:

```
for j = 1 to T + 1
        if A(j) not failed then stop
next
failed
```

This algorithm tries to add the new peer to the smallest delay tier within bandwidth constraints and fails if no more peers can be fed by peers in the same tier.

In the next section we describe an algorithm that increases the maximum number of peers accepted in the network, by adopting an insertion algorithm with the reconstruction of the network itself.

3.1.3 Tiers based algorithm with network Reconstruction (TR)

In this algorithm the tiers nearest to the source node must hold the nodes characterized by the greatest output bandwidth values. To guarantee this aim, the insertion of each new node can trigger a possible reconstruction of the distribution network. The elementary step of the TR algorithm is indicated with the recursive function $A(j,\Omega_N)$. A pseudo-code for $A(j,\Omega_N)$ is the following:

case j = 1
    if $B \geq p_j + 1$ then $p_j{+}{+}$
    else $<remove\ \ N_{m_j}>$, $<add\ \ N>$, $A(j+1, B_{m_j})$
case j = 2, …, T
    if $\sum_{i=1}^{p_{j-1}} \Omega_{i,j-1} \geq p_j + 1$ then $p_j{+}{+}$
    else $<remove\ \ N_{m_j}>$, $<add\ \ N>$, $A(j+1, B_{m_j})$
case j = T+1
    if $\sum_{i=1}^{p_T} \Omega_{i,T} \geq 1$ then $p_{T+1} = 1$ $T{+}{+}$ else failed

where p_j is the number of peers contained in the j-th tier (with j = 1…T) and $\Omega_{i,j-1}$ is the output bandwidth of the i-th peer contained in the (j−1)-th tier of the network. Ω_N is the output bandwidth of the new node N. N_{m_j} is the peer with the minimum output bandwidth of the j-th tier because the hypothesis of this algorithm is that each j-th tier (with j = 1…T-1) holds all the peers characterized by output bandwidths greater than the output bandwidths of the peers held in the (j+1)-th tier. B_{m_j} is the output bandwidth of the peer N_{m_j}. In this way the algorithm tries to add the new peer in the j-th tier if $\Omega_N > B_{m_j}$.

The new peer insertion algorithm has the following pseudo-code:

for j = 1 to T + 1
    if $\left(\min\left(\text{NodesOutputBandwidth}\left(\text{tier}_j\right)\right) < \Omega_N\right)\text{and}\left(j < T+1\right)$ then break
next
if $A(j,\Omega_N)$ not failed then stop else failed

This insertion algorithm tries to insert the new peer (N) in the j-th tier where the output bandwidth of the peer, characterized by the minimum output bandwidth, is less than the output bandwidth (Ω_N) of the new peer (N).

3.1.4 TR algorithm without Input Blockage (TRwIB)

The TRwIB algorithm derives from the TR algorithm. The TRwIB is the TR algorithm without Input Blockage.

The engine of this algorithm is the following:

- if the new peer can be inserted into the network (with network reconstruction if it is necessary) then the algorithm performs the insertion operation as the TR algorithm.
- Otherwise the new peer is inserted in a waiting queue; this queue contains the peers that are waiting a new node able to increase the residual output bandwidth of the network. When this event happens the algorithm wakes up the waiting peers and it tries to insert them. The peers, that are not inserted, are maintained in the waiting queue and they are waked up (by the insertion algorithm) if and only if a new peer, is able to increase the residual bandwidth of the network with its insertion.

The disadvantage of this algorithm is represented by the network reconstruction that introduces an additional delay to the network.

3.2 Peer List based algorithms

The second group of algorithms we will consider are classified under the peer list algorithm. The first new algorithm we introduce is the Peer List based Algorithm (PL). In this algorithm, the peer-to-peer distribution network is represented by a peers list, in which each peer is characterized by an id, its available output bandwidth and an id list of children nodes. At the beginning of this algorithm, the peers list contains only the source node. When a new node N wants access to the network, this peer requests, to the peers of the network, an amount of bandwidth equal to the bandwidth required to acquire the multimedia content. If N obtains the required bandwidth, from the network, then the new node is added to the network; otherwise the network isn't able to accept more peers. The second algorithm we consider is the PL algorithm with Reconstruction (PLR). This algorithm is formulated from the PL algorithm. The PLR algorithm inserts the new node (N) in the network if and only if N is able to increase the residual bandwidth of the network. In this case the PLR algorithm extracts the peer (of the network) with minimum output bandwidth and replaces it with the new node. The PLR algorithm exploits the increase of the residual bandwidth (brought by N) by re-inserting the extracted node. If N isn't able to increase the network residual bandwidth then the PLR algorithm doesn't insert N into the network and the network accepts no more peers. The third algorithm we consider is the PLR Algorithm without input blockage. In this algorithm, if the new peer is not accepted in the network, this peer is inserted into a waiting queue. When a new node able to increase the residual output bandwidth of the network is inserted, this algorithm wakes up and tries to insert the waiting peers.

A simple analytical formula for the maximum number of nodes accepted in a Peer List based network is:

$$n_{PL} = \left\lfloor B + \sum_{i=1}^{n_{PL}} \Omega_i \right\rfloor \qquad (2)$$

where Ω_i is the output bandwidth of the i-th peer of the network. In this way from the formulas (1) and (2) it is immediately proof that $n_{PL} \geq n_T$.

In a Tiers based network the maximum depth is T. If we collect the unused bandwidth B_u of the tiers and if $B_u \geq 1$ we can supply one or more new peers. In this way, in a Tiers based network , if we supply one or more new peers using the unused bandwidth B_u then the Tiers based network degenerates into a Peer List based network and in this case the maximum depth is ≥ T + 1. Thus $depth_{PL} \geq depth_T$. Therefore the Peer List based networks are optimal with respect to the maximum number of nodes accepted by the network but they don't have the minimum delay in terms of the maximum depth.

In the TR, TR without input blockage, PLR and PLR without input blockage algorithms the insertion of a new peer can trigger a reconstruction of the network required in order to maintain order in the structure of the network. The reconstruction makes a delay. Thus the maximum delay of the network, in terms of the maximum depth of the network, has to be increased by the reconstruction delays.

3.2.1 Peer List based algorithm (PL)

In the PL algorithm the peer-to-peer distribution network is represented by a peers list, where each peer is characterized by an id, its available output bandwidth and the id list of children nodes. At the beginning, the peers list contains only the source node. When a new node N wants access to the network, this peer requests, to the peers of the network, an amount of bandwidth equal to the bandwidth required to acquire the multimedia content. If N obtains the required bandwidth, from the network, then the new node is added to the network; otherwise the network isn't able to accept more peers. The next algorithm is an improvement of this algorithm and it allows to increase the maximum number of peers accepted in the network.

3.2.2 Peer List based algorithm with Reconstruction (PLR)

This algorithm has the same behaviour as the PL algorithm, when the network is able to accept a new peer otherwise it tries to insert this peer with a reconstruction of the network. The algorithm inserts the new node (N) in the network if and only if N is able to increase the residual bandwidth ($B + \left(\sum_{i=0}^{n} \Omega_i\right) - n$, where n is the number of peers of the network) of the network. In this case the algorithm extracts the peer (of the network) with minimum output bandwidth and replaces it with the new node. The algorithm exploits the increase of the residual bandwidth (brought by N) to re-insert the extracted node. If N isn't able to increase residual bandwidth of the network then the algorithm doesn't insert N into the network and the network accepts no more peers. The PLR as well as TR algorithm may have network reconstruction.

The PLR algorithm has the analytical formulation (2) where the output bandwidths of the peers are $\Omega_1 \geq \Omega_2 \geq ... \geq \Omega_n > 0$ and n is the number of peers accepted by the network.

3.2.3 PLR algorithm without Input Blockage (PLRwIB)

The PLRwIB algorithm derives the PLR algorithm. The PLRwIB is the PLR algorithm without input blockage.

The engine of this algorithm is the following:

- if the new peer can be inserted into the network (with network reconstruction if it is necessary) then the algorithm performs the insertion operation as the PLR algorithm.
- Otherwise the new peer is inserted in a waiting queue; this queue contains the peers that are waiting a new node able to increase the residual output bandwidth of the network. When this event happens the algorithm wakes up the waiting peers and it tries to insert them. The peers, that are not inserted, are maintained in the waiting queue and they are waked up (by the insertion algorithm) if and only if a new peer, is able to increase the residual bandwidth of the network with its insertion.

The disadvantage of this algorithm is represented by the network reconstruction that introduces an additional delay to the network.

4. Asymmetric channel

For the analysis of the peer-to-peer algorithms we introduced in section 3, we formulate a theoretical optimum in which the maximization of the average maximum number of the peers and the minimization of the average maximum delay of the network is achieved. We compare the results, in terms of the average maximum number of peers and the average maximum delay of the network, of the algorithms presented above, with respect to the theoretical optimum.

4.1 Maximum delay of a peer-to-peer distribution network

To estimate the maximum delay of a peer-to-peer distribution network, we suppose to have two different cases:

- in the first case, the network is generated by an algorithm (such as the T algorithm or the PL algorithm) that doesn't use a reconstruction of the network. In this case the maximum delay of the peer-to-peer distribution network is defined as the maximum depth of this network.
- In the second case, the network is generated by an algorithm (such as the TR, TRwIB, PLR and PLRwIB algorithms) that uses a reconstruction of the network. In this scenario the maximum delay of the peer-to-peer distribution network is defined as the maximum depth of this network plus the amount of the delays generated by each reconstruction of the network. For the insertion of a new peer N, we have a reconstruction of the network when it is necessary to extract a peer of the peer-to-peer network, replace it with the new peer N and the insertion algorithm exploits the increase of the residual bandwidth (brought by N) to re-insert the extracted node. The reconstruction delay for the insertion of a new node is the amount of replacement delays. The delay produced by each k-th substitution of two peers (p_1 and p_2) is: $d_{s_k} = \sum_{j=1}^{M} S_{cp} / B(p_1, j) + \sum_{h=1}^{W} S_{cp} / B(p_1, h)$, where: $B(p_1, j)$ is the bandwidth between the extracted peer p_1 and the j-th parent peer of p_1, M is the number of the parent peers of the peer p_1, $B(p_1, h)$ is the bandwidth between the extracted peer p_1 and the h-th child peer of p_1, W is the number of the child peers of the peer p_1 and the peer p_1 sends to its parent nodes and its child nodes a control packet (with size S_{cp}) used to perform the node replacement. Thus the reconstruction delay for the insertion of the i-th node is:

$d_i = \sum_k d_{s_k}$ and $d_i = 0$ if the are no node replacements for the insertion of the i-th peer.

Therefore the total reconstruction delay of a peer-to-peer network is: $d = \sum_{i=1}^{n} d_i$, where n is the maximum number of peers accepted by the network.

4.2 Theoretical optimum

The theoretical optimum is achieved when the ratio between the average maximum number of peers (n) and average minimum possible maximum delay (d) of the network is maximized. We indicate with, $\overline{n_1}$, the mean maximum number of peers accepted in the network with an output bandwidth between 1 and 2 ($1 \le \Omega_i < 2$). We indicate with, $\overline{n_2}$, the mean maximum number of peers with an output bandwidth between 0 and 1 ($0 < \Omega_i < 1$). We have the average maximum number of peers in the network with the minimum possible value of T if and only if the peers (that access the network) are ordered with respect to their output bandwidth namely $\Omega_1 \ge \Omega_2 \ge \Omega_3 \ge ... \ge \Omega_n$, where Ω_i (for $i = 1...n$) is the output bandwidth of the i-th peer that has access the network. In this way there are no reconstructions of the peer-to-peer network and there are no reconstruction delays. With this conditions we can partially apply the theoretical formulation of the article (Mazzini & Rovatti, 2008) to achieve the optimum value of the ratio between n and T.

In this way the network is divided in two parts. In the first part there are all the peers with output bandwidth $1 \le \Omega_i < 2$ (with average output bandwidth $\overline{\Omega^1}$) and they receive the multimedia content from the source ($\overline{n_1}$ number of peers and $\overline{T_1}$ the average minimum possible maximum delay of the first part of network). In the second part there are all the peers with output bandwidth $0 < \Omega_i < 1$ (with average output bandwidth $\overline{\Omega^2}$) and they receive the multimedia content from the leaf peers of the first part of the network ($\overline{n_2}$ number of peers and $\overline{T_2}$ the average minimum possible maximum delay of the first part of network).

The average maximum number of peers accepted in the network is $\overline{n} = \overline{n_1} + \overline{n^*}$. When $\left(\sum_{i=1}^{\overline{n}} \Omega_i\right) / \overline{n} < 1$ the system reaches its maximum number of peers.

We suppose to have n_T number of peers wants to access the peer-to-peer network. Thus the number of peers of the first part of the network is:

$$\overline{n_1} = \lfloor n_T \cdot (1 - p) \rfloor \quad (3)$$

Where p is the probability that each i-th node has output bandwidth $0 < \Omega_i < 1$ and $1 - p$ is the probability that each i-th node has output bandwidth $1 \le \Omega_i < 2$.

The average minimum possible maximum delay (formulation of the article (Mazzini & Rovatti, 2008) section III) in a $\overline{n_1}$ - nodes (non necessarily tiered) peer-to-peer network fed by a source with bandwidth B is:

$$\overline{T_1} = \left\lceil \log_{\overline{\Omega^1}} \left(\frac{\overline{n_1}}{B} \cdot (\overline{\Omega^1} - 1) + 1 \right) \right\rceil \tag{4}$$

The formula (4) give us the average minimum maximum delay of the first part of the network, because it is achieved through the average maximum number of peers $\overline{n_1}$.

The output bandwidth provided by the first part of the network to the second part of the network is:

$$\overline{B_{N_1}} = B + \overline{\Omega^1} \cdot \overline{n_1} - \overline{n_1} = B + \overline{n_1} \cdot (\overline{\Omega^1} - 1) \tag{5}$$

where B is the output bandwidth of the source.

The total residual bandwidth of the second part of the network is:

$$\overline{B_{N_2}} = \overline{B_{N_1}} + \overline{n^*} \cdot (\overline{\Omega^2} - 1) \tag{6}$$

where $\overline{n^*}$ is the mean maximum number of peers accepted in the second part of the peer-to-peer network.

When $\overline{B_{N_2}} = 1$ the second part of the network reaches the maximum number of peers – 1. Thus from the formula (6) the average maximum number of peers of the second part of the network is:

$$\overline{n^*} = \left\lfloor \frac{\overline{B_{N_1}} - 1}{1 - \overline{\Omega^2}} + 1 \right\rfloor \tag{7}$$

Therefore the average minimum possible maximum delay of the second part of the network (formula presented in section III of the article (Mazzini & Rovatti, 2008)) is:

$$\overline{T^*} = \left\lceil \log_{\overline{\Omega^2}} \left(\frac{\overline{n^*}}{\overline{B_{N_1}}} \cdot (\overline{\Omega^2} - 1) + 1 \right) \right\rceil \tag{8}$$

The formula (8) give us the average minimum maximum delay of the second part of the network, because it is achieved through the average maximum number of peers $\overline{n^*}$.

With the formula (7) the formula (8) becomes:

$$\overline{T^*} = \left\lceil \log_{\overline{\Omega^2}} \left(\frac{1}{\overline{B_{N_1}}} \right) + 1 \right\rceil \tag{9}$$

In this way if we define:

$$\overline{n_2} = \overline{n^*} - 1 = \left\lfloor \frac{\overline{B_{N_1}} - 1}{1 - \overline{\Omega^2}} \right\rfloor \tag{10}$$

then the average minimum possible maximum delay (formulation of the article (Mazzini & Rovatti, 2008) section III) in a $\overline{n_2}$ - nodes (non necessarily tiered) peer-to-peer network fed by an equivalent source with bandwidth $\overline{B_{N_1}}$ is:

$$\overline{T_2} = \begin{cases} 0 & if \ \overline{B_{N_1}} < 1 \\ 1 & if \ \overline{B_{N_1}} = 1 \\ \left\lceil \log_{\overline{\Omega^2}} \left(\frac{1}{\overline{B_{N_1}}} \right) \right\rceil & if \ \overline{B_{N_1}} > 1 \end{cases} \tag{11}$$

The formula (11) give us the average minimum maximum delay of the second part of the network, because it is achieved through the average number of peers $\overline{n_2}$.

Thus $\overline{T_2} = \overline{T^*} - 1$ and $\overline{n_2} = \overline{n^*} - 1$. Using the formulas (7), (9), (10) and (11), we can simply show that: $(\overline{n_1} + \overline{n_2}) / (\overline{T_1} + \overline{T_2}) \geq (\overline{n_1} + \overline{n^*}) / (\overline{T_1} + \overline{T^*})$. In conclusion, if $\overline{B_{N_1}} > 1$, the optimum for the ratio between the average maximum number of peers and the average minimum possible maximum delay of the network is:

$$\frac{\overline{n_1} + \overline{n_2}}{\overline{T_1} + \overline{T_2}} = \frac{\lfloor n_T \cdot (1-p) \rfloor + \left\lfloor \frac{\overline{B_{N_1}} - 1}{1 - \overline{\Omega^2}} \right\rfloor}{\left\lceil \log_{\overline{\Omega^1}} \left(\frac{\overline{n_1}}{B} \cdot (\overline{\Omega^1} - 1) + 1 \right) \right\rceil + \left\lceil \log_{\overline{\Omega^2}} \left(\frac{1}{\overline{B_{N_1}}} \right) \right\rceil} \tag{12}$$

4.3 Results

The comparison of the algorithms is performed by using the ratio between the average maximum number of peers (n) and the average maximum delay (d) of the network over 1000 samples of peers. The simulator uses 1000 different samples (random generated) and each sample contains 1000 peers. We now briefly describe the network parameters followed when making the comparison of the performance of the algorithms we discussed about in section 3. The value of the output bandwidth of the source node is $B \in [1,10]$. p is the probability that each i-th node has output bandwidth $0 < \Omega_i < 1$ and $1 - p$ is the probability that each i-th node has output bandwidth $1 \leq \Omega_i < 2$. We are supposed to have an uniform distribution for p, where $p \in [0,1]$. For each value of p between 0 and 1; with step of 0.01 in the simulation environment; the simulator uses 1000 different samples and each sample contains 1000 peers. We suppose that the size of the control packet used to replace a peer with a new peer is equal to 642 bits (where 192 bits are for the TCP header (RFC 793, 1981), 192 bits are for the IPv4 header (RFC 791, 1981), 96 bits of data make up of 32 bits for the IP address of the extracted peer, 16 bits for the port of the extracted peer, 32 bits for the IP address of the new peer, 16 bits for the port of the new peer and 162 bits for the lower layers). The simulation results give a map of the best algorithms with respect to the ratio between the average maximum number of peers and the average maximum delay of the network as functions in term of p and B . The space B, p ; with $1 \leq B \leq 10$ and $0 \leq p \leq 1$; is divided in three areas. The first area has $1 \leq B \leq 2$ and $0 \leq p \leq 1$. In this area the PLR algorithm without input blockage is closest to the optimum because it produces random trees (with $n / d > 1$) while all the Tier based algorithms produce networks that are chains of peers ($n / d = 1$). The second area has $2 \leq B \leq 10$ and $0.46 \leq p \leq 1$. In this area the best algorithm is the TR algorithm without Input Blockage. The third area has $2 \leq B \leq 10$ and $0 \leq p < 0.46$. In this area the best algorithm is the PLR algorithm without Input Blockage. The confidence intervals of n / d (with respect to B and p) have been evaluated for each algorithm and they have a maximum size of $4.6 \cdot 10^{-3}$, thus they are negligible in this approach.

5. Radio channel

We now briefly analyse the behaviour of the algorithms described above in a simple radio channel characterization; moreover the algorithm with the maximum percentage of bits correctly received is established.

5.1 Radio channel characterization

This subsection describes a simple radio channel characterization. Each wireless link between nodes is represented as an ideal wireless link with the following characteristics: the error probabilities over received bits are not independent (in the previous article (Merlanti & Mazzini, 2009) the error probabilities over received bits were independent). The average bit error probability with respect to small scale fading effects and coherent four phase PSK modulation is given as (Pages 785-486 formulas 14-4-36 and 14-4-38 of (J. Proakis, 1995)):

$$P_b = \frac{1}{2}\left[1 - \frac{\mu}{\sqrt{2-\mu^2}} \sum_{k=0}^{L-1} \binom{2k}{k} \left(\frac{1-\mu^2}{4-2\mu^2}\right)^k\right] \tag{13}$$

Where L is the order of diversity (for our channel L=4) and μ is the cross-correlation coefficient with perfect estimation given as (page 786 formula 14-4-36 or table C-1 page 894 of (J. Proakis, 1995)):

$$\mu = \sqrt{\frac{\overline{\gamma_c}}{1+\overline{\gamma_c}}} \tag{14}$$

Where $\overline{\gamma_c}$ is the average Signal to Noise Ratio with respect to small scale fading effects.

5.2 Analytical formulation model

In this section we present an analytical formulation (Merlanti & Mazzini, 2009) used to establish the bit error probability of each node of the network, produced by the previous algorithms. The main hypothesis used for this analytical model (and used in the simulator presented in the next section) are as follows:

- stationary network: during the simulation the system doesn't insert new nodes in the network because the aim is to estimate the network behaviour with an unreliable radio channel.
- Each segment sent by a peer i to another peer j has a constant and fixed dimension $d_{i,j}$.
- Each peer has one or more parent nodes from which it obtains the content; the content (namely the packet) is distributed among the parent peers with a static allocation, for example each peer receives the first segment of each packet from the first parent node, ..., each peer receives the n-th segment of each packet from the n-th parent node and so on.
- Each peer is identified by a unique peer ID; the peer ID of the source node is 0 and the network peers have incremental peer ID value starting from 1.
- The source node has each packet and transmits it to the peers directly connected to source node.
- The analytical formulation and the simulator considers only the uncoded communication between peers and the probability P_b is the average (with respect to small scale fading effects) bit error probability on decoded word. In this way if there is an error on a bit in the considered decoded segment then the entire segment is lost.

Consider the j-th node of the network:

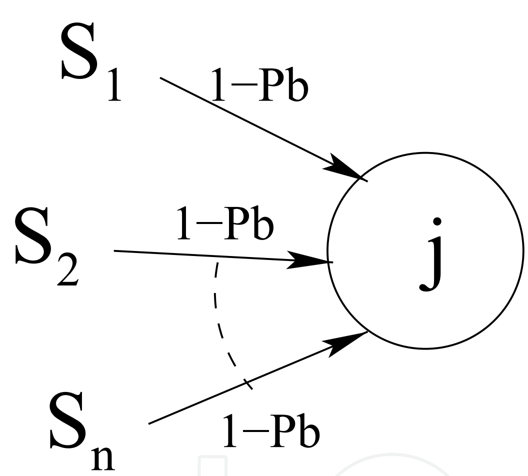

Fig. 1. j-th node of a p2p network

Suppose that the packet is divided in n segments and these are obtained from different parent nodes. So the j-th node receives the segments S_1… S_n of the packet from n different nodes. Each segment S_i (where i = 1…n) has g_i bits and suppose that these bits have different bit error probability namely, the first bit (b_1) of the segment S_i has a bit error probability equal P_{b1} ... the g_i-th bit (b_{gi}) of the segment S_i has a bit error probability equal P_{bgi}. In this way for each bit, the correct bit probability is: for the first bit is $P_1 = 1 - P_{b1}$ … for the g_i-th bit is $P_{gi} = 1 - P_{bgi}$. Now we have to establish the probability that the segment S_i is received correctly. A segment is correct if all the bits of this segment are received without errors. So the desired probability has the following expression:

$$P(S_i) = P_1 \cdot \ldots \cdot P_{g_i} \cdot (1 - P_b)^{g_i} \tag{15}$$

where P_b is the bit error probability of the radio channel and i=1…n. Therefore the average correct bit probability for the j-th node is:

$$\overline{P}(j) = \frac{\sum_{i=1}^{n} P(S_i)\cdot g_i}{\sum_{i=1}^{n} g_i} \tag{16}$$

This formula give us the wireless link model for each node of the network. Moreover, for the nodes directly connected to the source, the probabilities $P_1 \ldots P_{gi}$ for the segment S_i (where i = 1…n) have the following value:

$$P_1 = P_2 = \ldots = P_{g_i} = 1 \tag{17}$$

The bit error probability P_b of the radio channel, used in this section, is obtained through the formulas (13) and (14) with $\overline{\gamma_c}$ equal to the desired SNR.

5.3 Peer-to-peer network simulator

Each peer-to-peer network is simulated in the following way (Merlanti & Mazzini, 2009): for each packet transmitted by the source node S, the simulator analyses the peers in the order defined by their peer ID; for each i-th peer (where i = 1…n), the simulator performs the following operation: it searches the parent nodes of the i-th peer (we indicate this node with N). For each parent node N_f (N_f is the source node if N receives the packet from S), the simulator determines if N_f has the segment of the packet expected by N:

- if N_f has the segment then the simulator determines if N receives it without errors; this is done, whilst simulating the behaviour of the channel for each segment bit sent from N_f to N: the system generates a random number v uniformly distributed in [0,1); with this number the simulator establishes if the bit is lost or is correctly received. The bit is lost if $0 \le v \le P_b$. The bit is correctly received if $v > P_b$; where the parameter P_b is obtained through the formulas (13) and (14) with $\overline{\gamma_c}$ equal to the desired SNR. If the number of lost bits of the segment is greater than 0 then the entire segment is lost and therefore the simulator adds the number of bits of the segment to the number of bits lost by N. Otherwise the segment is correctly received and therefore the simulator adds the number of bits of the segment to the number of bits correctly received by N.
- If N_f doesn't have the segment, then the simulator adds the number of bits of the segment to the number of bits lost by N.

At the end of the simulation for each peer the system produces the number of the bits correctly received and the number of the bits lost.

5.4 Model validation through simulator

In order to validate the model of the network in an unreliable environment (radio channel) we use the autocorrelation test (pages 423-426 of the Book (Soderstrom & Stoica, 1989)).

We define the residuals $\varepsilon(t)$ as:

$$\varepsilon(t) = y(t) - \hat{y}(t) \tag{18}$$

where $y(t)$ are the simulated results about the average percentage of correctly received bits for each depth t of the network and $\hat{y}(t)$ are the results produced by the model.

If the model is accurately describing the observed data $y(t)$, then the residuals $\varepsilon(t)$ should be white. A way to validate the model is thus to test the hypotheses:

- H_0 : $\varepsilon(t)$ is a white sequence;
- H_1 : $\varepsilon(t)$ is not a white sequence.

The autocovariance of the residuals $\varepsilon(t)$ is estimated as:

$$\hat{r}_\varepsilon(\tau) = \frac{1}{N} \cdot \sum_{t=1}^{N-\tau} \varepsilon(t+\tau) \cdot \varepsilon(t) \tag{19}$$

where N is the maximum depth of the peer-to-peer distribution network.

If H_0 holds, then the square covariance estimates is asymptotically χ^2 distributed namely:

$$\frac{N}{\hat{r}_\varepsilon^2(0)} \cdot \sum_{i=1}^{m} \hat{r}_\varepsilon^2(i) \longrightarrow \chi^2(m) \tag{20}$$

where m is the number of degrees of freedom and it is equal to the maximum depth of the peer-to-peer distribution network.

Let x denote a random variable which is χ^2 distributed with m degrees of freedom. Furthermore, we define $\chi_\alpha^2(m)$ by:

$$\alpha = P(x > \chi_\alpha^2(m)) \tag{21}$$

For $\alpha = 0.01$ we have:

- if $\frac{N}{\hat{r}_\varepsilon^2(0)} \cdot \sum_{i=1}^{m} \hat{r}_\varepsilon^2(i) > \chi_\alpha^2(m)$ then we reject H_0.
- if $\frac{N}{\hat{r}_\varepsilon^2(0)} \cdot \sum_{i=1}^{m} \hat{r}_\varepsilon^2(i) \le \chi_\alpha^2(m)$ then we accept H_0.

We can see this property through the normalized covariance $\hat{r}_\varepsilon(\tau) / \hat{r}_\varepsilon(0)$. In this case $x_\tau = \hat{r}_\varepsilon(\tau) / \hat{r}_\varepsilon(0)$ and we plot (for each peer-to-peer algorithm, in the worst case, SNR = 4 dB) x_τ versus τ and a 99% confidence interval for x_τ.

Since $x_\tau \longrightarrow \mathbb{N}(0,1/N)$ the lines in the diagram are drawn at $x = \pm 2.5758 / \sqrt{N}$. It can be seen from the figures 2 – 7 (for all the peer-to-peer algorithms) that x_τ lies in this interval. One can hence expect $\varepsilon(t)$ is a white process for all the peer-to-peer algorithms.

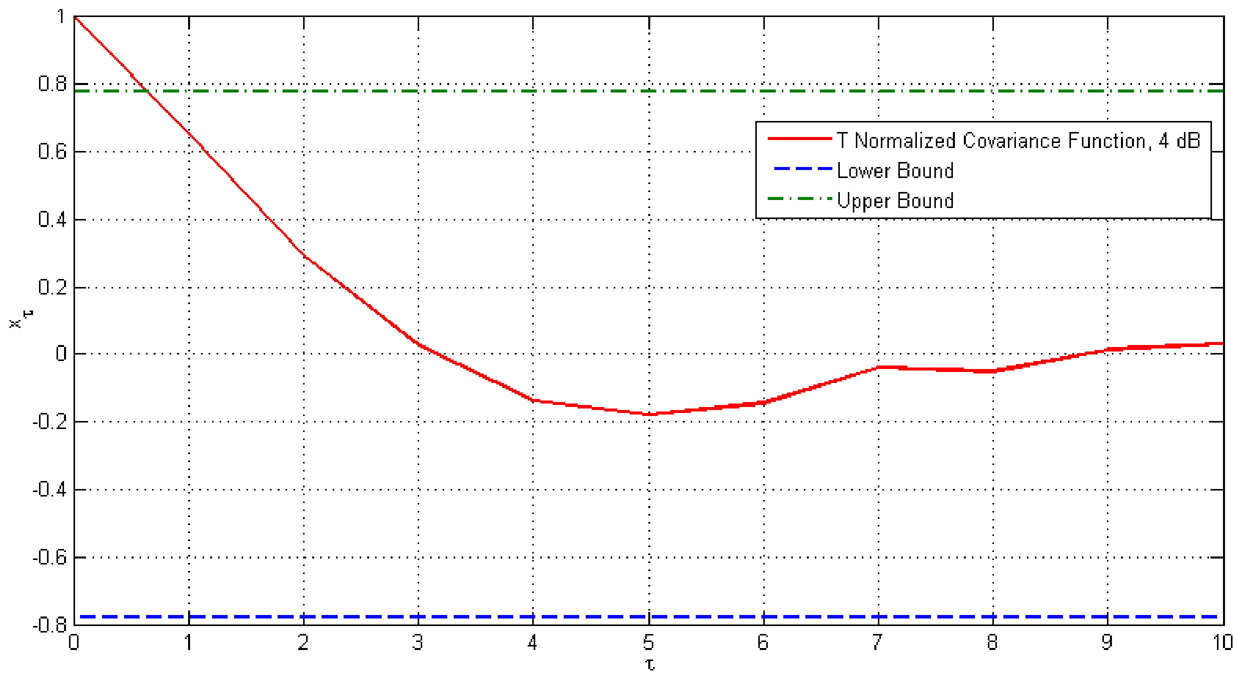

Fig. 2. Normalized covariance function of $\varepsilon(t)$ for the T algorithm in the worst condition (SNR = 4 dB)

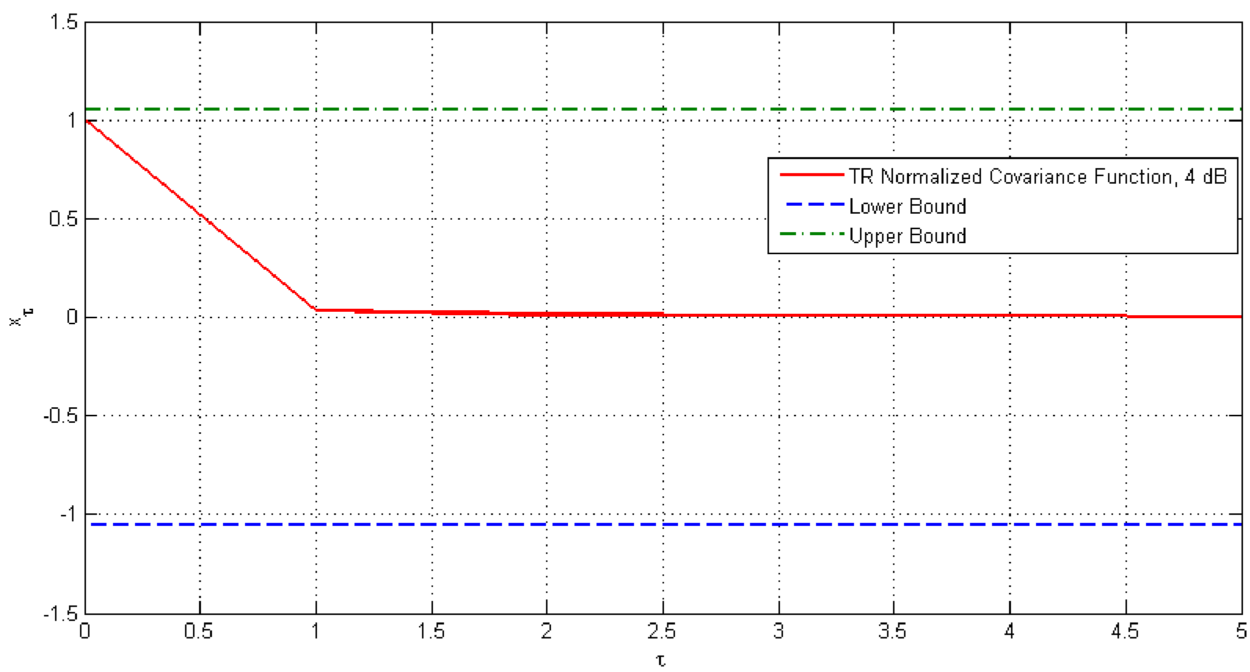

Fig. 3. Normalized covariance function of $\varepsilon(t)$ for the TR algorithm in the worst condition (SNR = 4 dB)

The result of the hypotheses test for each peer-to-peer algorithm is:

- T algorithm: the test quantity (20) is 17.5213 and $\chi^2_\alpha(m)$ is 24.7250 thus the variable $\varepsilon(t)$ is, under the null hypothesis H_0, approximately $\chi^2_\alpha(m)$.
- TR algorithm: the test quantity (20) is 14.0130 and $\chi^2_\alpha(m)$ is 16.8119 thus the variable $\varepsilon(t)$ is, under the null hypothesis H_0, approximately $\chi^2_\alpha(m)$.
- TRwIB algorithm: the test quantity (20) is 16.7519 and $\chi^2_\alpha(m)$ is 21.6660 thus the variable $\varepsilon(t)$ is, under the null hypothesis H_0, approximately $\chi^2_\alpha(m)$.
- PL algorithm: the test quantity (20) is 27.8567 and $\chi^2_\alpha(m)$ is 29.1412 thus the variable $\varepsilon(t)$ is, under the null hypothesis H_0, approximately $\chi^2_\alpha(m)$.
- PLR algorithm: the test quantity (20) is 57.1550 and $\chi^2_\alpha(m)$ is 63.6907 thus the variable $\varepsilon(t)$ is, under the null hypothesis H_0, approximately $\chi^2_\alpha(m)$.
- PLRwIB algorithm: the test quantity (20) is 154.0808 and $\chi^2_\alpha(m)$ is 180.7009 thus the variable $\varepsilon(t)$ is, under the null hypothesis H_0, approximately $\chi^2_\alpha(m)$.

In this case for all the peer-to-peer algorithms described above we observe that the prediction error $\varepsilon(t)$ is white with a level of significance $\alpha = 0.01$ thus the model is validated for all the algorithms.

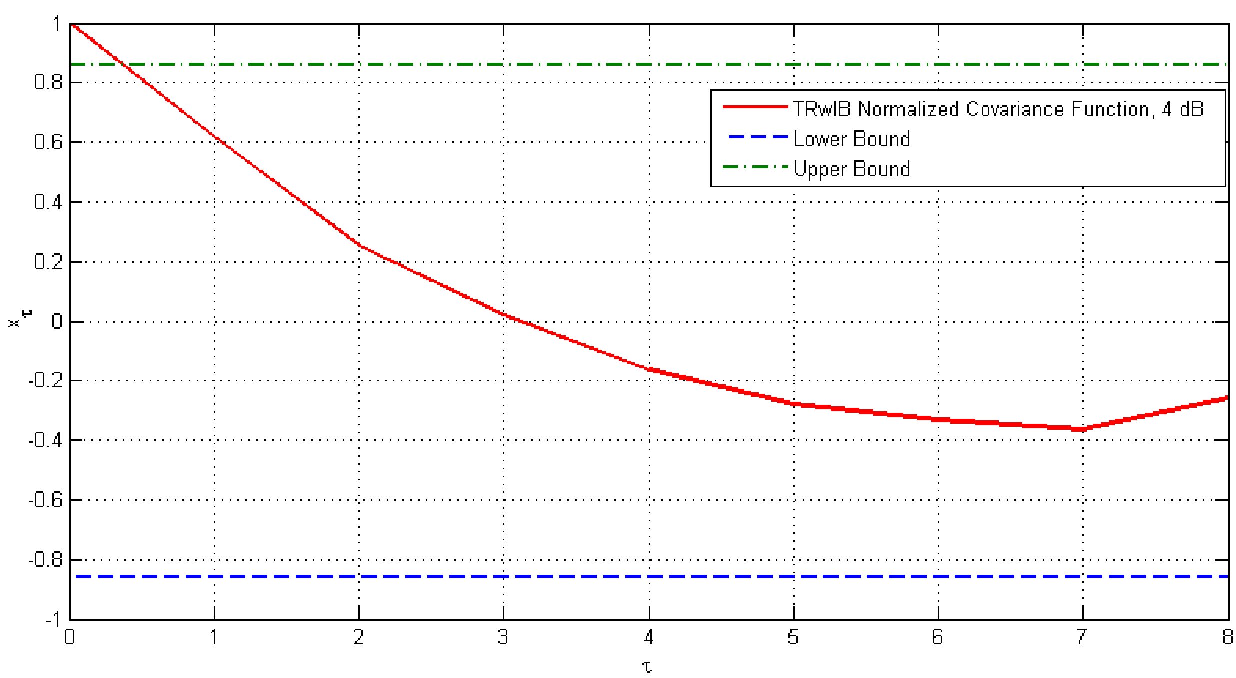

Fig. 4. Normalized covariance function of $\varepsilon(t)$ for the TRwIB algorithm in the worst condition (SNR = 4 dB)

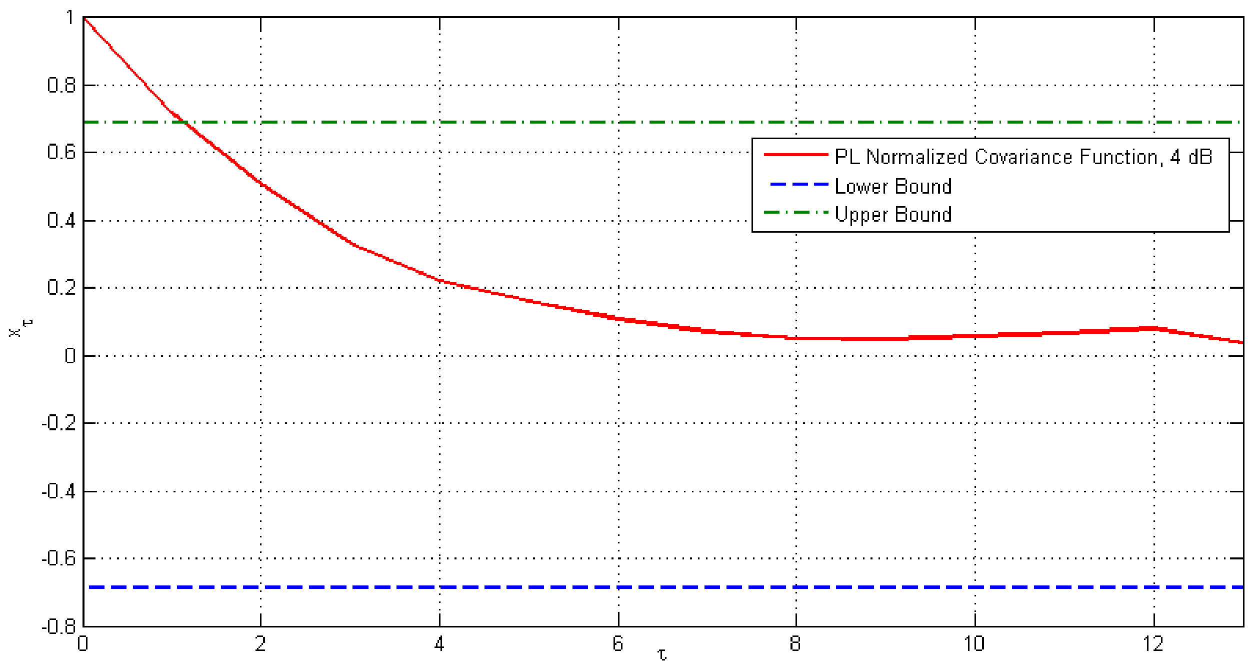

Fig. 5. Normalized covariance function of $\varepsilon(t)$ for the PL algorithm in the worst condition (SNR = 4 dB)

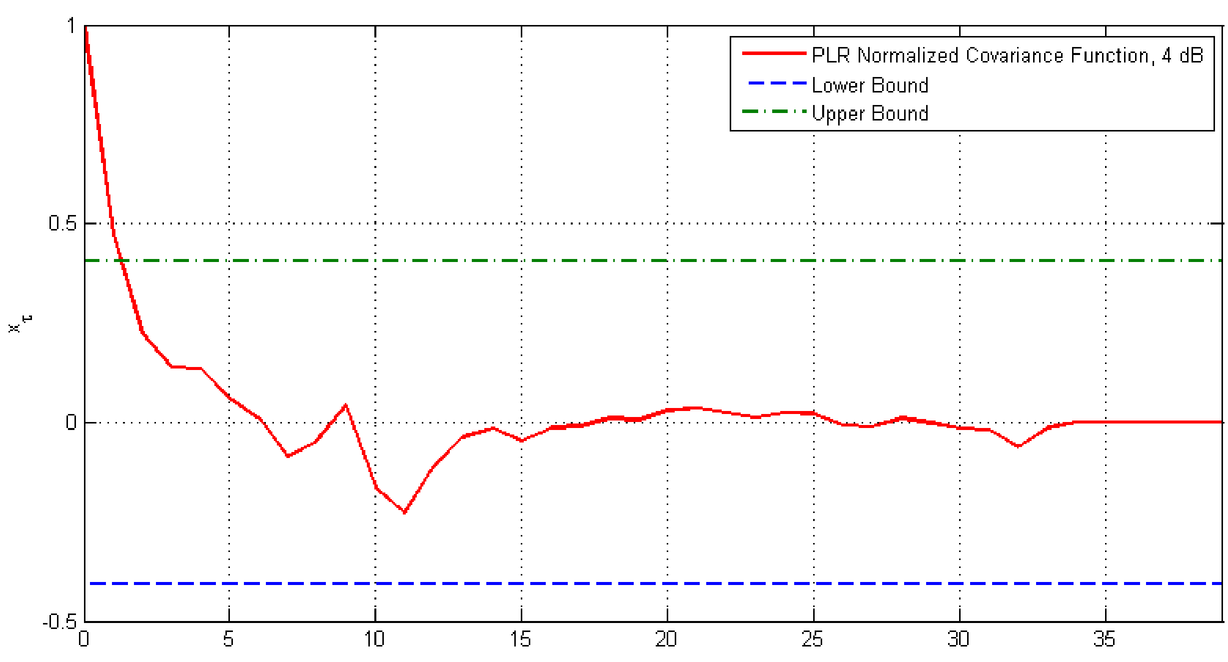

Fig. 6. Normalized covariance function of $\varepsilon(t)$ for the PLR algorithm in the worst condition (SNR = 4 dB)

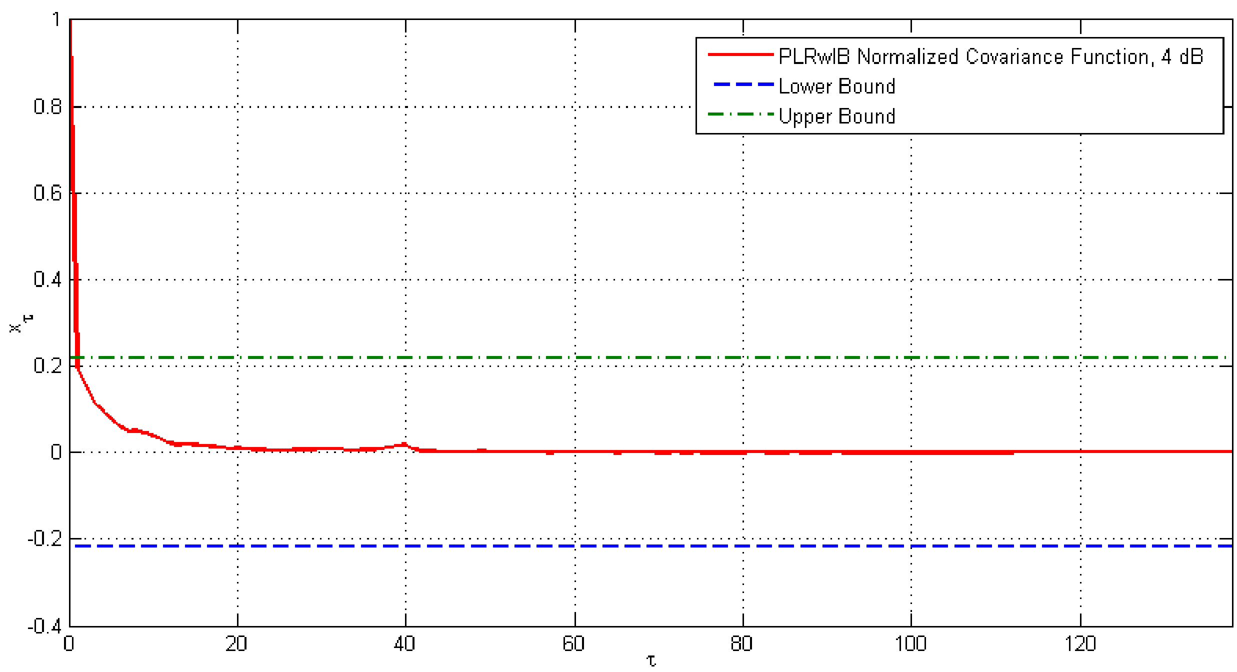

Fig. 7. Normalized covariance function of $\varepsilon(t)$ for the PLRwIB algorithm in the worst condition (SNR = 4 dB)

5.5 Results

The fundamental parameter used to analyze and compare the behaviour of the six types of peer-to-peer networks is represented by average percentage of correctly received bits as a function of depth level of the network. Through the simulation results we observe that by increasing the parameter of SNR (Signal to Noise Ratio) this produces an increase of the percentage of bits correctly received by each node of the network. Figures 8, 9 and 10 depict the comparisons of peer-to-peer networks under the six different types of algorithms we considered in section 3, with respect to the percentage of bits correctly received by each node with SNR = 4 dB, 7 dB and 10 dB. In this case the comparison parameter is the average percentage of correctly received bits as a function of depth level of the network. The best behaviour with respect to the average percentage of correctly received bits is obtained in the network generated by:

- the TR algorithm when the depth level is greater or equal to 4.
- The PLR algorithm and PLR algorithm without Input Blockage when the depth level is equal to 3.
- The PL algorithm when the depth level is less or equal to 2.

All the results, presented in this section, have been obtained by the following configuration parameters: number of bits supplied by the source node equal to 2048 Kbits divided in packets characterized by a length equal to 128 bits; we use the same sequences of peers, that require access to the network; dimension of each codeword is 16 bits and the number of bits that the receiver is able to detect and correct is 4 bits.

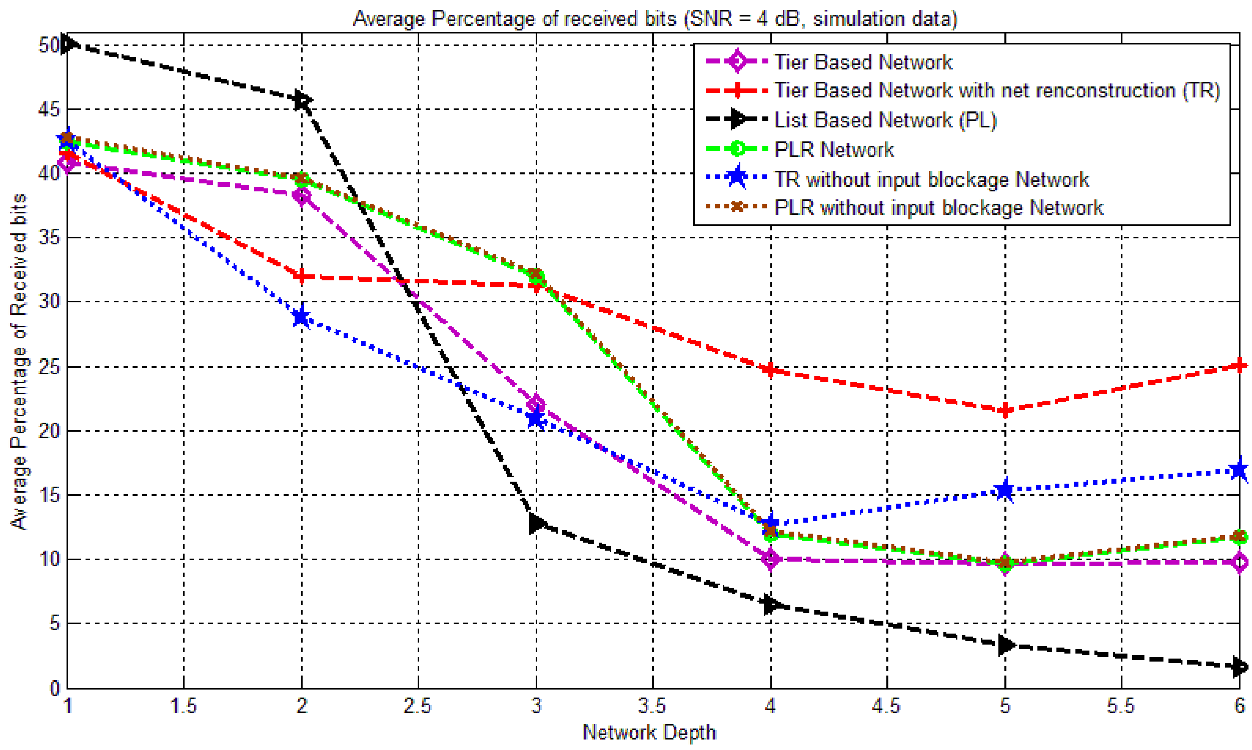

Fig. 8. Comparison, SNR = 4 dB

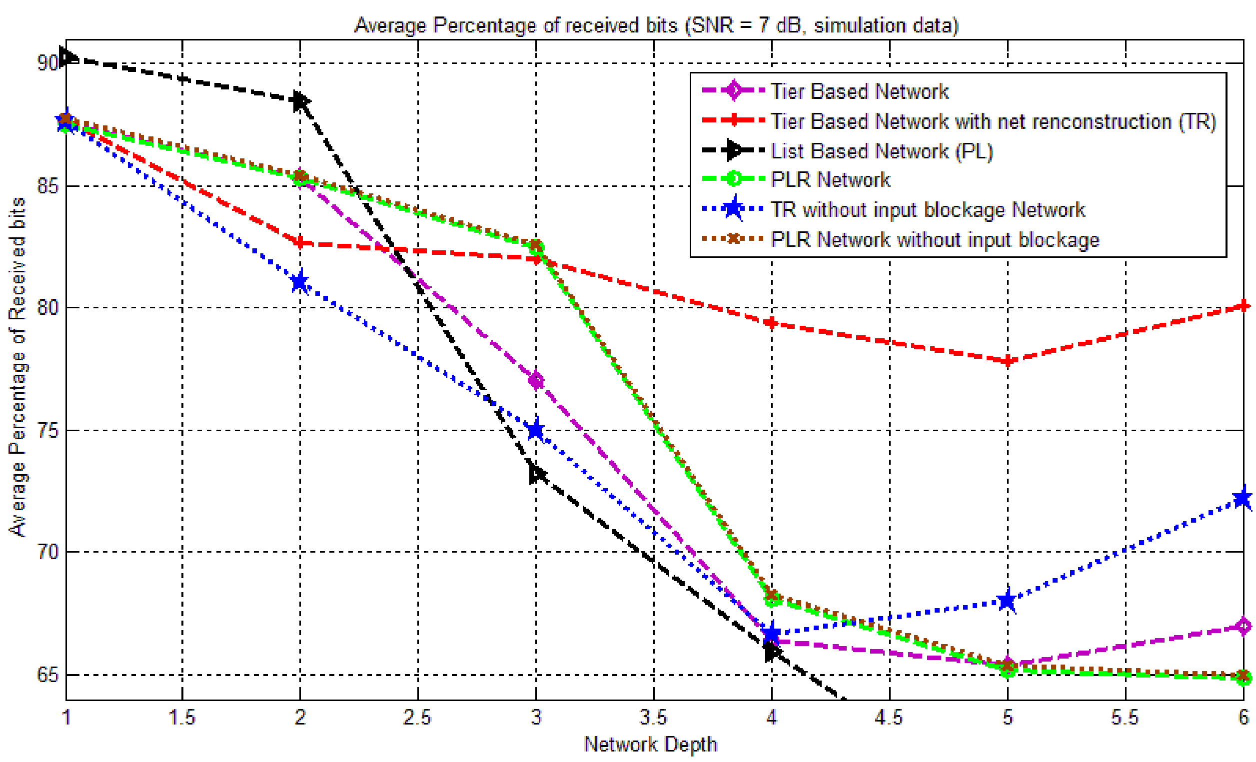

Fig. 9. Comparison, SNR = 7 dB

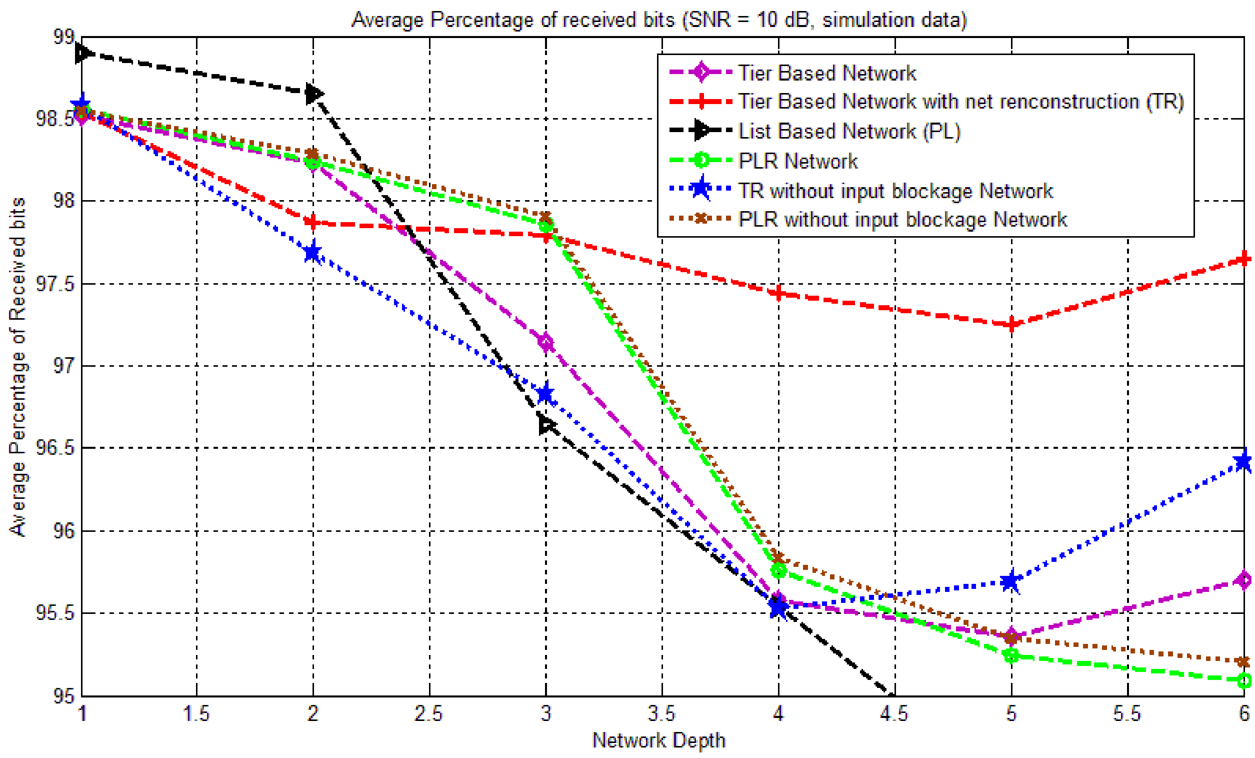

Fig. 10. Comparison, SNR = 10 dB

6. Conclusion

We can conclude that the maximization of the average maximum number of peers that can access the multimedia content and the minimization of the average maximum delay of the network is achieved, in the case of the asymmetric channel; when the source node is a home-user (where $1 \le B \le 2$) by using the PLR algorithm without Input Blockage, as in section 3 we showed that the PLR algorithm without Input Blockage is closest to optimum when $1 \le B \le 2$ and $0 \le p \le 1$. When the source node is a server (where $B \ge 2$) the best algorithm is:

- the TR algorithm without Input Blockage when $0.46 \le p \le 1$.
- The PLR algorithm without Input Blockage when $0 \le p < 0.46$.

We can also conclude that the TR and PLR algorithms without Input Blockage are a big improvement in comparison to Mazzini-Rovatti's algorithm (Mazzini & Rovatti, 2008) provided that new network conditions are followed, because they are suboptimal with respect to the theoretical optimum.

In the case of the radio channel, the best behaviour with respect to the percentage of correctly received bits is obtained in the network generated by:

- the TR algorithm when the depth level is greater or equal to 4.
- The PLR algorithm and PLR algorithm without Input Blockage when the depth level is equal to 3.
- The PL algorithm when the depth level is less or equal to 2.

7. References

Bakos, B., Farkas, L. & Nurminen, J. K. (2006). Peer-to-peer applications on smart phones using cellular communications, *WCNC 2006 proceedings*, ISBN 1-4244-0269-7, Las Vegas, Nevada, pp. 2222–2228, April 2006.

BitTorrent, http://www.bittorrent.com.

Boever, J. D. (2008). Value networks of peer-to-peer tv: An analysis of actors and their roles, *The Third International Conference on Internet and Web Applications and Services*, ISBN 978-0-7695-3163-2, Athens, Greece, pp. 686–695, June 2008.

Carter, C., Rhalibi, A.E., Merabti, M. & Bendiab, A.T. (2010). Hybrid Client-Server, Peer-to-Peer framework for MMOG, *Multimedia and Expo (ICME), 2010 IEEE International Conference on*, ISBN: 978-1-4244-7491-2, Suntec City, Singapore, pp. 1558 – 1563, July 2010.

Ciullo, D., Mellia, M., Meo, M., & Leonardi, E. (2008). Understanding peer-to-peer-tv systems through real measurements, *IEEE GLOBECOM 2008 proceedings*, ISBN: 978-1-4244-2324-8, New Orleans, Louisiana, pp. 1–6,November – December 2008.

KaZaA, http://www.kazaa.com.

Leonardi, E., Mellia, M., Horvath, A., Muscariello, L., Niccolini, S., & Rossi, D. (2008). Building a cooperative peer-to-peer-tv application over a wise network: The approach of the european fp-7 strep napa-wine, *IEEE Communications Magazine*, Vol. 46, No. 6, April 2008, pp. 20–22, ISSN: 0163-6804.

Lien, Y.-N. (2005). Performance issues of peer-to-peer file sharing over asymmetric and wireless networks, *ICDCSW '05: Proceedings of the First International Workshop on Mobility in Peer-to-Peer Systems*, ISBN : 0-7695-2328-5, Ohio, USA, pp. 850–855, June 2005.

Ma, M., Ma, Y., & Wang, J. (2008). The implementation and application of iptv supported on pull mode of peer-to-peer, *International Symposium on Knowledge Acquisition and Modeling*, ISBN : 978-0-7695-3488-6 , Wuhan, China, pp. 371–374, December 2008.

Matei, R., Iamnitchi, A. & Foster, P. (2002). Mapping the Gnutella network, *Internet Computing*, IEEE , Vol.6, No.1, January – February 2002, pp. 50-57, ISSN: 1089-7801.

Maymounkov, P. & Mazieres, D. (2002). Kademlia: A Peer-to-Peer Information System Based on the XOR Metric, *In International Workshop on Peer-to-Peer Systems, 2002*. Springer-Verlag, ISBN:3-540-44179-4, Cambridge, MA, USA, pp 53-65, March 2002.

Mazzini, G., & Rovatti, R. (2008). Peer-to-Peer distribution on asymmetric channels, *IEEE Communication Letters*, Vol. 12, No. 9, September 2008, pp. 699–701, ISSN: 1089-7798.

Merlanti D. and Mazzini G. (2009). Peer-to-peer distribution on radio channel, *Proceedings of the 17th international conference on Software, Telecommunications and Computer Networks. 24-26 Sept. 2009*, ISBN: 978-1-4244-4973-6, Hvar, pp.106-110, September 2009.

Papandreou, N. & Antonakopoulos, T. (2003). Cooperative bit-loading and fairness bandwidth allocation in adsl systems, *Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on*, ISBN: 0-7803-7761-3, Bangkok, II-352 - II-355 vol.2, May 2003.

Proakis, J. (15-th August 2000), Digital Communications 4th Edition, McGraw-Hill, ISBN: 978-0072321111, New York.

RFC 793 (1981), Transmission Control Protocol DARPA, Internet Program Protocol Specification, Information Sciences Institute University of Southern California 4676 Admiralty Way Marina del Rey, California 90291, September 1981.

RFC 791 (1981), Internet Protocol DARPA, Program Protocol Specification, Information Sciences Institute University of Southern California 4676 Admiralty Way Marina del Rey, California 90291, September 1981.

Soderstrom, T. & Stoica, P. (July 1989). System Identification, Prentice Hall, ISBN: 978-0138812362.

Tay, V. (2005). Massively multiplayer online game (MMOG) - a proposed approach for military application, *Cyberworlds, 2005. International Conference on Cyberworlds. 23-25 Nov. 2005. IEEE Computer Society*, ISBN: 0-7695-2378-1, Singapore, pp. 396-400, November 2005.

The Gnutella Protocol Specification v0.4, available on http://www.stanford.edu/class/cs244b/gnutella_protocol_0.4.pdf.

Tomoya, K., & Shigeki, Y. (2005). Application of peer-to-peer technology to marketing, *Proceedings of the 2003 International Conference on Cyberworlds (CW"03)*, ISBN: 0-7695-1922-9, Singapore, pp. 372–379, December 2003.

Wang, Y., Yun, X., Li, Y. (2007). Analyzing the Characteristics of Gnutella Overlays, *Information Technology, 2007. ITNG '07. Fourth International Conference on*, ISBN: 0-7695-2776-0, Las Vegas, Nevada, pp.1095-1100, April 2007.